SAPIENZA
UNIVERSITÀ DI ROMA

Dottorato di Ricerca in Statistica Metodologica

Tesi di Dottorato XXVII Ciclo – anno 2016

Dipartimento di Scienze Statistiche

Il contributo dei *topic models* alla navigazione del corpus delle sentenze della Cassazione

Paolo Fantini

Ad Agnese, Luigi e Claudio,
per il sostegno che mi hanno dato,
e per il tempo che con le loro rinunce mi hanno regalato.

A Ugo Berni Canani,
magistrato, linguista e matematico per passione e per curiosità,
informatico per necessità.

Indice

Elenco delle figure

Elenco delle tabelle

Introduzione

Come noto dalle indagini[1] sul funzionamento dei sistemi giudiziari europei della CEPEJ (The European Commission for the Efficiency of Justice), il nostro paese vanta da tempo il primato continentale del numero di cause giudiziarie avviate annualmente sia in materia civile che penale.
Questa enorme domanda di giustizia diventa a sua volta un insostenibile carico di lavoro per gli uffici giudiziari che, a detta di molti esperti, è una tra le ragioni più importanti della loro inefficienza.
Tutto ciò è particolarmente vero per il settore della giustizia civile, la cui lentezza, come molte volte sottolineato dalla Banca d'Italia, è tra i fattori che frenano l'economia e scoraggiano l'arrivo di investimenti stranieri in Italia.
Limitandoci dunque al settore civile e correndo forse il rischio di semplificare oltre il dovuto, il Ministero della Giustizia, nella sua veste di "organizzatore" della macchina giudiziaria, 1) può intervenire sull'offerta di giustizia migliorando la produttività degli uffici giudiziari, 2) può concorrere a deflazionare la domanda di giustizia rendendo il ricorso al giudice meno conveniente rispetto a forme alternative di risoluzione delle controversie (come ad esempio, la cosiddetta mediazione civile ad opera di soggetti accreditati che non sono tuttavia giudici).
Quanto al primo punto, da almeno un decennio gli uffici giudiziari italiani (soprattutto quelli di primo grado: i tribunali) sono seriamente impegnati in un complesso processo di digitalizzazione delle loro molteplici attività. Si pensi in particolare al Processo Civile Telematico (PCT) che, introdotto dopo una lunga fase sperimentale, sta gradualmente rimpiazzando il tradizionale flusso di lavoro basato esclusivamente su scambi di documentazione cartacea. Con i vantaggi che è facile immaginare in termini di tempi e costi.
Diretta conseguenza di questo fenomeno la produzione via via crescente di documenti *nativi* digitali, specialmente ricorsi e sentenze. Una ricchezza di informazioni, questa, che sconta ad oggi la mancanza di strumenti adeguati a renderla accessibile agli operatori del diritto (giudici e avvocati) e al più ampio pubblico dei cittadini.

[1] Si confronti il rapporto CEPEJ 2016 scaricabile dal sito www.coe.int.

Non mancano tuttavia le eccezioni. La principale delle quali costituita da *ItalGiure*, l'archivio elettronico della Corte Suprema di Cassazione[2] attivo a partire dagli inizi degli anni '70 e al quale vanno collegati l'origine e lo sviluppo dell'informatica giuridica in Italia.
L'archivio ItalGiure fu l'intuizione di maggior successo del Centro di Elaborazione Documentale[3] (CED) istituito nel 1970 presso l'Ufficio del Massimario della Corte di Cassazione. Inizialmente progettato per risolvere problemi di ricerca di precedenti giurisprudenziali, condensati allora in circa 300.000 schede di massime[4] inserite in appositi armadi raccoglitori e classificate manualmente per voci (Grandi voci) e sottovoci (Piccole Voci), è diventato nel tempo «la più grande banca dati telematica a livello nazionale in materia di documentazione giuridica, in termini di completezza, integrazione e accessibilità»[Peruginelli & Ragona (2014)].
Attualmente ItalGiure è articolato in una pluralità di archivi contenenti più di 35 milioni di documenti (accessibili via web[5] e navigabili utilizzando i tradizionali strumenti di *information retrieval*), che vanno dalla banca dati della legislazione italiana (a partire dal 1861) fino a quella della legislazione europea, passando per le massime e le sentenze in materia civile e penale pronunciate dalla Corte di Cassazione nel corso della sua lunga storia. In particolare, ad oggi[6] l'archivio delle massime risulta composto da 518.818 documenti in materia civile e 169.937 documenti in materia penale, mentre l'archivio delle sentenze conta 453.771 documenti in materia civile e 639.797 documenti in materia penale.
Si osservi tuttavia che contrariamente all'archivio delle massime, da sempre oggetto

[2]La giurisdizione ordinaria italiana prevede tre gradi di giudizio, con una Corte Suprema (di Cassazione) a fare da chiusura e vertice del sistema. Da questa posizione, la Corte svolge una funzione essenzialmente nomofilattica ed unificatrice, con il fine di assicurare "l'esatta osservanza e l'uniforme interpretazione della legge, l'unità del diritto oggettivo nazionale, il rispetto dei limiti delle diverse giurisdizioni" (art. 65 della Legge 12/1941 - legge fondamentale sull'ordinamento giudiziario). Come nella gran parte dei sistemi di *civil law* mutuati dal modello francese, la Corte è giudice supremo di legittimità, ovvero non entra nel merito della decisione del giudice di grado inferiore ma si limita a verificare la corretta applicazione della legge.

[3]La legge (DPR 195/2004, art. 1, co. 1) assegna al CED della Corte di Cassazione il compito di "svolgere un servizio pubblico di informatica giuridica, per diffondere la conoscenza della normativa, della giurisprudenza e della dottrina giuridica".

[4]Una *massima* è un principio di diritto stabilito dalla Corte di Cassazione nella sua funzione di interprete ultimo della legge (funzione nomofilattica). I giudici delle giurisdizioni inferiori in generale vi si uniformano, pur non essendone formalmente tenuti. A differenza infatti di ciò che accade nei sistemi di *common law*, nei sistemi di *civil law* (come il nostro) il precedente non è vincolante che per il solo giudice al quale la corte abbia eventualmente rinviato il caso per il riesame dei fatti relativi alla causa.

[5]www.italgiure.giustizia.it.

[6]Dato aggiornato ad Ottobre 2016.

di cura certosina[7] da parte dei magistrati dell'Ufficio del Massimario, quello delle sentenze (di più recente istituzione) appare come un mero elenco di documenti senza alcuna *struttura*, né manuale né automatica. Circostanza, quest'ultima, che è all'origine di questo lavoro[8].

In continuità ideale con la tradizione di sperimentazioni innovative del CED della Cassazione, stimolate in primo luogo da figure straordinarie come quella del compianto magistrato Ugo Berni Canani[9], qui illustriamo infatti i risultati della sperimentazione di metodi e tecniche di *Machine Learning* per l'analisi di testi in linguaggio naturale, con lo scopo di fornire una *struttura* a corpus di documenti altrimenti non strutturati.

Il primo passo per identificare la struttura di un documento consiste nel determinare i suoi "temi". È ciò che tentano di fare i cosiddetti *topic models* [Blei & Lafferty (2009)], i quali mirano ad automatizzare il processo di estrazione dei temi dai documenti.

Un topic model è infatti un semplice modello probabilistico che descrive il processo di generazione delle parole di un documento.

Siano $\mathcal{D}$ un corpus di documenti e $\mathcal{T}$ il vocabolario dei suoi termini (unici). Nell'approccio *Latent Dirichelet Allocation* (LDA) in Blei et al. (2003), un *topic* è una distribuzione discreta di probabilità[10] su $\mathcal{T}$, $\boldsymbol{\phi}_k = (\phi_{k1}, \phi_{k2}, \ldots, \phi_{k|\mathcal{T}|})$, e un documento $d \in \mathcal{D}$ è una mistura di un numero fissato K di topic (latenti) comuni a tutto il corpus.

Ogni documento viene generato scegliendo una distribuzione sui topic ed estraendo ogni sua parola da un topic selezionato in base a questa distribuzione. Se con $\boldsymbol{\theta}_d = (\theta_{d1}, \theta_{d2}, \ldots, \theta_{dK})$ indichiamo i coefficienti della mistura, otteniamo una rappresentazione esplicita del documento d come punto del simplesso $K-1$ dimensionale generato dai topic.

Diversamente da approcci più tradizionali, che trattano i documenti come vettori di pesi associati ai $|\mathcal{T}|$ termini del vocabolario (in genere frequenze o loro trasformazioni), in un approccio di tipo topic model un documento viene *ridotto* al vettore $\boldsymbol{\theta}_d$ (per-document topic proportions) di dimensione K, con $K \ll |\mathcal{T}|$.

Con un certo abuso di terminologia, potremmo dire che il contenuto di un documento corrisponde ai suoi "temi" (i topic) piuttosto che alle sue parole (i termini in $\mathcal{T}$).

[7] Limitandosi alla sola materia civile, il primo livello della classificazione (manuale) comprende circa 250 voci (le cosiddette Grandi voci).

[8] Risultato di una collaborazione che ha visto coinvolti il CED della Corte di Cassazione, il Dipartimento di Scienze Statistiche de La Sapienza di Roma e la Direzione di Statistica del Ministero della Giustizia, presso la quale l'autore ricopre il ruolo di funzionario statistico per il settore civile.

[9] Direttore del CED a cavallo tra gli anni '80 e '90.

[10] Ai valori più elevati di questa distribuzione sono associati i cosiddetti *top terms*, ossia i termini che identificano il contenuto semantico di un topic.

Riuscire a dotare un corpus (non altrimenti strutturato) di una struttura *tematica* potrebbe facilitare compiti come la ricerca di gruppi di documenti correlati o anche di documenti con temi simili a quelli di un documento assegnato.
Ciò potrebbe risultare di grande utilità, ad esempio come primo filtro nella distribuzione (automatica) dei ricorsi pendenti presso le varie sezioni della Corte di Cassazione specializzate per materia o nell'agevolare la *cultura del precedente* e lo scambio di informazioni tra gli addetti ai lavori.
Si noti tra l'altro che il punto di vista di questo lavoro è molto differente rispetto a quello di altri esperimenti che il Ministero della Giustizia sta attualmente portando avanti in collaborazione con l'Istituto di Teoria e Tecniche dell'Informazione Giuridica del Consiglio Nazionale delle Ricerche (ITTIG-CNR).
Da tempo quest'ultimo è fortemente impegnato in attività di sperimentazione di metodologie e implementazione di prototipi software con l'obiettivo di arricchire di metadati semantici documenti giurisprudenziali in maniera automatica[11].
L'approccio seguito dall'ITTIG-CNR si ispira ai principi della *Knowledge Engineering* (KE) e richiede un dispendioso intervento esterno di "codifica della conoscenza" ad opera di esperti del dominio applicativo.
Nulla di tutto questo è richiesto invece nell'approccio LDA, dove l'unico parametro da fissare a priori è K, il numero dei topic.
In termini più generali, LDA è un modello bayesiano gerarchico per dati discreti che generalizza in senso probabilistico tecniche algebriche, sviluppate inizialmente negli anni '90, di riduzione di testi in linguaggio naturale ai "concetti" in essi contenuti. Il riferimento fondamentale è a questo riguardo alla *Latent Semantic Analysis* (LSA) in Deerwester et al. (1990).
La strategia di ricerca dei topic del modello LDA richiede essenzialmente il calcolo di distribuzioni a posteriori su insiemi di variabili latenti. Ciò comporta tuttavia problemi di inferenza intrattabili, che ammettono soluzioni soltanto approssimate.
Varie tecniche di inferenza a posteriori approssimata sia deterministiche che stocastiche sono state applicate con successo alla risoluzione di questi problemi. Nel primo caso, basate su metodi variazionali (Variational Expectation Maximization) e nel secondo, su metodi del tipo Marchov Chain Monte Carlo (Gibbs sampling).
La nostra sperimentazione è stata condotta su un corpus di 74.858 sentenze civili (appartenenti a 20 materie differenti) pubblicate dalla Corte di Cassazione nel quinquennio 2010-2014.
Il modello utilizzato è stato addestrato tramite MALLET (MAchine Learning for LanguagE Toolkit) [Yao et al. (2009)], una suite Java che implementa un efficiente

[11] Si confronti quanto riportato in Agnoloni et al. (2014) relativamente alla categorizzazione automatica delle decisioni degli uffici giudiziari di primo grado (i tribunali).

Gibbs sampling per l'approssimazione delle distribuzioni a posteriori.
Due gli scenari considerati: il primo, di *base*, nel quale i documenti del corpus non sono stati pre-trattati se non per aspetti marginali, il secondo, *ridotto*, nel quale essi sono stati sottoposti ad una operazione di riduzione per selezione dei termini (secondo lo schema *tfidf* in Salton & Bucley (1988)) prima di essere passati in input al modello LDA.
In entrambi i casi i topic estratti si sono dimostrati in grado di catturare strutture semantiche significative nei dati, consistenti con le classi di materia originarie. Nondimeno, lo scenario ridotto ha fatto registrare tempi di esecuzione di molto inferiori rispetto a quello di base.
I risultati ottenuti sono stati presentati al CED della Corte di Cassazione tramite `Suprema`[12], un prototipo di applicazione web appositamente realizzato e basato sulla rappresentazione esplicita dei documenti in termini di vettori $\boldsymbol{\theta}_d$ e $\boldsymbol{\phi}_k$.
`Suprema` rende disponibili una serie di funzioni di navigazione aventi l'ambizione di integrare quelle attualmente utilizzabili in ItalGiure per l'archivio delle sentenze. Limitandosi alle più importanti, consente una veloce panoramica tematica del corpus, raggruppa le sentenze per topic più probabile, trova le sentenze più rilevanti per classe di materia e per topic, e infine cerca le sentenze simili (per temi trattati) ad una sentenza data.
In conclusione, due piccole curiosità.
La prima: grazie a `Suprema` siamo riusciti a isolare facilmente all'interno del nostro corpus diversi sottoinsiemi di documenti molto simili (quando non perfettamente identici), conseguenza di un fenomeno ben noto agli addetti ai lavori, quello dei ricorsi seriali (cui non può che seguire una sentenza altrettanto seriale). Si tratta generalmente di ricorsi prodotti in serie da gruppi di avvocati organizzati (e spesso molto interessati), ai quali negli ultimi tempi si è cercato di porre un argine introducendo anche nel nostro ordinamento l'istituto dell'azione collettiva (meglio nota come *class action*).
La seconda: il CED della Cassazione ha valutato positivamente i risultati (pur parziali) ottenuti. Al punto di proporci due ulteriori sviluppi. Da un lato estendere l'analisi all'intero corpus delle sentenze civili e penali. Dall'altro approfondire la questione dell'uso di LDA come filtro iniziale per la distribuzione automatica ai vari giudici specializzati per materia dei ricorsi pendenti in ambito tributario[13], non appena sarà terminato il processo della loro conversione in formato digitale.
Allo stato attuale delle cose, questi ricorsi vengono esaminati manualmente (uno per uno) da parte di un ufficio all'uopo dedicato e distribuiti alle varie sezioni giudicanti dopo averne letto il contenuto, con enorme dispendio di tempo ed energie.

[12] https://cassazione.shinyapps.io/Suprema/ (demo con $K = 50$).
[13] Alla data di Ottobre 2016 risultavano ancora da decidere più di 100.000 ricorsi tributari.

Tenendo conto del fatto che la Corte di Cassazione costituirà il modello verso il quale tenderanno tutti gli uffici giudiziari italiani quando il processo civile telematico andrà finalmente a regime, la possibilità di filtrare automaticamente (seppur solo in prima battuta) i ricorsi in entrata attraverso un topic model, cioè senza aver bisogno di alcuna codifica di conoscenza esperta, apre prospettive degne delle migliori considerazioni.
E qui sta forse il vero contributo di questa tesi.
Che peraltro è così articolata: nel capitolo 1, descriveremo il modello LDA e il percorso che ha portato alla sua introduzione; nei capitoli 2 e 4, illustreremo i risultati dei due scenari di sperimentazione (di base e ridotto) e ci soffermeremo in particolare sulla determinazione del valore di K; in mezzo, nel capitolo 3, presenteremo `Suprema` e le sue funzioni di navigazione applicandole a casi concreti.

Capitolo 1

Il modello *Latent Dirichlet Allocation*

Come già osservato nelle pagine introduttive, l'obiettivo di questo lavoro è quello di sperimentare metodi e modelli per ottenere descrizioni *tematiche* degli elementi di un corpus di documenti estratti dall'archivio delle sentenze civili della Corte Suprema di Cassazione (ItalGiure). Questo allo scopo di ampliare le possibilità di analisi, dotando di struttura un corpus altrimenti non strutturato.
Inizieremo, con questo capitolo, dall'illustrare le caratteristiche fondamentali di alcuni tra i più importanti modelli generativi di documenti: i cosiddetti "topic models". Un topic model è un modello generativo nel senso che specifica una semplice procedura probabilistica attraverso la quale i documenti di un corpus possono essere generati. È basato infatti sull'idea che quest'ultimi siano misture di topic, dove un topic una distribuzione di probabilità sui termini di un vocabolario. Per generare un nuovo documento si sceglie una distribuzione sui topic e poi, per ogni parola in quel documento, si sceglie un topic a caso secondo questa distribuzione, e si estrae una parola da quel topic.
Tecniche statistiche standard (Variational EM e Gibbs sampling) possono essere utilizzate per invertire questo processo e inferire dall'osservazione delle parole contenute nei documenti l'insieme dei topic e i coefficienti della mistura responsabili di aver generato ogni documento di un dato corpus.
Nel seguito descriveremo nel dettaglio il modello *Probabilistic Latent Semantic Indexing* (PLSI) in Hofmann (1999) e la sua generalizzazione chiamata *Latent Dirichlet Allocation* (LDA) in Blei et al. (2003), che è poi il topic model più diffuso nelle applicazioni pratiche (e anche il modello utilizzato nelle nostre sperimentazioni). Non senza prima aver fatto qualche cenno al modello *Latent Semantic Indexing* in Deerwester et al. (1990) che dei primi due appare come una sorta di progenitore.

1.1 Grafi orientati aciclici e modelli generativi

C'è una maniera di rappresentare graficamente un'ampia classe di distribuzioni di probabilità congiunte su un insieme finito di variabili casuali, che utilizzeremo nel seguito quando ci tornerà utile ridurre in componenti più semplici modelli probabilistici intrinsecamente complessi. È ciò che in letteratura è noto come *Modello Grafico Probabilistico* (Probabilistic Graphical Model).
Un modello grafico probabilistico è un grafo nel quale ogni nodo rappresenta una variabile casuale (o un insieme di variabili casuali) e gli archi che connettono i nodi indicano l'esistenza di relazioni probabilistiche tra questi. In generale, un modello grafico probabilistico esplicita il modo in cui la distribuzione di probabilità congiunta su tutti i nodi può essere decomposta in un prodotto di fattori dipendenti soltanto da un loro sottoinsieme.
Chiameremo *grafo orientato* una coppia ordinata $\mathcal{G} = (N, A)$ dove N è l'insieme dei nodi e A è l'insieme degli archi, ovvero coppie ordinate di nodi.
Un arco $a = (p, f)$ è diretto dal nodo p al nodo f; p prende il nome di *padre* di f e f quello di *figlio* di p. Due nodi sono detti *adiacenti* se condividono uno stesso arco.
Un *cammino orientato* è una sequenza di archi adiacenti, tutti orientati nella stessa direzione. Un grafo orientato è *aciclico* (Directed Acyclic Graph: *DAG*) se non ha cicli orientati, cioè, cammini orientati che inizino e finiscano sullo stesso nodo.
Supponiamo che $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_m)$ sia un vettore di m variabili casuali.
In generale, la distribuzione di probabilità congiunta di $\mathbf{Y}$ può essere espressa come prodotto di distribuzioni condizionate:

$$p(Y_1, Y_2, \ldots, Y_m) = p(Y_m|Y_1, Y_2, \ldots, Y_{m-1}) \cdots p(Y_2|Y_1)p(Y_1). \tag{1.1}$$

Per una data scelta di m, è possibile rappresentare questa distribuzione come un DAG dove ogni nodo corrisponde a una delle variabili casuali e, per ogni distribuzione condizionata, viene tracciato un arco dai nodi corrispondenti alle variabili rispetto alle quali la distribuzione è condizionata.
In Figura 1.1 riportiamo un semplice esempio per $m = 4$, dove la distribuzione congiunta ha la forma:

$$p(Y_1, Y_2, Y_3, Y_4) = p(Y_4|Y_1, Y_2, Y_3)p(Y_3|Y_1, Y_2)p(Y_2|Y_1)p(Y_1).$$

Il grafo in Figura 1.1 è detto *completamente connesso* poichè presenta un arco tra ogni coppia di nodi, e dunque risulta utile a rappresentare una distribuzione congiunta completamente generale, come quella in Eq. 1.1. Da notare tuttavia che è l'*assenza* di legami tra i nodi a fornire le informazioni più interessanti circa le proprietà della classe di distribuzioni che il grafo rappresenta.

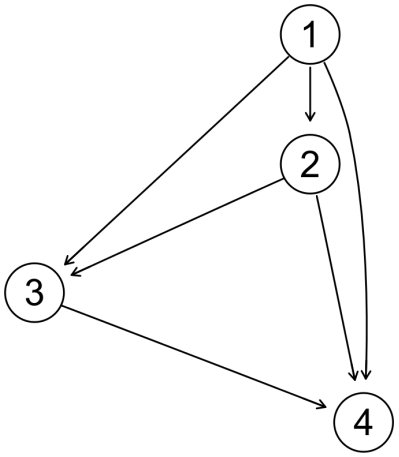


Figura 1.1: Un semplice DAG completamente connesso

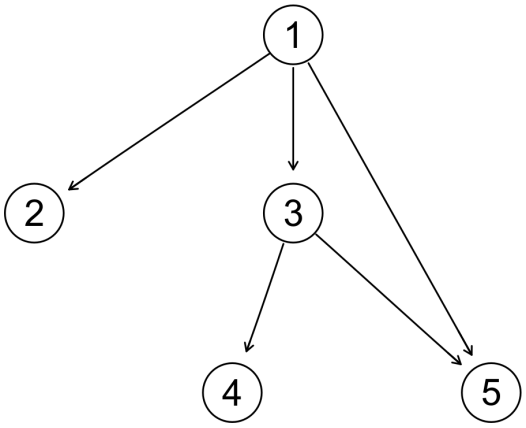


Figura 1.2: Un semplice DAG per la distribuzione congiunta sulle variabili $Y_1, Y_2 \ldots, Y_5$

Ad esempio, il grafo in Figura 1.2, che non è completamente connesso, descrive la seguente distribuzione congiunta:

$$p(Y_1, Y_2, Y_3, Y_4, Y_5) = p(Y_1)p(Y_2|Y_1)p(Y_3|Y_1)p(Y_4|Y_3)p(Y_5|Y_1, Y_3),$$

a partire dalla quale possiamo introdurre la relazione generale tra un DAG e la corrispondente distribuzione sulle variabili casuali associate ai nodi:

Definition 1.1.1. Siano $\mathcal{G} = (N, A)$ un grafo orientato aciclico e P la distribuzione di probabilità congiunta sul vettore casuale $\mathbf{Y}$ associato a N. Diciamo che P è *markoviana* rispetto a $\mathcal{G}$ quando:

$$p(\mathbf{Y}) = \prod_{i=1}^{m} p(Y_i|\mathrm{pa}_i).$$

In altre parole, la distribuzione congiunta corrispondente a un grafo orientato aciclico è data dal prodotto delle distribuzioni di ogni nodo i, condizionate dall'in-

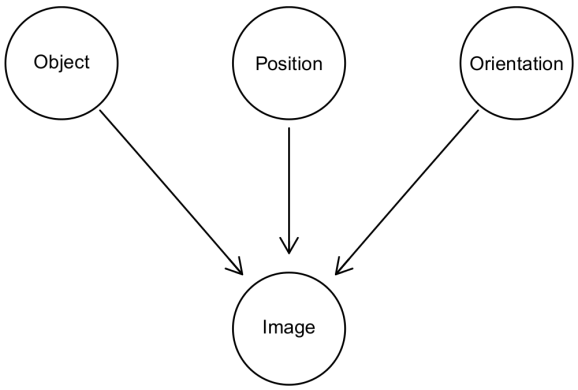


Figura 1.3: Un esempio di modello generativo

sieme dei *padri* di quel nodo (l'insieme pa_i).
Si osservi che oltre ai grafi orientati aciclici, la teoria dei modelli grafici probabilistici include anche i grafi non orientati (in letteratura noti come *Markov Random Fields*). I primi[1] si prestano perlopiù a esprimere relazioni causali tra variabili aleatorie, e i secondi vincoli più deboli tra queste variabili.
In un grafo orientato aciclico infatti l'ordinamento dei nodi è fisso, ovvero non ci possono essere archi diretti da un nodo con numerazione inferiore a un altro con numerazione superiore.
Nelle applicazioni pratiche, i nodi terminali (quelli con numerazione superiore) rappresentano tipicamente le *osservazioni*, i nodi iniziali (quelli con numerazione inferiore) le *variabili latenti*, dalle quali le osservazioni dipendono[2].
Consideriamo a questo proposito il modello grafico in Figura 1.3. Esso si riferisce a un problema di riconoscimento di oggetti [Bishop (2007)], nel quale ogni osservazione corrisponde a un'immagine (un vettore di intensità di pixel) di un oggetto e i nodi *Object*, *Position*, *Orientation* corrispondono a un insieme di variabili latenti, dai quali la distribuzione del nodo *Image* è dipendente. L'obiettivo è trovare la distribuzione sulla variabile latente *Object* data una particolare immagine osservata, $p(Object|Image)$.
Tali modelli prendono il nome di *modelli generativi* (Generative Models), in quanto descrivono il processo *causale* di generazione dei dati osservati.
In questo capitolo ci occuperemo di modelli generativi per gli elementi di un corpus di documenti di testo. Tali modelli si basano su semplici regole probabilistiche che descrivono come le parole nei documenti possono essere generate a partire da un insieme di variabili casuali latenti.

[1] Spesso chiamati anche *reti bayesiane* (bayesian networks).

[2] Il ruolo principale delle variabili latenti consiste nel fare in modo che una distribuzione complessa sulle variabili osservate possa essere rappresentata in termini di più semplici distribuzioni condizionate.

Ma prima ripercorriamo brevemente il percorso che ha portato alla definizione del modello *Latent Dirichlet Allocation* (LDA), il modello che abbiamo utilizzato nei nostri esperimenti.

1.2 Il modello *Latent Semantic Indexing*

In questo paragrafo discuteremo il problema di trovare descrizioni *ridotte* degli elementi di un corpus di documenti di testo in linguaggio naturale.
Tali descrizioni devono poter rendere possibili elaborazioni efficienti di grandi collezioni di dati e allo stesso tempo preservare quelle relazioni statistiche che sono essenziali per compiti basilari di esplorazione del corpus (come la classificazione dei documenti, la ricerca di insiemi di documenti correlati e più in particolare la ricerca di documenti simili a un documento assegnato).
I primi ad affrontare seriamente la questione furono nel corso degli anni '80 i ricercatori nel campo dell'*information retrieval* (IR), i quali proposero inizialmente di ridurre documenti di lunghezza arbitraria a vettori numerici di lunghezza fissata.
In letteratura questo approccio è chiamato *bag of words*[3]: ogni documento d è rappresentato come un vettore di pesi $d = (w_1, w_2, \ldots, w_{|\mathcal{T}|})$, dove $\mathcal{T}$ è il vocabolario dei termini (*terms*) del corpus, ovvero la lista dei termini ricorrenti almeno una volta nel corpus esaminato, e w_j misura il contributo del termine t_j alla descrizione del contenuto del documento d.
Il caso più semplice è quello in cui i pesi corrispondano alla frequenza dei termini nei documenti. Tuttavia, spesso si ricorre alla funzione *tfidf*[4] in Salton & Bucley (1988), definita come:

$$tfidf(d_i, t_j) = n_{ij} \cdot \log \frac{|\mathcal{D}|}{n_j}, \tag{1.2}$$

dove $\mathcal{D}$ è l'insieme dei documenti del corpus, n_{ij} è la frequenza relativa del termine t_j nel documento d_i e n_j è il numero di documenti che contengono il termine t_j. Ne segue che la *rilevanza* di un termine per un documento è direttamente proporzionale alla sua frequenza in quel documento e inversamente proporzionale alla sua frequenza nell'intero corpus. Il tentativo è chiaramente quello di individuare in questo modo i migliori candidati a rappresentare il contenuto dei singoli documenti.
Nella pratica si impiegano spesso forme di normalizzazione per ottenere pesi compresi nell'intervallo $[0, 1]$. La più diffusa delle quali è quella di tipo coseno:

$$w_{ij} = \frac{tfidf(d_i, t_j)}{\sqrt{\sum_{j=1}^{|\mathcal{T}|} tfidf^2(d_i, t_j)}}. \tag{1.3}$$

[3] O anche *Vector Space Model*.
[4] *tfidf* sta per term frequency-inverse document frequency.

La matrice $DTM = \{w_{ij}\}$ costruita a partire dai documenti-vettore prende il nome di matrice documenti per termini (**D**ocument **T**erm **M**atrix) ed è ovviamente basata sull'assunzione che l'ordine dei termini in un documento sia un fattore trascurabile. Lo schema *tfidf* ha diversi aspetti positivi, su tutti, come già osservato, la vettorizzazione dei documenti e l'identificazione di termini *discriminativi*. E però rivela poco della struttura statistica nei e tra i documenti del corpus.
Per superare questi limiti, un nuovo metodo di natura algebrica, noto come *Latent Semantic Indexing* (LSI), fu proposto all'inizio degli anni '90 da Deerwester et al. (1990).
LSI opera una decomposizione in valori singolari della matrice DTM con lo scopo di identificare un sottospazio lineare nello spazio dei punteggi *tfidf* in grado di catturare la gran parte della varianza del corpus.
L'idea chiave è quella di proiettare i documenti-vettore su uno spazio vettoriale di dimensione ridotta, il cosiddetto *spazio semantico latente*.
Si può dimostrare infatti che è sempre possibile scomporre la matrice DTM come:

$$DTM = U\Sigma V^t, \tag{1.4}$$

dove U, e V sono matrici ortogonali ($U^tU = V^tV = I$) e Σ è la matrice diagonale che contiene i *valori singolari* di DTM, $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r$ (con r rango di DTM).
Tenendo tra i valori singolari soltanto i primi K più elevati e annullando i rimanenti si ottiene la matrice diagonale $\widetilde{\Sigma}$ tramite la quale approssimare la matrice DTM:

$$D\tilde{T}M = U\widetilde{\Sigma}V^t \approx U\Sigma V^t = DTM. \tag{1.5}$$

Come ben noto dall'algebra lineare, l'approssimazione $D\tilde{T}M$ ha rango K ed è ottimale nel senso della norma L_2, ossia è una proiezione ortogonale di DTM su un sottospazio lineare di dimensione ridotta ($K < R$).
A differenza dell'approccio $tfidf$, quello LSI consente significative compressioni dei documenti. Inoltre, documenti che condividono termini frequentemente cooccorrenti hanno una rappresentazione simile nello spazio semantico latente (anche senza termini in comune). E dunque LSI riesce a cogliere aspetti di nozioni linguistiche di base, come la sinonimia (più termini che hanno lo stesso significato) e la polisemia (lo stesso termine che ha significati diversi), che costituiscono il cuore di ciò che più avanti chiameremo *topic*. Aspetti che l'approccio $tfidf$ non riesce invece a cogliere.
LSI è basato esclusivamente su considerazioni di tipo algebrico e prescinde dall'esistenza di un modello di generazione dei dati. Sebbene sia stato applicato con notevole successo, soprattutto nell'ambito dell'indicizzazione[5] automatica dei docu-

[5] Con il termine *indicizzazione* si indica l'attività di rappresentare in forma compatta il contenuto testuale di un documento.

menti [Deerwester et al. (1990), Dumais (1995)], presenta notevoli lacune derivanti dai suoi insoddisfacenti presupposti statistici.
Sono queste lacune ad averne determinato una riformulazione in termini probabilistici, come modello generativo dei dati, che non poteva che chiamarsi *Probabilistic Latent Sematic Indexing* (PLSI) [Hofmann (1999)].
PLSI "aggiunge" a LSI un modello di generazione dei dati, cioè sostanzialmente una distribuzione di probabilità sulle coppie di osservazioni (d, w), essendo d un documento del corpus e w una parola del vocabolario $\mathcal{T}$.
Alla descrizione di questo modello, che storicamente è stato il primo esempio di topic model, sarà dedicata la sezione che segue.
In definitiva, l'approccio LSI considera tre aspetti: 1) l'informazione semantica sta nella matrice documenti per termini (bag of words), 2) la riduzione di dimensionalità di quest'ultima è una questione essenziale, 3) le parole e i documenti possono essere rappresentati come punti in uno spazio euclideo.
L'approccio di tipo topic model condivide i punti 1) e 2) ma differisce nel terzo, in quanto le proprietà semantiche delle parole e dei documenti sono espresse in termini di distribuzioni di probabilità. Che, come tra poco vedremo, sono individualmente interpretabili con facilità.

1.3 Il modello *Probabilistic Latent Semantic Indexing*

Formalmente PLSI è un modello a variabili latenti[6] per dati discreti, nel quale si suppone che una classe latente $z \in \mathcal{Z} = \{1, 2, \ldots, K\}$ sia associata a ogni occorrenza di una parola $w \in \mathcal{T} = \{w_1, w_2, \ldots, w_T\}$ in un documento $d \in \mathcal{D} = \{d_1, d_2, \ldots, d_D\}$.
Il problema dal quale si parte è quello di assegnare una distribuzione di probabilità congiunta sulle coppie di osservazioni (d, w): $p(d, w)$.
Questo problema può essere semplificato se si introduce la variabile latente z e si fa l'assunzione che d e w siano indipendenti condizionatamente ad essa.
Infatti:

$$p(d, w) = \sum_{z \in \mathcal{Z}} p(d, w, z) = \sum_{z \in \mathcal{Z}} p(d, w|z)p(z) = \sum_{z \in \mathcal{Z}} p(d|z)p(w|z)p(z), \tag{1.6}$$

dove si è appunto sfruttato il fatto che, per assunzione modellistica, $p(d, w|z) = p(d|z)p(w|z)$.

[6] I modelli a variabili latenti sono distribuzioni di probabilità strutturate nelle quali dati osservati interagiscono con variabili casuali nascoste.

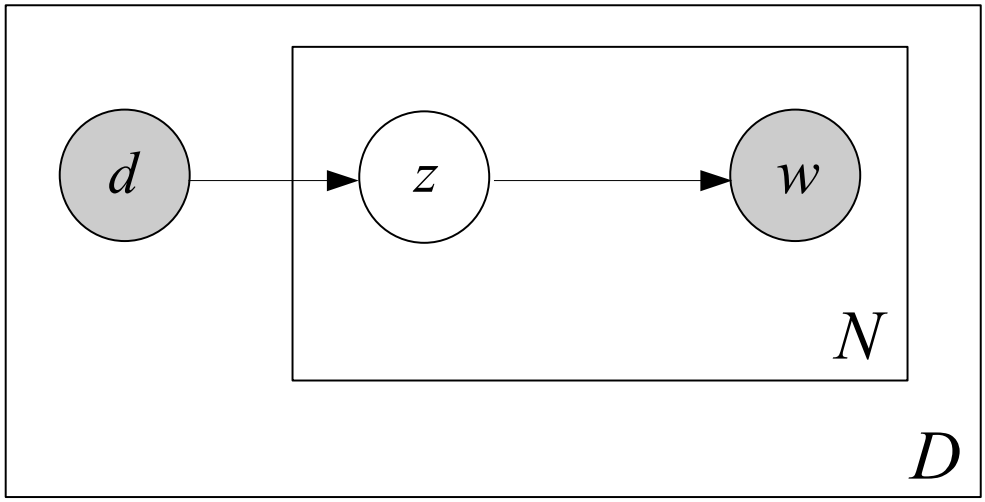


Figura 1.4: Modello grafico di PLSI

Tenendo conto che:

$$p(d|z) = \frac{p(d,z)}{p(z)} = \frac{p(z|d)p(d)}{p(z)},$$

con semplici calcoli si ottiene:

$$p(d,w) = p(d)\sum_{z\in\mathcal{Z}} p(z|d)p(w|z), \tag{1.7}$$

dove $p(z|d)p(w|d)$ è la forma che assume la probabilità che la parola w sia contenuta nel documento d, cioè $p(w|d) = p(z|d)p(w|d)$.
In Figura[7] 1.4 riportiamo la descrizione del processo generativo delle osservazioni (d,w) implicato da Eq. 1.7.
Questo processo risulta articolato nel seguente modo:

1. un documento d è scelto con probabilità $p(d)$,
2. una classe latente z è scelta con probabilità $p(z|d)$,
3. una parola w è *generata* con probabilità $p(w|z)$.

Si noti che per ottenere $p(w|d)$ occorre sommare rispetto a tutti i possibili valori di z che potrebbero aver generato l'osservazione w in d.
Il modello PLSI è dunque un modello *mistura* basato essenzialmente su due assunzioni di indipendenza. E cioè, le osservazioni (d,w) sono generate indipendentemente le une dalle altre (ciò che corrisponde all'assunzione implicita *bag of words*), e,

[7] I nodi bianchi rappresentano variabili osservate, quelli oscurati variabili latenti. I rettangoli indicano ripetizioni, per un numero di volte pari alla variabile posta nell'angolo. N è il numero di parole in un documento e D è il numero di documenti.

condizionatamente al valore della classe latente z, le parole w sono generate indipendentemente dallo specifico documento d: $p(d, w|z) = p(d|z)p(w|z)$. Oppure in altri termini, $p(w|z) = p(w|d, z)$, il che implica che i topic non dipendano dal valore dell'etichetta d e dunque siano comuni a tutti i documenti del corpus.
In definitiva, è la sola classe latente z a determinare il valore di w in d. Tenendo conto che il numero delle classi K è in genere molto inferiore a quello dei documenti, l'introduzione della variabile z semplifica notevolmente il calcolo di $p(w|d)$.
Diversamente dai modelli di *document clustering*, che si limitano ad assegnare ogni documento ad un singolo cluster, nel modello PLSI le distribuzioni $p(w|d)$, specifiche per documento, sono ottenute come combinazioni convesse dei *fattori*[8] $p(w|z)$, distribuzioni multinomiali sui termini del vocabolario del corpus. I documenti non vengono quindi assegnati a singoli cluster (argomenti); piuttosto risultano caratterizzati da una specifica mistura di fattori con coefficienti $p(z|d)$, i quali rendono possibile che un documento possa contenere una molteplicità di argomenti.
Tali coefficienti restituiscono quella descrizione ridotta dei documenti che è poi lo scopo dei modelli che stiamo esaminando.
Una volta definito il modello generativo in termini di $p(d)$, $p(z|d)$, e $p(w|z)$, la stima di queste quantità si ottiene massimizzando la funzione di log-verosimiglianza:

$$\mathcal{L} = \log \prod_d \prod_w p(d, w)^{n(d,w)} = \sum_d \sum_w n(d, w) \log p(d, w), \tag{1.8}$$

dove $n(d, w)$ è la frequenza del termine w nel documento d.
Come noto, l'algoritmo *Expectation Maximization* (EM) [Dempster et al. (1977), McLachlan & Krishnan (1997)] è una tecnica iterativa generale per trovare stime di massima verosimiglianza in modelli mistura a variabili latenti. Esso sfrutta infatti la presenza di quest'ultime per approssimare la soluzione di un problema altrimenti non risolvibile per via diretta.
In generale, l'algoritmo EM alterna due passi, 1) un passo E dove vengono calcolate le distribuzioni a posteriori delle variabili latenti in funzione dei valori correnti dei parametri e delle variabili osservate, 2) un passo M dove i parametri e le variabili osservate vengono aggiornate in funzione delle distribuzioni a posteriori ottenute al passo precedente.
Nel caso del modello PLSI, Hofmann (1999) dimostra che, utilizzando la formula di Bayes e l'assunzione di indipendenza condizionale $p(w|d, z) = p(w|z)$, il passo E assume la forma:

$$p(z|d, w) = \frac{p(z)p(d|z)p(w|z)}{\sum_{z'} p(z')p(d|z')p(w|z')} \tag{1.9}$$

[8] Più avanti, quando passeremo a descrivere il modello *Latent Dirichlet Allocation* (LDA) questi fattori prenderanno il nome di *topic*.

che è la probabilità che una parola w in un documento d sia sia spiegata dal fattore (topic) corrispondente a z.
E, con un pò di algebra, l'aggiornamento al passo M diventa:

$$p(w|z) = \frac{\sum_{d} n(d,w)p(z|d,w)}{\sum_{d,w'} n(d,w')p(z|d,w')} \tag{1.10}$$

$$p(d|z) = \frac{\sum_{w} n(d,w)p(z|d,w)}{\sum_{d',w} n(d',w)p(z|d',w)} \tag{1.11}$$

$$p(z) = \frac{\sum_{d,w} n(d,w)p(z|d,w)}{\sum_{d,w} n(d,w)} \tag{1.12}$$

Iterando i passi E e M, l'algoritmo converge a un minimo locale della log-verosimiglianza in Eq. 1.8. A partire dai valori in Eq. 1.10 corrispondenti a tale massimo è immediato il calcolo delle quantità di interesse, $p(w|z)$ e $p(z|d)$, ovvero i topic comuni a tutto il corpus e il loro peso relativo all'interno del documento d.
Hofmann (1999) propone anche una generalizzazione della procedura EM standard, chiamata *Tempered EM* (TEM), con l'obiettivo di evitare problemi di *overfitting* sull'insieme di addestramento. In sostanza, modifica il passo E introducendo un parametro di controllo β con il compito di mitigare l'influenza dell'insieme di addestramento nel determinare il valore delle stime. Il passo E generalizzato diventa infatti:

$$p_\beta(z|d,w) = \frac{p(z)[p(d|z)p(w|z)]^\beta}{\sum_{z'} p(z')[p(d|z')p(w|z')]^\beta} \tag{1.13}$$

dove, assegnando opportunamente a β un valore minore di 1, si riduce il peso della parte di verosimiglianza nella formula di Bayes.

PLSI *vs* LSI

La Figura 1.5 presa a prestito da Hofmann (1999) fornisce una rappresentazione geometrica molto interessante del modello PLSI, che risulta utile per chiarire la relazione tra quest'ultimo e il modello LSI.
I topic $p(w|z)$ sono rappresentati come punti del simplesso S_{T-1} di dimensione $T-1$ di tutte le possibili distribuzioni multinomiali sul vocabolario dei termini del corpus, $\mathcal{T}$ (di dimensione T). L'involucro convesso di questi K punti definisce un sottospazio

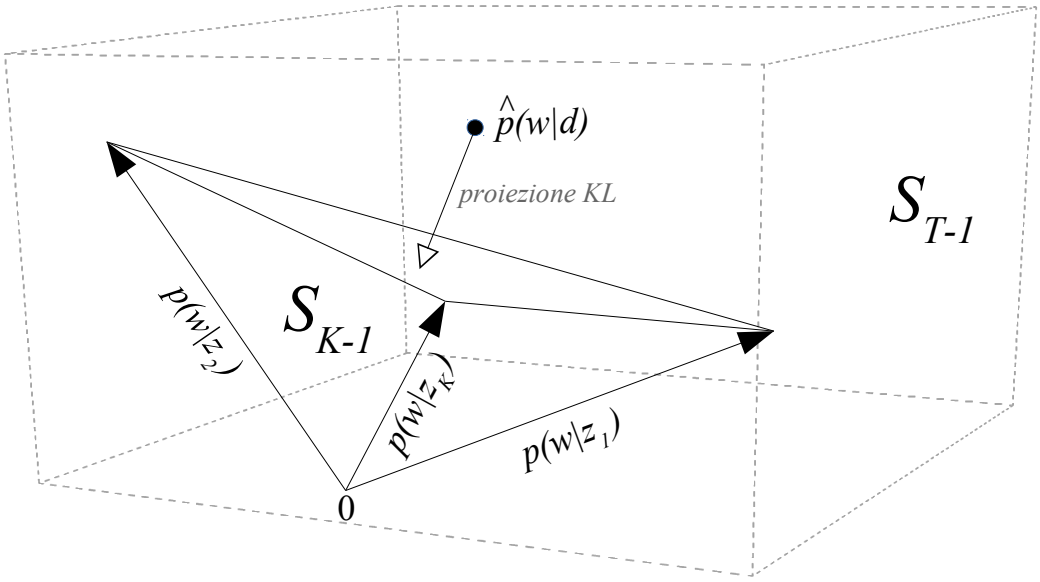


Figura 1.5: Un interpretazione geometrica del modello PLSI

S_{K-1} (ancora un simplesso) di dimensione $K-1$. L'assunzione del modello $p(w|d) = \sum_z p(z|d)p(w|z)$ esprime inoltre le distribuzioni multinomiali $p(w|d)$ come punti del simplesso S_{K-1} di coordinate $p(z|d)$, che sono poi le coordinate dei documenti nello spazio ridotto generato dai topic.
Il simplesso S_{K-1} prende il nome di *spazio semantico probabilistico latente* e ha una dimensione assai ridotta rispetto al simplesso completo S_{T-1} dal momento che tipicamente $K \ll T$.
Si osservi che tramite semplici calcoli è possibile scrivere la log-verosimiglianza in Eq. 1.8 come:

$$\mathcal{L} = \sum_d n(d) \left[\sum_w \frac{n(d,w)}{n(d)} \log p(w|d) + \log p(d) \right], \tag{1.14}$$

dove il primo termine tra parentesi quadre corrisponde alla divergenza (negativa) di Kullback-Leibler tra la distribuzione empirica delle parole in un documento $\hat{p}(w|d) = \frac{n(d,w)}{n(d)}$ e la corrispondente distribuzione come descritta dal modello, $p(w|d)$. In altri termini, massimizzare la log-verosimiglianza rispetto alle coordinate $p(z|d)$ e per topic $p(w|z)$ fissati, significa proiettare le osservazioni $\hat{p}(w|d)$ sul sottospazio generato dai topic in modo da minimizzare la divergenza di Kullback-Leibler.
Il modello PLSI è dunque dal punto di vista logico equivalente al modello LSI in quanto al pari di quest'ultimo cerca un sottospazio di dimensione ridotta[9] sul quale proiettare i dati (le distribuzioni empiriche $\hat{p}(w|d)$ nel caso del modello PLSI, diret-

[9]Mentre lo spazio ridotto del modello LSI è ottimale rispetto a proiezioni ortogonali, quello del modello PLSI lo è rispetto a proiezioni basate sulla divergenza di Kullback-Leibler.

tamente i vettori-riga della matrice DTM nel caso del modello LSI).
E tuttavia ne costituisce una generalizzazione in quanto i fattori (topic) e i loro pesi all'interno dei documenti possono vantare un'immediata interpretazione probabilistica che ne amplia le possibilità di utilizzo.

1.4 Il modello *Latent Dirichlet Allocation*

Il modello PLSI assume che il processo generativo delle osservazioni abbia la forma:

$$p(d, w) = p(d) \sum_z p(z|d)p(w|z),$$

la quale implica che i coefficienti della mistura, $p(z|d)$, siano indicizzati dal valore dell'etichetta d e dunque siano "appresi" dal modello sono relativamente a quei documenti su cui è stato addestrato.
Blei et al. (2003) osservano che per questa ragione PLSI non può essere considerato un modello generativo ben definito: soffre di scarsa capacità di generalizzazione, in quanto non propone una procedura naturale per assegnare una probabilità a documenti che siano fuori dall'insieme di addestramento.
Inoltre, lo stesso numero dei parametri da stimare cresce linearmente con il numero dei documenti utilizzati in fase di addestramento. I parametri di un modello PLSI con K fattori-topic sono infatti K distribuzioni multinomiali, ognuna di dimensione T, e D vettori dei coefficienti della mistura ognuno di dimensione K.
Pertanto $KT + KD$ parametri, una quantità che cresce linearmente con il numero di documenti D sui quali il modello viene addestrato.
Come ben noto (si confronti al riguardo Hastie et al. (2009)), un elevato numero di parametri da stimare può causare seri problemi di *overfitting*. Lo stesso Hofmann (1999), essendo pienamente consapevole di ciò, propone come possibile soluzione quella di adottare la versione temperata dell'algoritmo EM. Tuttavia, non basta. Popesul et al. (2001) dimostrano che anche in questo modo possono verificarsi con facilità problemi di overfitting.
Il modello *Latent Dirichelet Allocation* (LDA) in Blei et al. (2003) prova a superare queste difficoltà introducendo un processo di generazione anche per il vettore dei coefficienti $p(z|d)$, nello specifico una distribuzione di Dirichlet con parametro $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_K)$.
Si ottiene così un modello generativo ben definito, capace di generalizzare facilmente a nuovi documenti fuori dell'insieme di addestramento. Contrariamente a quanto accade per il modello PLSI, i $K + KT$ parametri del modello LDA (K per il vettore $\boldsymbol{\alpha}$ e KT per le distribuzioni multinomiali dei topic) non risultano infatti diretta-

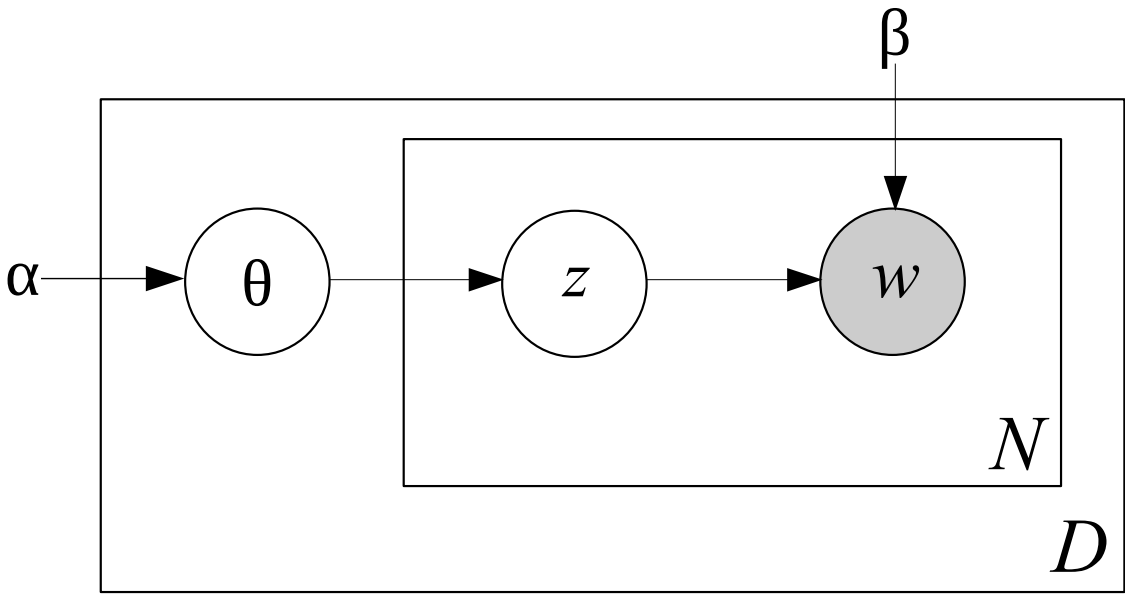


Figura 1.6: Modello grafico di LDA

mente collegati alla dimensione del corpus di addestramento e dunque non crescono linearmente con essa.

1.4.1 Specificazione del modello

Prima di passare a descrivere gli aspetti formali del modello LDA ridefiniamo le nozioni di *parola*, *documento* e *corpus* adattandole alla notazione che adotteremo da questo momento in poi (leggermente diversa da quella utilizzata finora), così come proposto in Blei et al. (2003).
Pertanto:

- Una *parola* w (word), l'unità di base delle osservazioni, è un elemento di un vocabolario di T termini, $\mathcal{T} = \{t_1, \ldots, t_j, \ldots, t_T\}$.
- Un documento è una sequenza di N parole $\mathbf{w} = (w_1, \ldots, w_n, \ldots, w_N)$[10].
- Un corpus è una collezione di D documenti $\mathcal{D} = \{\mathbf{w}_1, \ldots, \mathbf{w}_d, \ldots, \mathbf{w}_D\}$.

LDA è un modello probabilistico generativo del corpus $\mathcal{D}$ costruito intorno all'idea che i documenti siano misture *casuali*[11] di topic latenti, essendo un *topic* niente altro che una distribuzione di probabilità multinomiale sui termini del vocabolario $\mathcal{T}$.
Il modello grafico in Figura 1.6 descrive il processo di generazione delle parole dei

[10] Approccio *bag of words*.
[11] Qui sta la differenza fondamentale con PLSI.

documenti assunto da LDA. Rispetto a quello in Figura 1.4 esso si caratterizza principalmente per la presenza del nodo aleatorio (latente) denotato con θ.
Formalmente, si tratta di un modello Bayesiano gerarchico a tre livelli del quale proponiamo qui una lettura a partire dal livello più basso, quello delle osservazioni: le parole nei documenti.
Dunque:

Primo livello: le parole nel documento

Ognuna delle N parole w_n di un documento $\mathbf{w} \in \mathcal{D}$ è associata a una classe latente $z_n \in \mathcal{Z} = \{1, 2, \ldots, K\}$. Si ha:

$$w_n \sim p(w_n|z_n, \boldsymbol{\beta}) \equiv Mult(\boldsymbol{\beta}_{z_n}),$$

dove $\boldsymbol{\beta} = \{\beta_{kj}\}$ è la matrice $K \times T$ dei topic comuni a tutto il corpus, tale che $\beta_{kj} = p(t_j|z = k)$, e $Mult(\boldsymbol{\beta}_{z_n})$ è la distribuzione multinomiale su $\mathcal{T}$ corrispondente alla riga di $\boldsymbol{\beta}$ individuata dal valore di z_n. Si noti che la matrice $\boldsymbol{\beta}$ è considerata qui come un parametro da stimare.

Secondo livello: il documento

Ognuno dei valori z_n di un documento $\mathbf{w} \in \mathcal{D}$ è generato da una distribuzione multinomiale di parametro $\boldsymbol{\theta}$, cioè:

$$z_n \sim p(z_n|\boldsymbol{\theta}) \equiv Mult(\boldsymbol{\theta}).$$

Terzo livello: il corpus

Per ogni documento $\mathbf{w} \in \mathcal{D}$, il vettore $\boldsymbol{\theta}$ è generato da una distribuzione di Dirichlet di parametro $\boldsymbol{\alpha}$. In simboli:

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \equiv Dir(\boldsymbol{\alpha}).$$

Come la Figura 1.6 rende chiaro, nel processo di generazione del corpus i parametri $\boldsymbol{\alpha}$ e $\boldsymbol{\beta}$ sono estratti una sola volta per l'intero corpus, il vettore casuale (latente) $\boldsymbol{\theta}$ dei coefficienti della mistura dei topic una volta per ogni documento, e infine le variabili z_n e w_n una volta per ogni parola in ogni documento.
L'etichetta di classe z_n viene chiamata *topic assignment* e il generico elemento del vettore $\boldsymbol{\theta}$ *topic proportion*. Inoltre si assume che il valore di K, il numero dei topic, sia noto perchè fissato a priori.
Un vettore casuale $\boldsymbol{\theta}$ di dimensione K tale che $\theta_k \geq 0$ e $\sum_k \theta_k = 1$ ha distribuzione

di Dirichlet di parametro $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_K)$ quando:

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \theta_1^{\alpha_1 - 1} \cdots \theta_K^{\alpha_K - 1}, \tag{1.15}$$

dove $\alpha_k > 0$ e $\Gamma(\cdot)$ è la funzione Gamma.
Come noto, la distribuzione di Dirichlet è una conveniente distribuzione sul simplesso di dimensione $K-1$, che appartiene alla famiglia esponenziale, ha statistiche sufficienti di dimensione finita e soprattutto risulta coniugata alla distribuzione multinomiale. Tutte proprietà che facilitano le procedure di stima del modello.
Si osservi che l'assunzione $\boldsymbol{\theta} \sim Dir(\boldsymbol{\alpha})$ implica che le proporzioni dei topic all'interno di ogni documento siano generate indipendentemente dal documento stesso.
In questo modo, pagando il prezzo della stima di $\boldsymbol{\alpha}$, ci si libera dalla dipendenza dall'etichetta del documento (e quindi dalla dipendenza dall'insieme di addestramento). Lo ripetiamo, qui sta la differenza fondamentale tra l'approccio PLSI e quello LDA.
Dati i parametri $\boldsymbol{\alpha}$ e $\boldsymbol{\beta}$, la distribuzione congiunta sulle variabili latenti e osservate per il generico documento $\mathbf{w}_d \in \mathcal{D}$ prende la forma:

$$p(\boldsymbol{\theta}_d, \mathbf{z}_d, \mathbf{w}_d|\boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \prod_{n=1}^{N_d} p(z_{dn}|\boldsymbol{\theta}_d) p(w_{dn}|z_{dn}, \boldsymbol{\beta}), \tag{1.16}$$

dove $p(z_{dn}|\boldsymbol{\theta}_d)$ è semplicemente pari a θ_{dk} quando $z_{dn} = k$.
Marginalizzando rispetto alle variabili latenti $\boldsymbol{\theta}_d$ e $\mathbf{z}_d$ otteniamo:

$$p(\mathbf{w}_d|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\boldsymbol{\theta}_d) p(w_{dn}|z_{dn}, \boldsymbol{\beta}) \right) d\boldsymbol{\theta}_d, \tag{1.17}$$

ovvero la distribuzione marginale del documento $\mathbf{w}_d \in \mathcal{D}$.
Sfruttando infine l'assunzione implicita di indipendenza tra i documenti, otteniamo la probabilità del corpus $\mathcal{D}$ che stiamo cercando:

$$p(\mathcal{D}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{d=1}^{D} p(\mathbf{w}_d|\boldsymbol{\alpha}, \boldsymbol{\beta}). \tag{1.18}$$

A partire da questa probabilità si costruisce la funzione di log-verosimiglianza da massimizzare rispetto ai parametri incogniti $\boldsymbol{\alpha}$ e $\boldsymbol{\beta}$. Alla soluzione di questo problema di stima dedichiamo la sezione che segue.

1.4.2 Inferenza a posteriori e stima dei parametri: VEM

Dato un corpus di documenti $\mathcal{D} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_D\}$, si vogliono trovare i valori dei parametri $\boldsymbol{\alpha}$ e $\boldsymbol{\beta}$ che risolvono il seguente problema:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \ell(\boldsymbol{\alpha}, \boldsymbol{\beta}), \tag{1.19}$$

dove $\ell(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{d=1}^{D} \log p(\mathbf{w}_d|\boldsymbol{\alpha}, \boldsymbol{\beta})$ è la funzione di log-verosimiglianza dei dati ottenuta a partire dalla Eq. 1.18.

Sfortunatamente, come sottolineato in Blei et al. (2003), questo problema non può essere risolto direttamente in quanto la distribuzione $p(\mathbf{w}_d|\boldsymbol{\alpha}, \boldsymbol{\beta})$ è computazionalmente intrattabile.

Tuttavia, similmente al caso PLSI, ci può venire in soccorso l'algoritmo EM. In verità, una sua variante *variazionale* imposta dal fatto che occorre approssimare anche la distribuzione a posteriori sulle variabili latenti $p(\boldsymbol{\theta}_d, \mathbf{z}_d|\mathbf{w}_d, \boldsymbol{\alpha}, \boldsymbol{\beta})$ dal momento che anch'essa dipende da $p(\mathbf{w}_d|\boldsymbol{\alpha}, \boldsymbol{\beta})$, essendo:

$$p(\boldsymbol{\theta}_d, \mathbf{z}_d|\mathbf{w}_d, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{p(\boldsymbol{\theta}_d, \mathbf{z}_d, \mathbf{w}_d|\boldsymbol{\alpha}, \boldsymbol{\beta})}{p(\mathbf{w}_d|\boldsymbol{\alpha}, \boldsymbol{\beta})}. \tag{1.20}$$

La differenza fondamentale tra gli algoritmi EM e VEM (*Variational Expectation-Maximization*) per LDA sta nel passo E. Nell'approccio VEM, alla distribuzione a posteriori sulle variabili latenti $p(\boldsymbol{\theta}_d, \mathbf{z}_d|\mathbf{w}_d, \boldsymbol{\alpha}, \boldsymbol{\beta})$ si sostituisce una sua approssimazione trattabile.

L'idea di base infatti è quella di approssimare la distribuzione a posteriori *vera* con una distribuzione più semplice, scelta tra i membri di una famiglia (parametrizzata) di questo tipo:

$$q(\boldsymbol{\theta}_d, \mathbf{z}_d|\boldsymbol{\gamma}_d, \boldsymbol{\phi}_d) = q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d) \prod_{n=1}^{N} q(z_{dn}|\boldsymbol{\phi}_{dn}), \tag{1.21}$$

dove $\boldsymbol{\phi}_d = (\boldsymbol{\phi}_{d1}, \boldsymbol{\phi}_{d2}, \ldots, \boldsymbol{\phi}_{dN})$, $q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d)$ è una Dirichlet di parametro $\boldsymbol{\gamma}_d$ e $q(z_{dn}|\boldsymbol{\phi}_{dn})$ è una multinomiale di parametro $\boldsymbol{\phi}_{dn}$.

Le distribuzioni in Eq. 1.21 prendono il nome di distribuzioni *variazionali* e si basano sull'assunzione che le variabili latenti $\boldsymbol{\theta}_d$ e $\mathbf{z}_d$ siano indipendenti condizionatamente ai valori dei parametri (variazionali) $\boldsymbol{\gamma}_d$ e $\boldsymbol{\phi}_d$.

Chiaramente il problema di inferenza diventa ora quello di determinare i valori ottimali di tali parametri, $(\boldsymbol{\gamma}_d^*, \boldsymbol{\phi}_d^*)$.

Questo problema viene risolto minimizzando la divergenza di Kullback-Leibler tra la distribuzione variazionale e la distribuzione a posteriori vera.

Ovvero:

$$\min_{(\boldsymbol{\gamma}_d,\boldsymbol{\phi}_d)} D(q(\boldsymbol{\theta}_d, \mathbf{z}_d|\boldsymbol{\gamma}_d, \boldsymbol{\phi}_d) \,\|\, p(\boldsymbol{\theta}_d, \mathbf{z}_d|\mathbf{w}_d, \boldsymbol{\alpha}, \boldsymbol{\beta})), \tag{1.22}$$

dove $D(\cdot)$ è la divergenza di Kullback-Leibler.
L'approccio variazionale trasforma dunque il problema della ricerca della distribuzione a posteriori sulle variabili latenti (e cioè sostanzialmente il passo E dell'algoritmo EM) in un problema di ottimizzazione.
Ciò detto, l'algoritmo VEM per LDA risulta una procedura iterativa che alterna i seguenti passi:

- **E-step**: Per ogni documento $d \in \mathcal{D}$ si trovano i valori ottimali dei parametri variazionali $(\boldsymbol{\gamma}_d^*, \boldsymbol{\phi}_d^*)$ come in Eq. 1.22, ottenendo una funzione di $\boldsymbol{\alpha}$ e $\boldsymbol{\beta}$ che è un *lower-bound* della log-verosimiglianza $\ell(\boldsymbol{\alpha}, \boldsymbol{\beta})$;

- **M-step**: si massimizza rispetto ad $\boldsymbol{\alpha}$ e $\boldsymbol{\beta}$ il lower-bound trovato al passo precedente.

Questi due passi sono ripetuti fintantoché l'algoritmo converga ad un massimo locale della funzione di log-verosimiglianza $\ell(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

1.5 Un'estensione bayesiana del modello LDA

Per ragioni che saranno chiare alla fine di questo paragrafo, nelle nostre sperimentazioni abbiamo utilizzato un'estensione bayesiana del modello LDA appena visto. Stiamo parlando del modello LDA esteso discusso in Griffiths & Steyvers (2004).
Tale modello assume che la distribuzione di Dirichlet sui vettori $\boldsymbol{\theta}$ sia simmetrica, ovvero $\alpha_1 = \alpha_2 = \ldots = \alpha_K = \alpha$, e inoltre che una distribuzione a priori sia posta anche sul parametro $\boldsymbol{\beta}$, la matrice $\{\beta_{kj} = p(t_j|z = k)\}$ delle distribuzioni multinomiali sui termini del vocabolario. Una distribuzione ancora di Dirichlet simmetrica, che indicheremo con il simbolo[12] $Dir(\beta)$.
Conseguentemente d'ora in avanti useremo il simbolo $\boldsymbol{\phi}_k$ per indicare la distribuzione multinomiale associata al k-esimo topic, cioè $\phi_{kj} = p(t_j|z = k)$.
Diversamente da quanto accadeva nel paragrafo precedente, i parametri α e β devono essere trattati qui come costanti, tipicamente fissate pari a valori inferiori ad 1. Questo perchè per tali valori si favorisce la concentrazione della massa di probabilità rispettivamente su (relativamente) pochi topic per documento e su (relativamente) pochi termini per topic.
Steyvers & Griffiths (2006) suggeriscono di usare i valori $\alpha = \frac{50}{K}$, dove K è il numero dei topic, e $\beta = 0.01$.

[12] Si faccia attenzione al fatto che ora β in $Dir(\beta)$ indica il parametro di concentrazione della distribuzione.

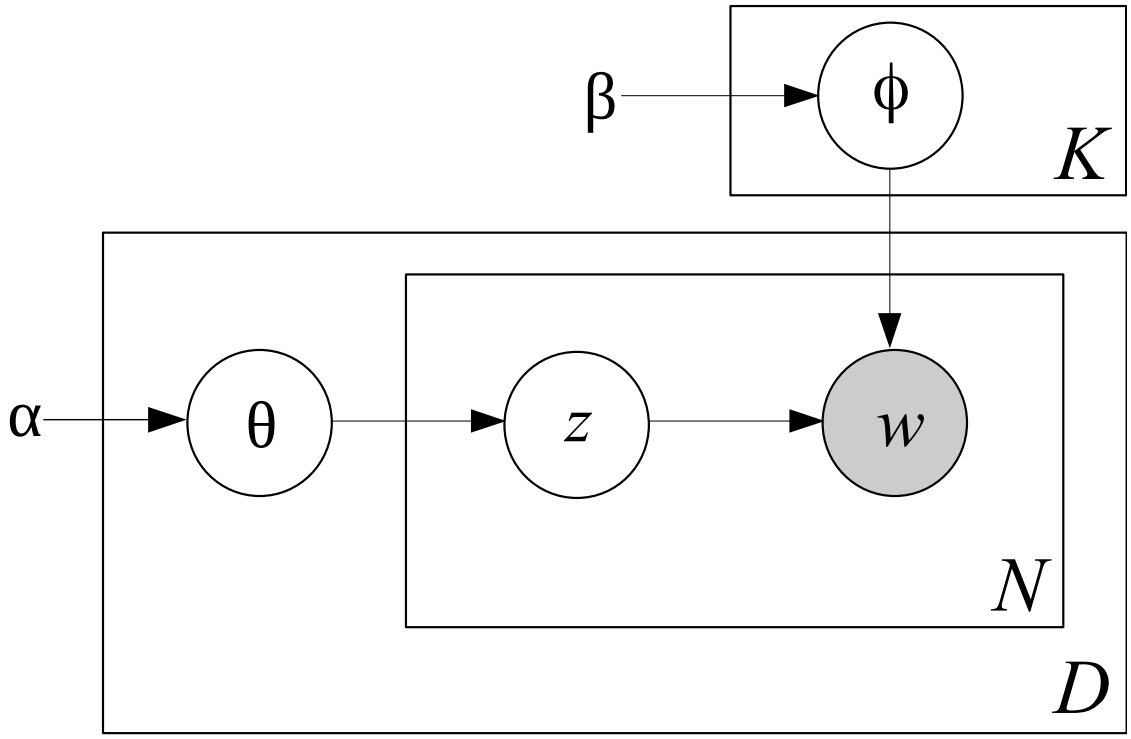


Figura 1.7: Modello grafico di LDA (esteso)

Il modello grafico in Figura 1.7 chiarisce il ruolo di tutte le variabili, latenti e osservate, coinvolte in questa versione estesa del modello LDA. A titolo puramente esemplificativo inoltre, la Figura 1.8 (adattata da una simile in Steyvers & Griffiths (2006)) illustra la sua natura *duale*, come modello generativo e come problema di inferenza statistica a posteriori.

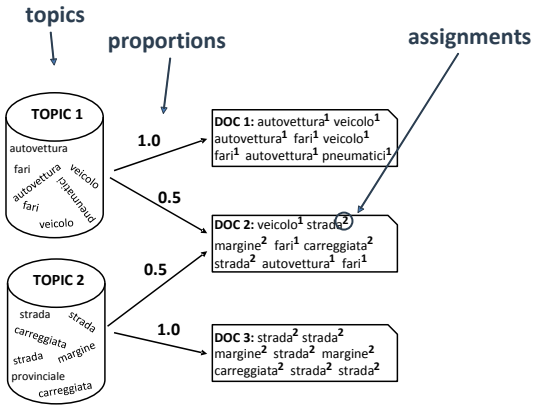
Come è evidente, le variabili principali del modello sono le distribuzioni (latenti) $\boldsymbol{\phi}_k$ per ogni topic e $\boldsymbol{\theta}_d$ per ogni documento. Per arrivare alla stima di queste distribuzioni Griffiths & Steyvers (2004) propongono di partire dalla stima della distribuzione a posteriori sulle variabili $\mathbf{z}$, le etichette dei topic assegnate a ogni parola nei documenti.

Per semplicità di esposizione, nel seguito chiameremo *token* l'occorrenza di una parola in un documento e *topic assignment* i singoli valori delle etichette $\mathbf{z}$. Indicheremo inoltre con i l'indice di ogni token.

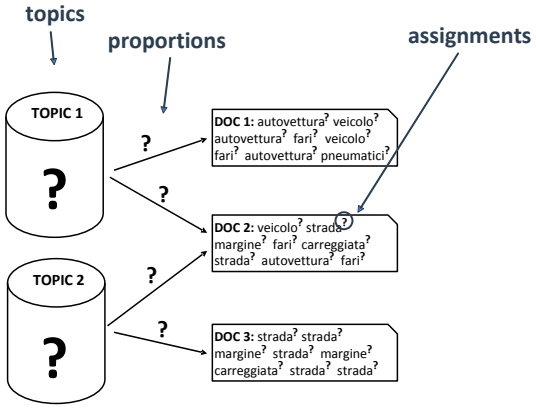
Ogni topic assignment z_i può assumere un valore in $\{1, 2, \ldots, K\}$ e deve essere valutato per ogni token. Ne segue che la stima della distribuzione a posteriori su $\mathbf{z}$ non può che essere affidata a procedure molto efficienti se si vuole adottare questo approccio in situazioni pratiche[13].

Griffiths & Steyvers (2004) propongono una procedura basata su *Gibbs sampling*,

[13]Solo per fare un esempio, si consideri che il nostro corpus, che pur è di media dimensione, conta più di 52 milioni di token.



(a) Modello generativo. Si noti che le etichette associate con le parole (token) dei documenti indicano quale topic ha generato quella parola.



(b) Inferenza a posteriori.

Figura 1.8: LDA come modello generativo e come problema di inferenza statistica

relativamente facile da implementare e che fornisce un metodo efficiente per estrarre topic da vaste collezioni di documenti di testo.

Gibbs sampling (GS) è una forma di *Marchov Chain Monte Carlo* (MCMC), un insieme di tecniche iterative approssimate aventi lo scopo di campionare valori da distribuzioni di probabilità complesse (spesso di elevata dimensione) per poi approssimarle con la loro distribuzione empirica.

GS[14] consiste nel costruire una catena di Markov (una sequenza di variabili casuali ognuna dipendente dalla precedente) la cui distribuzione limite è proprio la distribuzione che si vuole approssimare. In particolare, GS simula una distribuzione obiettivo con un numero elevato di variabili campionando da sottoinsiemi di que-

[14]Per maggiori approfondimenti sul *Gibbs sampling* rimandiamo a Bishop (2007).

st'ultime, dove ogni sottoinsieme è condizionato a tutti gli altri. Il campionamento è sequenziale e continua finché i valori campionati non approssimano la distribuzione obiettivo.
Nel nostro caso la distribuzione obiettivo è appunto la distribuzione a posteriori sulle etichette latenti $\mathbf{z}$, a partire dalla quale ottenere successivamente le stime di $\boldsymbol{\phi}_k$ e di $\boldsymbol{\theta}_d$.

1.5.1 Inferenza a posteriori: Gibbs sampling

Sia i l'indice del singolo token, t_i e d_i rispettivamente il termine del vocabolario osservato in corrispondenza del token i e il documento in cui è presente il token i.
L'algoritmo che stiamo descrivendo considera un token alla volta e stima la probabilità di assegnare il token corrente a ognuno dei K topic, condizionata alle etichette dei topic di tutti gli altri token. Da questa distribuzione condizionata viene poi estratto il nuovo topic da assegnare al token corrente.
Più nel dettaglio, tale distribuzione può essere scritta come:

$$p(z_i = k|\mathbf{z}_{-i}, t_i, d_i, \cdot), \tag{1.23}$$

dove $z_i = k$ indica che il token i è assegnato al topic k, $\mathbf{z}_{-i}$ si riferisce alle etichette dei topic di tutti gli altri token e "$\cdot$" indica tutte le altre informazioni osservate o conosciute come tutti gli altri termini $\mathbf{t}_{-i}$, tutti gli altri documenti $\mathbf{d}_{-i}$ e i valori degli iperparametri α e β.
Griffiths & Steyvers (2004) dimostrano che:

$$p(z_i = k|\mathbf{z}_{-i}, t_i, d_i, \cdot) \propto \frac{C^{TK}_{t_i k} + \beta}{\sum_{t=1}^{T} C^{TK}_{tk} + T\beta} \frac{C^{DK}_{d_i k} + \alpha}{\sum_{k=1}^{K} C^{DK}_{d_i k} + K\alpha}, \tag{1.24}$$

dove T è la dimensione del vocabolario, D è il numero totale dei documenti, K è il numero fissato dei topic e C^{TK} e C^{DK} sono matrici di conteggio di dimensione $T \times K$ e $D \times K$ rispettivamente.
$C^{TK}_{t_i k}$ è il numero di volte che il termine t_i viene assegnato al topic k (e quindi β può essere interpretato come l'equivalente a priori) e $C^{DK}_{d_i k}$ è il numero di volte che un token del documento d viene assegnato al topic k (idem per α).
Chiaramente la parte sinistra del secondo membro in Eq.1.24 è una stima della probabilità del termine t_i sotto il topic k, ϕ_{kt_i}, mentre la parte destra è una stima della probabilità del topic k nel documento d, θ_{dk}.
In questo modo, i token di un documento vengono assegnati ai topic in base a quanto frequentemente i termini corrispondenti sono assegnati a un topic (su tutto il

corpus) e a quanto questo topic è dominante nel documento.
L'algoritmo comincia dall'assegnare le etichette dei topic ai token a caso. Quindi, per ogni token viene estratta una nuova etichetta dalla distribuzione in Eq.1.24 e aggiornate le matrici di conteggio. Una sola iterazione del Gibbs sampling consiste nell'attribuzione delle etichette dei topic a tutti i token del corpus.
Le etichette assegnate nel corso delle prime iterazioni vengono rigettate in quanto costituiscono stime molto povere, risentendo eccessivamente delle condizioni di partenza. Dopo un periodo iniziale opportunamente fissato (burn-in period), le iterazioni successive cominciano ad approssimare la distribuzione a posteriori. A questo punto, scegliendone un certo numero ad intervalli regolari per prevenire fenomeni di correlazione, si ottiene un campione con il quale costruire la distribuzione empirica che stima quella a posteriori.
In definitiva, al termine del numero massimo di iterazioni fissato si dispone di un'etichetta di topic per ogni parola di ogni documento del corpus. Nelle applicazioni pratiche tuttavia si lavora sulle stime delle distribuzioni $\boldsymbol{\phi}_k$ e $\boldsymbol{\theta}_d$, che, lo ricordiamo ancora una volta, modellano rispettivamente i topic comuni a tutto il corpus e il loro peso all'interno di ogni documento.
Da quanto detto, non dovrebbe sorprendere che tali stime si possono ottenere in questo modo:

$$\hat{\phi}_{kj} = \frac{C^{TK}_{t_j k} + \beta}{\sum_{t=1}^{T} C^{TK}_{tk} + T\beta}, \tag{1.25}$$

$$\hat{\theta}_{dk} = \frac{C^{DK}_{dk} + \alpha}{\sum_{k=1}^{K} C^{DK}_{dk} + K\alpha}. \tag{1.26}$$

Griffiths & Steyvers (2004) dimostrano che questi valori corrispondono a medie a posteriori condizionate ai valori delle etichette $\mathbf{z}$.

In conclusione, osserviamo che nei nostri esperimenti abbiamo provato tanto l'approccio VEM di Blei et al. (2003) quanto quello GS di Griffiths & Steyvers (2004). Il primo implementato nel pacchetto `topicmodels` in `R` [Grün & Hornik (2011)] e il secondo in MALLET[15] una suite Java per l'analisi automatica dei testi sviluppata dall'Università del Massachusetts-Amherst.
In entrambi i casi, la complessità computazionale dipende essenzialmente dal numero delle parole del corpus (token). Nel caso VEM è determinata dal numero di operazioni che servono per il passo E ($O(N^2K)$ per ogni documento); nel caso GS

[15]MALLET: MAchine Learning for LanguagE Toolkit (http://mallet.cs.umass.edu).

da quante volte si decide di iterare la procedura di attribuzione delle etichette ai token. Tipicamente l'algoritmo VEM richiede meno iterazioni dell'algoritmo GS ma ogni sua iterazione è molto più costosa [Yao et al. (2009)].
Per quanto ci riguarda, fin quando ci è stato possibile confrontare i risultati ottenuti, e cioè per valori bassi di K (non oltre 100 topic), i due algoritmi hanno manifestato un comportamento molto simile sia in termini di prestazioni che di qualità dei topic estratti.
Per valori di $K \geq 100$, l'algoritmo VEM ha smesso di funzionare mentre quello GS ha continuato a farlo anche per valori molto superiori (fino a $K = 4.000$ topic). Ciò è dipeso probabilmente più dall'implementazione di GS fornita da MALLET, molto più efficente rispetto a quella VEM messa a disposizione dal pacchetto `topicmodels`, che da limitazioni intrinseche di quest'ultimo approccio.
Questa è tuttavia la sola vera ragione che ha motivato la nostra scelta a favore dell'approccio GS rispetto a quello VEM.

Capitolo 2

Gli scenari di sperimentazione: la DTM base

In questo capitolo illustreremo il primo dei due scenari di sperimentazione sui quali abbiamo concentrato i nostri sforzi. Lo abbiamo chiamato "DTM base" visto che tutti gli esperimenti sono stati eseguiti a partire da una matrice documenti per termini *di base*, costruita cioè senza operare alcun pre-trattamento dei testi dei documenti oltre la rimozione dei termini tematicamente superflui (stop-words) e dei caratteri numerici.
Dopo aver descritto le caratteristiche del corpus delle sentenze a nostra disposizione, analizzeremo quali conseguenze derivino dalla scelta di una matrice DTM siffatta. Ogni sentenza, infatti, è per sua natura caratterizzata da un profilo legale (*motivi di diritto*) e da un profilo fattuale (*motivi di fatto*). Nel nostro corpus il lessico tipico del primo profilo è allo stesso tempo più omogeneo e più diffuso di quanto non sia quello, per forza di cose più eterogeneo, del secondo profilo. Ne segue che la matrice DTM base risulta essere inevitabilmente destinata a riflettere al suo interno la preponderanza del lessico procedurale-giuridico su quello relativo alle situazioni di fatto richiamate nelle sentenze. Ciò che produce conseguenze significative sull'effettiva interpretabilità dei topic estratti dai documenti soprattutto se, come lo siamo noi, si è interessati più ai "fatti" delle sentenze che non al "diritto" in esse applicato.
Passeremo infine a trattare il problema della selezione del modello, ovvero della scelta del numero dei topic K. Due sono i metodi che abbiamo preso in considerazione nei nostri esperimenti, uno indiretto basato sulla misura della performance rispetto ad un compito assegnato, nello specifico la classificazione dei documenti, e l'altro diretto basato sulla misura della capacità di generalizzazione del modello, ovvero la capacità di predire documenti *nuovi* precedentemente non visti.

```
<snintegrale tipoprov=" " anno=" " numdec=" " datdec=" " nsz=" " szdec=" ">
<col>
 ... ... ...
 ... ... ...
</col>
<tes>
 ... ... ...
 ... ... ...
</tes>
<sic timestamp=" " nrg=" " anno_nrg=" " materia=" " autorita=" " localita=" "
anno_prov=" " data_ud=" " datdep=" " num_provv=" " >
<coll CON=" " <coll REL=" "> <coll PRE=" "> <coll PM=" ">
<parte ric=" "> <parte int=" ">
</sic>
</snintegrale>
```

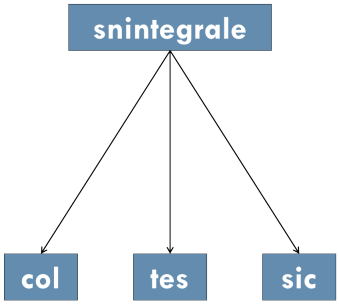


Figura 2.1: Albero XML delle sentenze

2.1 Il corpus

Oggetto della nostra analisi è l'insieme dei documenti che ci è stato messo a disposizione (non senza qualche difficoltà) dal Centro di Elaborazione Documentale della Corte di Cassazione (CED). Esso contiene 74.858 sentenze in materia civile pubblicate nel quinquennio 2010-2014. Costituisce evidentemente una frazione ridotta dell'archivio ItalGiure ma sufficiente per gli scopi di questo lavoro.
La Figura 2.1 illustra il formato originale di tali documenti, ovvero un albero XML con nodo radice *snintegrale* (sentenza integrale) e nodi foglia *col*, *tes* e *sic*, contenenti rispettivamente i riferimenti dei membri del collegio giudicante, il testo della sentenza, e una serie di informazioni a corredo estratte dal registro informatico[1] della Corte, di cui la Tabella 2.1 riporta una descrizione sommaria.
Un processo interamente sviluppato in ambiente `R` provvede a trasformare i dati iniziali nel *corpus* documentale che costituirà la base delle elaborazioni future. E il pacchetto `Supreme` appositamente realizzato[2] rende disponibili tutte le funzioni che implementano le varie fasi di questo processo.
All'inizio del processo, la funzione `xml2dfciv` legge l'albero XML delle sentenze e lo trasforma in un `dataframe` di classe `dfciv`, dove cioè ogni riga corrisponde a un documento e ogni colonna a un suo attributo (in questa logica, anche il testo è un attributo). A partire dal dataframe così ottenuto, consistente di 74.858 righe e 15 colonne, la funzione `dfciv2corpus` genera un `corpus` completo di metadati, sfruttando l'infrastruttura di text mining in `R` fornita dal pacchetto `tm` (disponibile in CRAN) descritto in Feinerer et al. (2008). Al termine del processo, la funzione `corpus2dtm` crea la matrice documenti per termini (**D**ocument **T**erm **M**atrix). Ad

[1] Sistema Informativo della Cassazione (SIC).
[2] `Supreme` può essere scaricato da `github.com/paolofantini/Supreme`.

Colonna	Descrizione
Id_doc	ID del documento
tipoProv	tipo della decisione
annoDec	anno della decisione
numDec	numero della decisione
numSez	numero della sezione
testo	testo della decisione
dispositivo	dispositivo della decisione
annoNrgSic	anno di iscrizione del ricorso
nrgSic	numero del ricorso
annoProvOrig	anno della decisione appellata
numProvOrig	numero della decisione appellata
autorita	autorità della decisione appellata
localita	località della decisione appellata
materia	materia della decisione appellata
Id_materia	ID della materia della decisione appellata

Tabella 2.1: Colonne del dataframe di classe `dfciv` (metadati)

essa sarà dedicato il paragrafo che segue. Si osservi che la matrice DTM costituisce tipicamente l'input del modello LDA.
Per completezza e perchè queste informazioni ci saranno utili più avanti, le Tabelle 2.2 e 2.3 riportano le distribuzioni delle sentenze per materia (20 classi di materia) e per anno di pubblicazione.

2.2 La matrice documenti per termini (DTM)

Seguendo Sebastiani (2002) chiameremo *indicizzazione* del documento l'attività di rappresentare in forma compatta il suo contenuto testuale.
In generale essa dipende a) dal modo in cui si definiscono le *unità* significative di testo, b) dalle regole in base alle quali queste ultime si combinano. Il primo viene definito un problema di *semantica lessicale* e il secondo uno di *semantica composizionale*.
Nel seguito ci occuperemo esclusivamente di semantica lessicale e identificheremo le unità significative di testo con le singole parole (*words*) del documento, senza fare alcun riferimento alle loro combinazioni grammaticali e/o sintattiche.
In letteratura questo approccio è chiamato *bag of words*: ogni documento d_i è rappresentato come un vettore di pesi $d_i = (w_{i1}, w_{i2}, \ldots, w_{i|\mathcal{T}|})$, dove $\mathcal{T}$ è il vocabolario dei termini (*terms*), ovvero la lista dei termini ricorrenti almeno una volta nel corpus

Materia	ID	Freq	Materia	ID	Freq
Tributi	148	16.375	Espropriazione	53	1.377
Lavoro	76	16.217	Sanzioni amministrative	129	1.343
Contratti	30	10.579	Esecuzione forzata	51	1.128
Equa riparazione	50	8.225	Appalto	9	748
Responsabilità civile	114	4.195	Comunione e condominio	25	653
Previdenza	96	3.812	Banca e borsa	18	624
Diritti reali	37	2.284	Ricorsi contro giudici speciali	123	526
Fallimento	55	2.247	Possesso	93	521
Vendita, Permuta, Riporto	155	1.624	Edilizia e urbanistica	46	501
Famiglia	56	1.401	Procedura civile	101	478

Tabella 2.2: Distribuzione delle sentenze per classe di materia

Anno	Freq
2010	13.878
2011	13.834
2012	13.707
2013	17.321
2014	16.118
	74.858

Tabella 2.3: Distribuzione delle sentenze per anno di pubblicazione

esaminato, e w_{ij} misura il contributo del termine t_j alla descrizione della contenuto del documento d_i.
Il principale vantaggio di questo approccio consiste nella *vettorizzazione* dei documenti, ovvero nella riduzione di documenti di lunghezza arbitraria a vettori di pesi numerici di lunghezza fissata.
Il caso più semplice è quello in cui i pesi corrispondano alla frequenza dei termini all'interno dei documenti. Tuttavia, similmente a quanto avviene nelle applicazioni di *information retrieval*, considereremo anche il caso di pesi calcolati tramite la funzione *tfidf*[3] in Salton & Bucley (1988), definita come:

$$tfidf(d_i, t_j) = n_{ij} \cdot \log \frac{|\mathcal{D}|}{n_j} \tag{2.1}$$

dove $\mathcal{D}$ è l'insieme dei documenti del corpus, n_{ij} è la frequenza relativa del termine t_j nel documento d_i e n_j è il numero di documenti che contengono il termine t_j. Ne segue che la *rilevanza* di un termine per un documento è direttamente proporzionale alla sua frequenza in quel documento e inversamente proporzionale alla sua

[3] *tfidf* sta per term frequency-inverse document frequency.

Documenti	74.858
Terms	250.270
Words	**52.540.175**
Lunghezza media (terms)	407
Lunghezza media (words)	702
Lunghezza mediana (terms)	362
Lunghezza mediana (words)	573
Deviazione standard (terms)	212
Deviazione standard (words)	516

Tabella 2.4: Statistiche della DTM base

frequenza nell'intero corpus. Il tentativo è chiaramente quello di individuare i migliori candidati a rappresentare il contenuto dei singoli documenti.
Nella pratica si ricorre spesso a forme di normalizzazione per ottenere pesi compresi nell'intervallo $[0, 1]$. La più diffusa delle quali è quella di tipo coseno:

$$w_{ij} = \frac{tfidf(d_i, t_j)}{\sqrt{\sum_{j=1}^{|\mathcal{T}|} tfidf^2(d_i, t_j)}} \tag{2.2}$$

La matrice di indicizzazione $DTM = \{w_{ij}\}$ costruita a partire dai documenti-vettore d_i prende il nome di matrice documenti per termini (**D**ocument **T**erm **M**atrix). Con pesi w_{ij} posti pari alla frequenza dei termini nei documenti, costituisce l'input del modello LDA.

2.2.1 La DTM base

Nei nostri esperimenti abbiamo considerato due diversi scenari, indicati rispettivamente come "DTM base" e "DTM ridotta".
Nel primo scenario, una sequenza di filtri tipicamente utilizzati in ambito *information retrieval* viene applicata ai testi dei documenti allo scopo di ridurre la dimensione del vocabolario, cioè il numero di colonne della matrice DTM.
Il più importante tra questi filtri consiste nella rimozione dei termini superflui, quelli cioè che non apportano alcun contributo tematico ai documenti come ad esempio gli avverbi, le congiunzioni, i verbi ausiliari, e in generale tutti i termini contenuti in una lista di *stop-words* fornita al riguardo. Seguono la riduzione delle maiuscole a minuscole, la rimozione della punteggiatura e quella dei caratteri numerici. Raramente, e noi non lo abbiamo fatto perchè sperimentalmente non ne abbiamo ravvisato i vantaggi, si ricorre anche alla lemmatizzazione (stemming), ovvero alla riduzione dei termini alla loro radice fondamentale (lemma).

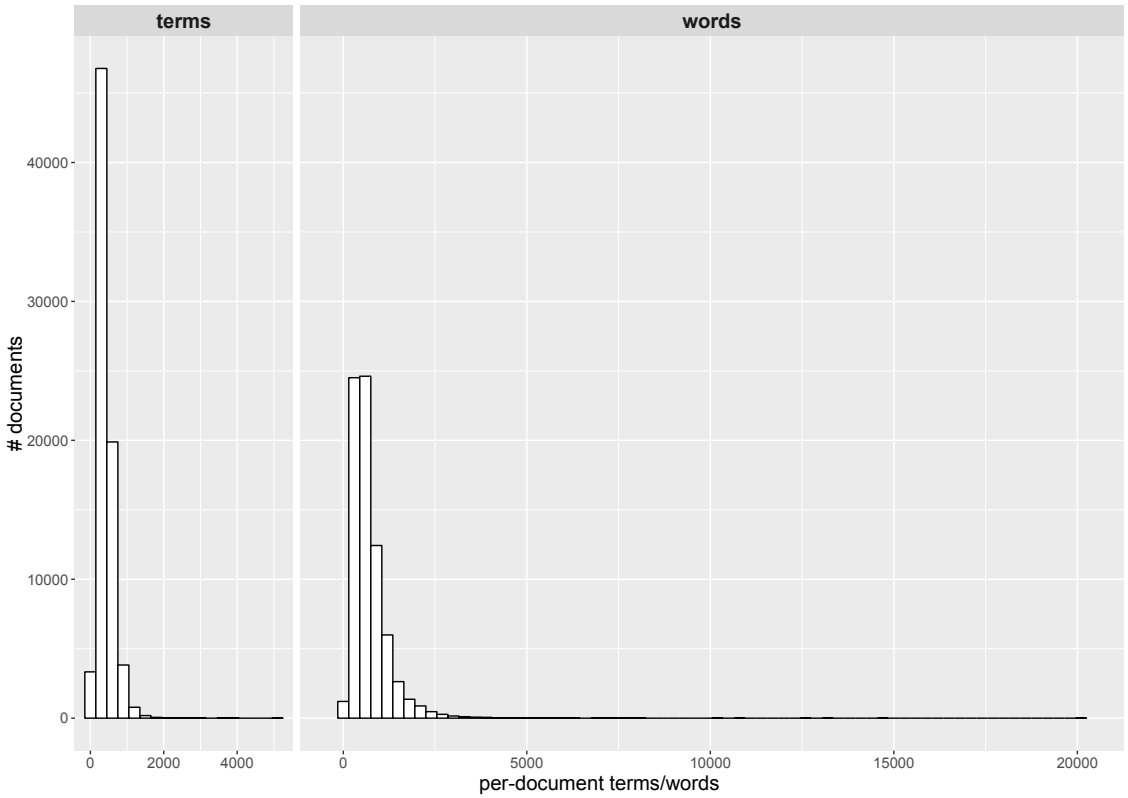


Figura 2.2: Distribuzioni dei documenti per numero di *terms* e *words* (DTM base)

Nel secondo scenario, al quale dedicheremo il prossimo capitolo, la matrice DTM è ottenuta a partire da quella di base, applicando ad essa una ulteriore riduzione dei termini-colonna basata sui punteggi tfidf.
La Tabella 2.4 riporta alcune statistiche generali della DTM base ad integrazione degli istogrammi in Figura 2.2. Da notare in particolare il valore assai elevato del numero di *words*, le occorrenze di tutte le parole, ciò da cui dipenderà in sostanza l'efficienza computazionale dei modelli impiegati[4].
Qualche riflessione merita la questione della rimozione dei caratteri numerici. Nel lessico giuridico, infatti, i *numeri* hanno un'importanza cruciale. Solo per fare qualche esempio, identificano leggi, articoli di legge e commi. Costituiscono cioè lo strumento privilegiato attraverso il quale il giudice *cita* la legge nelle sue sentenze. Ci siamo interrogati a lungo tanto sulle conseguenze della loro ammissione quanto su quelle della loro esclusione. E abbiamo preferito per ragioni di semplicità quest'ultima opzione, confidando nel fatto che l'elevata dimensione media dei documenti (sia in termini di terms che di words) fornisse sufficienti garanzie alla rappresentazione del loro contenuto, almeno per quelli che sono i nostri scopi, anche in assenza di

[4]Si ricordi che ogni iterazione del Gibbs sampling produce l'aggiornamento del valore di z (topic assignment) per ciascuna parola (word) di ciascun documento.

simboli numerici. Si tratta di una semplificazione, certo, che è tuttavia pienamente giustificata nel contesto di questo lavoro.
Ciò detto, cogliamo l'occasione per tornare su quanto osservato nelle pagine introduttive di questo lavoro. Dove cioè abbiamo accennato alla collaborazione in corso tra il Ministero della Giustizia e l'Istituto di Teoria e Tecniche dell'Informazione Giuridica del Consiglio Nazionale delle Ricerche (ITIIG-CNR) per la sperimentazione di metodologie e l'implementazione di prototipi software con l'obiettivo di arricchire di metadati semantici corpora di documenti giurisprudenziali [Agnoloni et al. (2014)]. Nell'approccio da noi sperimentato infatti, ispirato ai principi del *Machine Learning*, non occorre avere alcuna conoscenza a priori del dominio applicativo (c'è solo da fissare il numero dei topic K) mentre in quello seguito dall'ITTIG-CNR che si ispira ai principi della *Knowledge Engineering* è richiesto un dispendioso intervento esterno di "codifica della conoscenza" ad opera di esperti del dominio. Ciò che può diventare particolarmente problematico se si pensa agli sforzi necessari per tenere aggiornata la codifica dei riferimenti legislativi (che sono appunto simboli numerici strutturati) in un paese come l'Italia con una produzione normativa senza pari nel mondo.
Resta tuttavia il fatto che ragionare su come integrare efficacemente la conoscenza esperta in un topic model, avendo a riferimento applicazioni in ambito giurisprudenziale, può costituire un problema di ricerca molto interessante e dagli sviluppi inaspettati.
A titolo esemplificativo, la Tabella 2.5 contiene la lista dei 50 termini più frequenti nei documenti del corpus ordinata in senso decrescente. Come era da attendersi, trattandosi di testi di sentenze, la gran parte delle posizioni disponibili risulta occupata da termini propri del lessico procedurale-giuridico.
Si tenga conto a questo proposito che in generale una sentenza ha la forma di un documento *tipico* normalmente composto di quattro sezioni: intestazione, motivi di fatto, motivi di diritto, decisione. Nella *intestazione* sono riportate le generalità degli attori del processo, nei *motivi di fatto* si richiamano i fatti del ricorso, ovvero le circostanze che lo hanno originato, nei *motivi di diritto* si elencano le questioni giuridiche poste a fondamento sia del ricorso che della decisione finale, e nella *decisione* è contenuto il *dispositivo* della sentenza, ovvero la decisione finale presa dal collegio giudicante. Si confronti a tale proposito il testo della sentenza riportata in Tabella 2.6.
Al netto della intestazione, è lecito ritenere che il lessico dei *motivi di diritto* (anche detto profilo legale) sia allo stesso tempo più omogeneo e più diffuso nel corpus di quanto non sia quello, per forza di cose più eterogeneo, dei *motivi di fatto* (anche detto profilo fattuale). Questo perchè in generale allo "stesso" profilo legale possono corrispondere più profili fattuali differenti. Si tratta in sostanza di una conseguenza

Termine	Freq	Termine	Freq
motivo	376.161	sensi	138.763
ricorrente	324.022	contratto	138.551
parte	278.001	generale	138.235
essere	275.361	lavoro	137.747
giudizio	269.989	base	130.433
violazione	236.396	processo	128.034
diritto	233.963	ex	126.435
motivazione	220.525	grado	122.854
relazione	211.688	termine	120.164
motivi	205.930	parti	118.415
fatto	202.384	legge	115.637
merito	190.868	atti	114.786
roma	189.383	riferimento	106.769
decisione	187.026	principio	106.144
causa	183.765	udienza	102.388
persona	171.051	pagamento	102.019
domanda	164.242	attività	101.678
applicazione	163.240	studio	97.903
spese	160.405	artt	97.555
società	156.850	rapporto	97.099
caso	156.441	questione	96.719
atto	147.948	esame	96.599
avverso	144.014	giusta	96.522
impugnata	143.270	pubblica	95.448
stata	141.300	accertamento	93.433

Tabella 2.5: I 50 termini più frequenti nella DTM originale

del principio che una stessa norma può regolare una pluralità di fatti differenti.
La questione che si pone allora a questo punto è: quali effetti tutto ciò produce sui modelli adottati e in termini di prestazioni computazionali e, cosa ancora più importante, in termini di capacità di rispondere alle specifiche esigenze che ne hanno suggerito l'utilizzo?
Nelle pagine che seguono, qui e nel prossimo capitolo, le risposte che ci siamo dati.

2.3 La selezione del modello

La natura non supervisionata dei "topic models" rende difficile risolvere il problema della selezione del modello. Nell'approccio LDA infatti il numero dei topic K deve

SENTENZA
sul ricorso 28429/2011 proposto da: FIAT GROUP AUTOMOBILES SPA - ricorrente - contro MARIO BIANCHI - intimato.
MOTIVI DI FATTO
1. - Con ricorso al Giudice del lavoro di Torino, Mario Bianchi conveniva in giudizio il datore di lavoro FIAT Auto spa e, assumendo illegittima la sua collocazione in cassa integrazione guadagni straordinaria (CIGS) per il periodo 28.7-1.8.2003 e 9.9-21.11.2003 ne chiedeva la condanna al pagamento della differenza tra quanto percepito a titolo di integrazione e quanto spettante a titolo di retribuzione .....
MOTIVI DI DIRITTO
8. - Neppure può sostenersi che l'accordo 18.3.03.7.2003 avrebbe sanato ogni eventuale vizio della procedura attivata con la lettera 31.10.02.In proposito va precisato che la giurisprudenza richiamata dalla ricorrente (Cass. 2.8.04 n. 14721, 21.8.03 n. 12307 ed altre) parte dal presupposto che l'accordo sia di per sè esaustivo delle esigenze conoscitive e di esternazione imposte dal combinato normativo della L. n. 164, art. 5 e della L. n. 223, art. 1, commi 7 e 8, in quanto .....
DECISIONE
11. - In conclusione, il ricorso è infondato e deve essere rigettato. La mancata costituzione della parte resistente rimasta intimata esime il collegio dal provvedere sulle spese del giudizio di legittimità.

Tabella 2.6: Una sentenza generica

essere specificato *a priori*, quale parametro da fissare prima (o al più durante) della fase di addestramento del modello.
La determinazione del valore di K ha stimolato nel tempo un ampio dibattito, nel corso del quale si sono delineati quattro diversi orientamenti:

1. il valore di K è parte del problema ed è specificato da una fonte esterna;

2. il valore di K è stabilito misurando l'errore rispetto a un compito assegnato (ad esempio, come nel nostro caso, la classificazione dei documenti) [Blei & Lafferty (2009)];

3. il valore di K è deciso sulla base della capacità di generalizzazione del modello, ovvero sulla base di una misura di verosimiglianza[5] dell'insieme di prova (*test set*) rispetto ai parametri stimati sull'insieme di addestramento (*training set*) [Wallach et al. (2009), Grün & Hornik (2011)];

4. il numero dei topic K è una variabile endogena del modello stimata direttamente durante la fase di addestramento [Teh et al. (2006)].

[5]Nella linguistica computazionale tale misura (o qualche sua variante) viene chiamata *perplexity*.

In questo lavoro abbiamo sperimentato gli approcci di cui ai punti 2 e 3. Al loro esame è dedicato il paragrafo che segue.

2.3.1 Un compito assegnato: la classificazione dei documenti

La classificazione (automatica) dei documenti, ovvero «l'attività di etichettare in modo automatico testi in linguaggio naturale con classi tematiche appartenenti a un insieme predefinito»[Sebastiani (2002)], costituisce uno dei problemi principali dell'analisi automatica dei testi.
Almeno fino agli anni '80, il problema è stato affrontato ricorrendo alla definizione manuale di regole basate su una conoscenza approfondita del dominio applicativo e rispetto alle quali classificare un documento sotto una classe piuttosto che sotto un'altra (approccio di tipo KE-Knowledge Engineering). Nel corso degli anni '90, si è affermato invece, fino a diventare dominante, un altro approccio che contrariamente al precedente non richiede alcuna conoscenza del dominio applicativo (approccio di tipo ML-Machine Learning).
Nel paradigma ML la prospettiva è rovesciata rispetto a quello KE e il processo di classificazione è un processo induttivo che *apprende* le caratteristiche delle classi da un insieme di documenti pre-classificati. I suoi vantaggi stanno in un'accuratezza paragonabile a quella raggiunta da esperti umani e in un considerevole risparmio in termini di tempo-lavoro, dal momento che la costruzione del classificatore non deve richiedere alcun intervento esterno.
Diversamente dai tentativi di classificazione automatica di corpora giurisprudenziali sperimentati dal Ministero della Giustizia, nella sua più volte richiamata collaborazione con l'ITTIG-CNR, in questo lavoro utilizzeremo la classificazione dei documenti soltanto come metodo indiretto di selezione del modello, ossia di determinazione del valore di K. Con risultati che rimangono tuttavia interessanti se considerati anche in una prospettiva più ampia.

2.3.2 La classificazione dei documenti e il modello LDA

Siano $\mathcal{D} = \{d_i\}$ un insieme di D documenti[6] e $\mathcal{C} = \{c_j\}$ un insieme di C classi predefinite. Sia inoltre $\Phi : \mathcal{D} \times \mathcal{C} \to \{T, F\}$ la funzione (incognita) che descrive l'associazione *vera* tra documenti e classi: per ogni coppia (d_i, c_j) il valore T (F) indica di classificare (non classificare) il documento d_i sotto la classe c_j.
Un *classificatore* $\hat{\Phi} : \mathcal{D} \times \mathcal{C} \to \{T, F\}$ è una funzione alla quale è richiesto di *approssimare* nella maniera migliore possibile l'incognita Φ.

[6] Sarebbe forse meglio dire di loro rappresentazioni.

Ne segue che nessuna conoscenza *esterna* al contenuto testuale del documento risulta disponibile o, se disponibile, utilizzabile e le stesse classi devono essere considerate niente altro che pure *etichette* simboliche.
Il caso in cui ogni documento può essere assegnato a un'unica classe viene chiamato *singola-etichetta* (classi non sovrapponibili), quello in cui può essere assegnato a più classi *multi-etichetta* (classi sovrapponibili). D'ora in avanti considereremo esclusivamente il caso singola-etichetta.
Si ricordi che uno dei vantaggi del modello LDA consiste nel fornire una rappresentazione esplicita dei documenti di un corpus come vettori $\widehat{\boldsymbol{\theta}}_d$:

$$\widehat{\boldsymbol{\theta}}_d = (\hat{\theta}_{d1}, \hat{\theta}_{d2}, \ldots, \hat{\theta}_{dK}) \tag{2.3}$$

cioè come punti del simplesso $K-1$ dimensionale generato dai topic.
In aggiunta, nella Tabella 2.2 la colonna `ID` riporta le etichette delle 20 classi di materia dei nostri documenti.
Tutto ciò premesso, la strategia di selezione del modello adottata si può descrivere nel seguente modo:

1. Abbiamo generato 20 modelli LDA al variare di K da un minimo di 50 ad un massimo di 1000 topic (con passo 50).

   Quindi, per ogni valore di K:

2. abbiamo separato i vettori $\widehat{\boldsymbol{\theta}}_d$ in due insiemi, uno di addestramento (training set) pari al 75% dei documenti e uno di prova (testing set) pari al restante 25%. La selezione dei due insiemi è stata eseguita (casualmente) in modo tale da bilanciare la distribuzione interna delle etichette di classe e mantenerla coerente con quella dell'intero corpus[7];

3. abbiamo adattato all'insieme di addestramento un modello di regressione multinomiale[8] regolarizzata con penalizzazione di tipo *lasso* con i vettori $\widehat{\boldsymbol{\theta}}_d$ nel ruolo di predittori e le etichette di classe `ID` in quello di variabili-target;

4. abbiamo infine calcolato l'errore di classificazione sull'insieme di prova.

Tutte le elaborazioni sono state eseguite sul cluster *Terastat*, la piattaforma di calcolo distribuito disponibile presso il Dipartimento di Scienze Statistiche de La Sapienza di Roma. Con tempi variabili, limitatamente all'addestramento del modello LDA, da un minimo di circa due ore con $K = 50$ ad un massimo di quasi 8 ore con

[7] Gli insiemi di *training* e di *testing* sono stati ovviamente tenuti costanti per tutti i modelli al variare di K.

[8] Il nostro *classificatore* $\hat{\Phi}$.

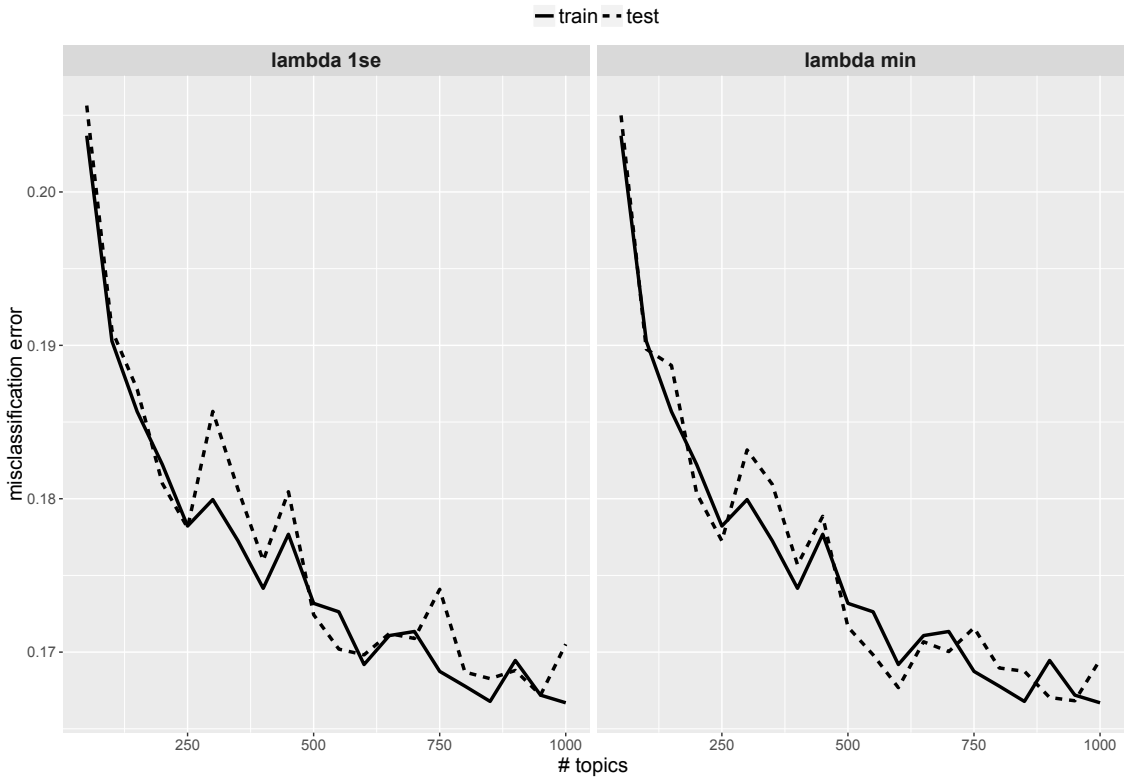


Figura 2.3: Errori di classificazione sul train e sul test set (10-fold cross-validation)

$K = 1000$.
Il modello di regressione multinomiale utilizzato è quello descritto in Friedman et al. (2010), implementato attraverso le funzioni del pacchetto `glmnet` in `R`. Rispetto ad altri modelli di classificazione (tipo reti neurali o support vector machines): 1) ha prodotto sperimentalmente risultati migliori in termini di accuratezza, 2) è un modello penalizzato che come tale prova a mitigare gli effetti della struttura di dipendenza implicita dei vettori $\widehat{\boldsymbol{\theta}}_d$ dovuta al vincolo della somma unitaria, 3) consente una selezione (automatica) dei predittori più utili ai fini della classificazione (cosa da non sottovalutare dato il loro numero elevato).
Per completezza bisogna aggiungere che la natura composizionale dei vettori $\widehat{\boldsymbol{\theta}}_d$, che sono come già detto "composizioni" di topic con il vincolo di sommare a 1, ha causato in più di qualche occasione un mal funzionamento dell'algoritmo di classificazione. Problema, questo, che è stato superato con un'opportuna trasformazione logaritmica[9] dei $\widehat{\boldsymbol{\theta}}_d$, in grado di proiettare punti di un simplesso su uno spazio euclideo permettendo quindi di operare in quest'ultimo senza vincoli.
I grafici in Figura 2.3 riportano gli errori di classificazione complessivi (*overall accu-*

[9]La *isometric log ratio transformation* (ilr) secondo quanto suggerito in Van den Boogaart & Tolosana-Delgado (2013)

	valore vero	
valore predetto	positivo	negativo
positivo	A	B
negativo	C	D

Tabella 2.7: Matrice di confusione per problemi a due classi

racy), calcolati come percentuale di documenti classificati correttamente sul totale dei documenti nelle due fasi di addestramento e di prova, al variare del numero dei topic K. Si osservi che "lambda 1se" e "lambda min" si riferiscono a due diverse modalità di fissare il valore del parametro di tuning λ nella fase di addestramento (via cross-validation[10]). In entrambi i casi, peraltro assai simili, l'andamento è decrescente e il modello migliore, ovvero con l'errore di classificazione minimo sull'insieme di prova, risulta essere quello con un numero di topic pari a 950. Ulteriori sperimentazioni con un numero di topic superiore a 1000 (che qui per semplicità non riportiamo) hanno confermato tale andamento.

Come valutare i risultati della classificazione?

Limitandoci al caso "lambda min", un esame approfondito delle matrici di confusione associate rispettivamente al modello ottimale con $K = 950$ e al modello sub-ottimale con $K = 250$ (valore scelto perchè pari ai livelli di classificazione manuale delle massime utilizzati in Cassazione) ci consente di fare qualche riflessione, che ci sarà di grande utilità quando nei capitoli successivi passeremo all'esame di situazioni concrete incontrate nel corso dei nostri esperimenti.
Gli errori di classificazione complessivi dei due modelli risultano nell'ordine pari al $16,68\%$ e al $17,72\%$.
Per avere un riferimento si consideri che assumendo come predittori i vettori-riga della matrice DTM base, e ripetendo i passi di cui al paragrafo precedente, il medesimo[11] problema di regressione multinomiale produce errori di classificazione complessiva pari al $17,00\%$ nella fase di addestramento e al $16,80\%$ in quella di prova. Nelle Tabelle 2.8, 2.9 e 2.10 proponiamo pertanto un confronto tra le matrici di confusione relative a questi tre casi.

[10] Per i dettagli si confronti la sezione 3.4 in Hastie et al. (2009). Per ora ci basti dire che nel primo caso si segue un approccio conservativo e si sceglie il valore di λ corrispondente al modello più parsimonioso entro la soglia di 1 errore standard dalla massima accuratezza media ottenuta via cross-validation. Nel secondo caso, si sceglie semplicemente il valore di λ cui corrisponde il modello con la massima accuratezza media (sempre ottenuta via cross-validation).

[11] Restando cioè invariati gli insiemi di addestramento e di prova.

Come risulta evidente in ognuna di esse, le classi più numerose, con l'eccezione della classe 30 (Contratti), sono ben classificate. Gli errori di classificazione più gravi si concentrano tutti nelle classi meno numerose.
È questo un fenomeno che si può spiegare se si considera che quest'ultime si riferiscono a domini che tendono spesso ad essere già contenuti nelle classi più numerose (si pensi, tanto per fare alcuni esempi, alle questioni di *Previdenza* che intersecano quelle di *Lavoro* o alle questioni di *Appalto* che si sovrappongono a quelle dei *Contratti*). Se vogliamo, gli errori di classificazione nella classe *Contratti* costituiscono una conferma della vastità delle situazioni di fatto e di diritto da essa contemplate, le quali propendono per tale ragione ad intersecarsi con quelle di tutte le altre classi.
Avendo a riferimento la matrice di confusione a due classi della Tabella 2.7 si possono definire le seguenti misure di *sensitività*, *specificità* e *accuratezza bilanciata*:

$$\text{sensitività} = \frac{A}{A+C}$$

$$\text{specificità} = \frac{D}{B+D}$$

$$\text{accuratezza bilanciata} = \frac{\text{sensitività} + \text{specificità}}{2}$$

Le misure di *sensitività* e di *specificità* si riferiscono rispettivamente al tasso di veri positivi e di veri negativi, cioè la proporzione di positivi (negativi) che sono correttamente identificati come tali. L'*accuratezza bilanciata* non è che una semplice media aritmetica delle due. Nel caso di più di due classi, queste misure vengono calcolate confrontando ogni livello con i rimanenti (approccio uno verso tutti).
I valori dell'accuratezza bilanciata in Tabella 2.11 confermano il sostanziale accordo tra le tre situazioni considerate. Tranne che per le classi 9, 18, 51, relative a domini molto specifici, i valori dell'accuratezza bilanciata appaiono allineati tra di loro. L'unico caso di classificazione espressamente non riuscita è relativo alla classe 101, quella con il minor numero di documenti (e riferita a un dominio, *Procedura civile*, che per ragioni oggettive tende a sovrapporsi a tutti gli altri, data la natura "civile" del nostro corpus).
Possiamo dunque a ragion veduta concludere osservando che la riduzione operata dai topic mantiene una certa coerenza con la struttura di classificazione originaria dei dati, quella cioè associata alla DTM base. Quantomeno non opera distorsioni evidenti rispetto a quest'ultima. E questo anche se si considera un numero di topic di molto inferiore rispetto a quello "ottimale". Ecco spiegato perchè, tutte le volte che nelle applicazioni pratiche ci è capitato di dover scegliere un valore di K tra quelli disponibili, abbiamo preferito seguire un criterio di semplicità e di comunicabilità dei risultati piuttosto che un criterio formale di minimizzazione dell'errore di

pred/truth	9	18	25	30	37	46	50	51	53	55	56	76	93	96	101	114	123	129	148	155
9	69	0	2	33	3	1	0	0	1	2	0	0	0	0	2	6	0	0	0	2
18	0	85	1	26	0	0	0	1	0	3	0	3	0	0	4	4	1	1	1	1
25	1	0	86	14	15	7	0	1	0	1	0	0	2	0	1	4	0	0	0	2
30	83	34	24	1954	95	14	5	53	15	38	9	53	10	9	50	217	4	9	21	149
37	1	0	34	58	314	49	0	3	10	3	1	2	49	0	8	12	2	2	1	25
46	0	0	2	2	32	39	0	1	1	0	0	0	4	0	0	2	0	1	0	1
50	0	0	1	14	3	1	2028	0	4	5	4	9	1	5	2	5	0	5	9	2
51	0	9	0	42	6	0	2	195	7	5	2	7	2	2	7	6	0	1	5	1
53	2	0	1	23	2	0	0	1	271	1	0	1	3	0	0	35	6	0	2	2
55	2	12	0	25	4	0	3	2	3	464	1	17	3	4	4	6	0	1	7	4
56	1	1	0	10	8	0	0	0	2	2	329	1	0	1	2	3	0	0	2	1
76	5	5	0	63	5	2	12	8	3	22	2	3717	1	229	18	35	8	5	11	4
93	0	1	1	4	38	1	0	0	1	1	0	0	51	0	0	0	0	0	0	0
96	0	0	1	4	1	0	0	2	0	2	0	209	0	695	1	2	2	1	2	0
101	0	0	0	4	0	0	0	3	0	1	0	0	0	0	2	0	0	0	0	0
114	4	5	8	193	12	9	0	3	10	0	0	13	2	2	15	677	2	2	3	10
123	1	0	0	4	2	0	2	1	7	0	0	2	0	0	1	6	97	1	0	2
129	3	0	0	33	4	2	0	1	0	0	0	4	1	0	1	11	2	288	0	1
148	3	2	1	37	3	0	4	4	8	8	1	15	0	6	1	9	7	17	4027	0
155	12	2	1	101	24	0	0	3	1	3	1	1	1	0	0	8	0	1	2	199

Tabella 2.8: Matrice di confusione associata al modello con 950 topic (lambda minimo)

classificazione. Resta il fatto che un percorso da esplorare in futuro è certamente quella della combinazione di più modelli per diversi valori di K, come in parte già osservato in Hofmann (1999).

pred/truth	9	18	25	30	37	46	50	51	53	55	56	76	93	96	101	114	123	129	148	155
9	74	0	3	36	4	1	0	0	0	2	0	0	0	0	2	7	0	0	0	11
18	0	83	1	28	1	0	0	5	0	4	0	1	0	0	1	4	1	0	0	3
25	3	0	103	21	17	9	0	2	0	0	0	1	7	0	4	6	0	0	0	1
30	78	33	16	1900	91	12	4	42	15	45	9	53	13	7	42	179	7	14	22	167
37	0	1	22	55	318	44	0	3	5	2	1	2	72	0	5	12	5	2	1	32
46	0	0	4	2	36	46	0	1	0	0	0	0	2	0	2	3	0	0	0	0
50	1	0	0	19	2	0	2025	1	3	1	1	9	1	8	1	9	0	7	5	0
51	0	12	1	46	5	0	1	192	9	5	1	11	3	1	11	9	0	1	4	3
53	2	0	1	27	6	0	0	2	299	0	0	1	5	0	1	38	12	0	4	4
55	4	16	1	44	4	0	6	10	2	441	0	19	2	2	11	12	0	1	5	13
56	0	1	1	14	4	1	0	1	1	5	328	3	0	3	1	5	0	1	1	3
76	5	5	1	63	3	4	5	10	0	28	4	3689	1	242	14	48	17	9	11	6
93	0	0	0	4	31	1	0	0	1	2	0	0	19	0	0	0	0	0	0	0
96	0	0	0	4	0	0	2	2	0	0	1	210	0	683	0	6	2	3	4	2
101	0	0	0	3	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0
114	5	2	5	186	10	4	1	1	4	4	2	29	3	4	15	674	8	4	3	8
123	1	0	0	8	1	0	2	0	2	0	0	3	0	0	2	12	66	2	0	0
129	2	1	1	26	1	1	2	2	0	2	0	3	0	1	1	5	0	273	4	1
148	1	1	1	44	4	0	8	3	2	10	2	18	0	2	4	10	13	18	4029	3
155	11	1	2	114	33	2	0	4	1	10	1	1	2	0	1	9	0	0	0	149

Tabella 2.9: Matrice di confusione associata al modello con 250 topic (lambda minimo)

pred/truth	9	18	25	30	37	46	50	51	53	55	56	76	93	96	101	114	123	129	148	155
9	40	1	1	15	0	0	0	0	0	0	0	7	0	0	0	5	0	0	0	5
18	0	62	0	14	0	0	0	7	0	5	0	2	0	0	2	3	1	2	0	1
25	1	0	87	13	9	7	0	3	0	0	0	0	4	0	2	3	0	0	0	0
30	128	74	45	2200	189	33	11	114	47	65	22	66	22	17	77	301	27	62	25	235
37	0	0	18	28	298	40	0	2	8	3	0	0	54	0	3	8	1	1	1	7
46	0	0	1	1	25	35	0	1	0	0	0	0	1	0	1	3	0	0	0	0
50	0	1	0	6	0	0	2037	0	0	3	5	5	0	4	4	2	0	3	4	0
51	0	4	0	17	4	0	0	125	7	1	3	7	1	3	4	2	0	1	2	0
53	0	0	0	19	2	0	0	2	266	1	0	1	4	0	0	27	9	0	5	1
55	2	8	0	23	1	1	0	2	2	451	0	10	2	3	5	9	0	1	5	10
56	0	0	0	5	1	0	0	0	1	2	318	2	0	1	0	1	0	0	0	1
76	1	1	0	33	1	0	4	11	3	14	1	3737	1	175	0	25	14	4	19	2
93	0	0	2	2	13	1	0	0	0	1	0	0	38	0	0	0	0	0	0	0
96	0	0	0	4	0	0	0	1	0	0	0	193	0	746	0	3	1	1	2	0
101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0
114	9	3	5	156	13	8	1	4	7	4	0	18	3	2	12	637	4	7	0	6
123	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	4	70	0	1	0
129	1	2	2	13	0	0	0	0	0	0	0	2	0	1	0	5	0	246	3	0
148	0	0	1	27	0	0	3	2	1	2	0	3	0	1	0	7	4	7	4026	0
155	5	0	1	67	15	0	0	8	1	9	1	0	0	0	0	3	0	0	0	138

Tabella 2.10: Matrice di confusione (lambda minimo) associata alla DTM base

ID materia	K = 950	K = 250	DTM base
9	0.6831	0.6961	0.6060
18	0.7712	0.7647	0.6977
25	0.7625	0.8140	0.7657
30	0.8418	0.8329	0.8675
37	0.7678	0.7712	0.7561
46	0.6548	0.6827	0.6391
50	0.9911	0.9904	0.9943
51	0.8429	0.8371	0.7201
53	0.8917	0.9318	0.8847
55	0.9108	0.8889	0.8996
56	0.9691	0.9673	0.9539
76	0.9435	0.9387	0.9504
93	0.6949	0.5720	0.6456
96	0.8582	0.8517	0.8856
101	0.5082	0.5041	0.5336
114	0.8147	0.8131	0.7965
123	0.8694	0.7510	0.7669
129	0.9281	0.9060	0.8664
148	0.9876	0.9873	0.9898
155	0.7407	0.6783	0.6669

Tabella 2.11: Valori dell'accuratezza bilanciata per classe

2.3.3 Misurare la capacità di generalizzazione del modello

Nel paragrafo precedente abbiamo discusso i risultati dell'applicazione di un metodo indiretto di valutazione del modello LDA basato sulla misura della performance rispetto ad un compito assegnato. Qui discuteremo invece i risultati dell'applicazione di un metodo di valutazione diretto basato sulla misura della capacità di generalizzazione del modello. È il metodo implementato in MALLET e che, come dimostrato da Wallach et al. (2009) in esperimenti su dati reali, è accurato, computazionalmente efficiente e universale.

Siano $\mathcal{D}_{train}$ e $\mathcal{D}_{test}$ due insiemi di documenti rispettivamente di addestramento e di prova. Una metrica naturale e diretta di valutazione del modello LDA è la probabilità marginale dell'insieme di prova $\mathcal{D}_{test}$ dato l'insieme di addestramento $\mathcal{D}_{train}$:

$$p(\mathcal{D}_{test}|\mathcal{D}_{train}) = \int p(\mathcal{D}_{test}|\alpha, \boldsymbol{\phi}) p(\alpha, \boldsymbol{\phi}|\mathcal{D}_{train})\, d\alpha\, d\boldsymbol{\phi} \tag{2.4}$$

Questo integrale non può essere calcolato esattamente e tipicamente viene approssimato dalla seguente stima puntuale:

$$p(\mathcal{D}_{test}|\alpha_{train}, \boldsymbol{\phi}_{train}) = \prod_{d \in \mathcal{D}_{test}} (\boldsymbol{w}_d|\alpha_{train}, \boldsymbol{\phi}_{train}) \tag{2.5}$$

che è la verosimiglianza dell'insieme di prova rispetto ai parametri stimati in fase di addestramento. Un modello sarà tanto migliore, in termini di capacità di generalizzazione, quanto più elevato risulterà questo valore.

Si noti che nel derivare 2.5 si è usato il fatto che ogni documento in $\mathcal{D}_{test}$ può essere valutato separatamente, data l'assunzione di indipendenza tra documenti implicita nel modello LDA.

Il problema della valutazione del modello diventa così quello di calcolare la probabilità $p(\boldsymbol{w}_d|\alpha_{train}, \boldsymbol{\phi}_{train})$ per ogni $d \in \mathcal{D}_{test}$. Quest'ultima può essere vista come la costante di normalizzazione in:

$$p(\boldsymbol{z}_d|\boldsymbol{w}_d, \alpha_{train}, \boldsymbol{\phi}_{train}) = \frac{p(\boldsymbol{z}_d, \boldsymbol{w}_d|\alpha_{train}, \boldsymbol{\phi}_{train})}{p(\boldsymbol{w}_d|\alpha_{train}, \boldsymbol{\phi}_{train})} \tag{2.6}$$

dove $p(\boldsymbol{z}_d|\boldsymbol{w}_d, \alpha_{train}, \boldsymbol{\phi}_{train})$ è la distribuzione a posteriori sulle variabili latenti $\boldsymbol{z}_d$ e $p(\boldsymbol{z}_d, \boldsymbol{w}_d|\alpha_{train}, \boldsymbol{\phi}_{train})$ è la distribuzione congiunta su quest'ultime e sulle variabili osservate $\boldsymbol{w}_d$.

In letteratura sono stati proposti molti metodi di stima di costanti di normalizzazione siffatte e una rassegna può essere trovata in Wallach et al. (2009). Noi abbiamo utilizzato quello[12] implementato in MALLET perchè, lo ripetiamo, si è dimostrato più accurato ed efficiente degli altri in esperimenti con dati reali.

[12] Chiamato anche *"left-to-right" evaluation algorithm* si basa su un approccio Monte Carlo di tipo sequenziale.

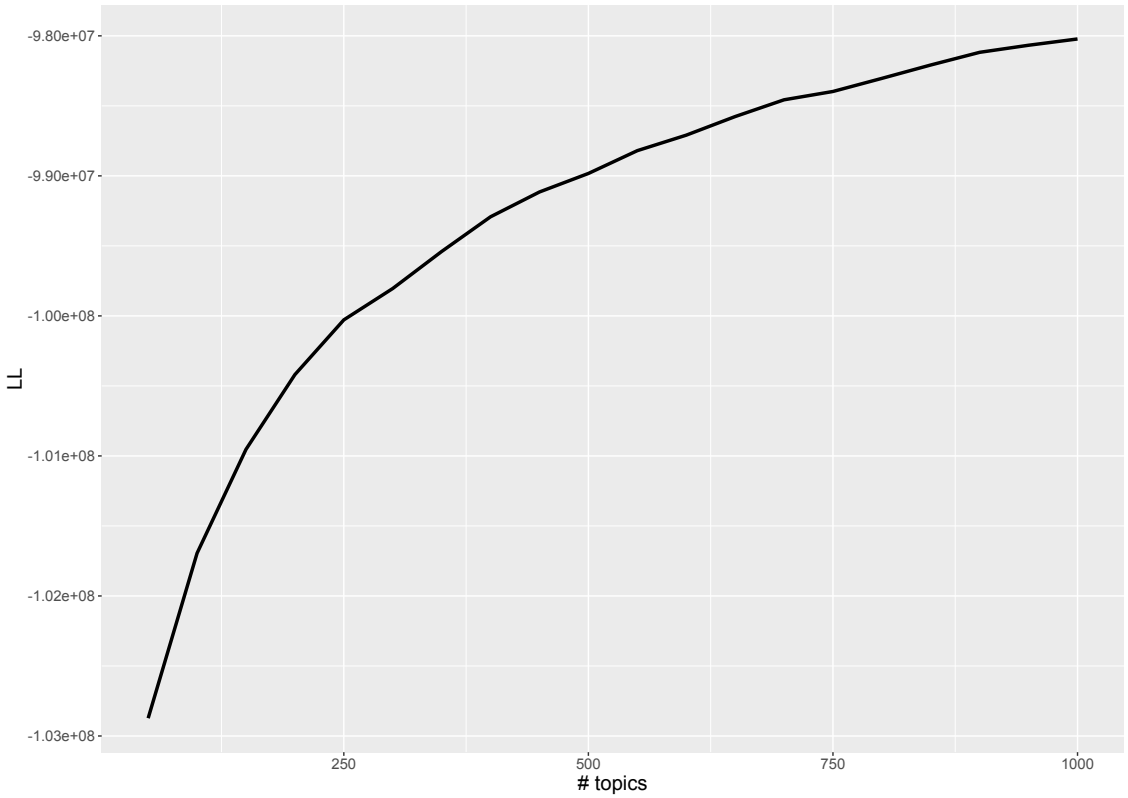


Figura 2.4: Perplexity sull'insieme di prova (DTM base)

La Figura 2.4 riporta i valori della (log) verosimiglianza in Eq. 2.5, per numero dei topic K e documenti di addestramento e di prova identici a quelli considerati al paragrafo precedente.
In accordo con quanto lì già osservato, al crescere del numero dei topic cresce la capacità di generalizzazione del modello e dunque i modelli migliori risultano essere quelli con i valori di K più elevati. Altre indicazioni al riguardo non sono date.
Ciò detto, l'andamento del grafico è una ulteriore conferma del fatto che almeno nel nostro caso la scelta di K deve essere affidata più a considerazioni di tipo pratico che teorico.
Nei capitoli successivi avremo modo di chiarire questi aspetti con alcuni esempi concreti.

Capitolo 3

L'applicazione web *Suprema*

Se pensato come strumento di analisi statistica il modello LDA non si presta a un uso immediato. Il solo output del modello non è sufficiente per consentire una facile esplorazione dei documenti, dal momento che è necessario esaminare attentamente le distribuzioni $\widehat{\boldsymbol{\theta}}_d$ e $\widehat{\boldsymbol{\phi}}_k$ per comprendere il significato dei risultati ottenuti.
In questo capitolo presenteremo `Suprema`, un metodo per rendere *visibile* l'output del modello LDA. A partire da quest'ultimo, infatti, `Suprema` crea un navigatore dei documenti capace di esplorare la struttura complessa del corpus.
In particolare ci concentreremo su quattro aree di interesse nelle quali i benefici dell'approccio considerato si sono dimostrati notevoli: 1) nel fornire una panoramica generale del corpus, 2) nel raggruppamento per sentenze omogenee, 3) nella ricerca di sentenze rilevanti per un dato tema, 4) nella ricerca di sentenze simili ad una sentenza assegnata.
Ci occuperemo infine di alcuni interessanti esempi di *anomaly detection* rivelati da `Suprema`, con riferimento a gruppi di sentenze di fatto identiche perchè corrispondenti a ricorsi seriali.

3.1 *Suprema*: un metodo per rendere *visibile* l'output di LDA

`Suprema` è un prototipo di applicazione web[1] realizzata secondo il paradigma *client-server* del pacchetto `shiny`[2] in `R`. È costruita intorno all'idea che il modello LDA fornisca rappresentazioni tematiche dei documenti e si limita a trasformare queste

[1] `Suprema` è una generalizzazione al nostro caso di studio di un'analoga applicazione descritta in Chaney & Blei (2012). È raggiungibile all'indirizzo: https://cassazione.shinyapps.io/Suprema/.
[2] http://shiny.rstudio.com/

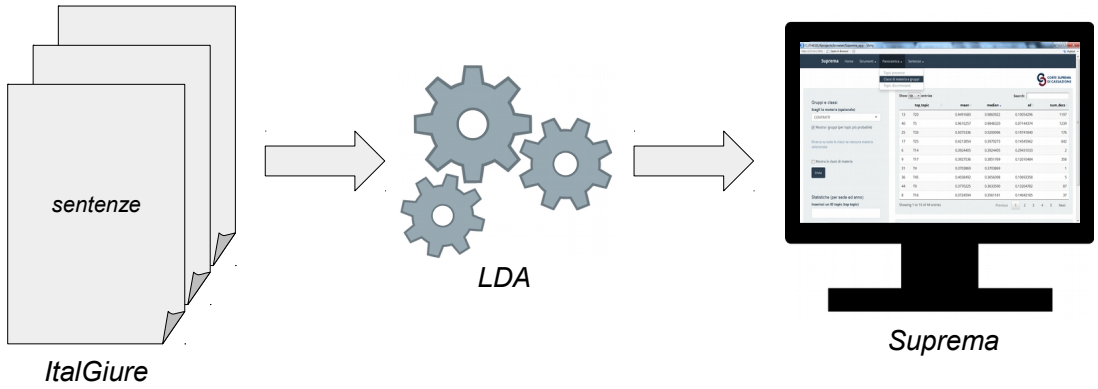


Figura 3.1: Dalle sentenze alla loro analisi tematica

rappresentazioni in un sistema utile per esplorare il corpus e interagire con la sua struttura, con particolare enfasi sulle relazioni tra topic e documenti e tra documenti e documenti.
Come illustrato in Figura 3.1, `Suprema` è un metodo per rendere *visibile* l'output del modello LDA, ovvero i vettori $\widehat{\boldsymbol{\theta}}_d$ e $\widehat{\boldsymbol{\phi}}_k$, che restituiscono le distribuzioni di probabilità rispettivamente dei topic per ogni documento e dei termini del vocabolario per ogni topic.
A partire da questi vettori `Suprema` costruisce una serie di funzioni che nelle nostre sperimentazioni si sono dimostrate capaci di far emergere strutture significative nei dati.
Allo stato attuale `Suprema` mette a disposizione le seguenti funzioni:

- **Carica modello**: carica il modello in input tra quelli disponibili.
- **Termini identificativi**: restituisce i termini identificativi (top terms) di un topic.
- **Ricerca per parole**: cerca i topic più probabili per un dato termine (o coppia di termini).
- **Topic presence**: restituisce una panoramica generale dei topic del corpus.
- **Classi di materie e gruppi**: raggruppa le sentenze per topic più probabile.
- **Topic discriminanti**: produce il grafico dei topic discriminanti di ogni classe di materia.

- **Topic correlati**: cerca i topic correlati a un dato topic.
- **Sentenze rilevanti**: cerca le sentenze più rilevanti per classe di materia e per topic.
- **Sentenze simili**: cerca le sentenze simili a una sentenza assegnata.
- **Composizione sentenza**: restituisce la composizione tematica di una sentenza.

Nei paragrafi successivi passeremo in rassegna ognuna di queste funzioni con riferimento a vari casi concreti. Lo faremo dopo averle suddivise in tre gruppi, "Strumenti", "Temi" e "Sentenze", che costituiscono le porte di accesso a `Suprema`.
Mostreremo in particolare quale uso farne se l'obiettivo è quello di integrare le funzioni di navigazione dell'archivio delle sentenze civili di ItalGiure muovendo da un approccio diverso ma complementare rispetto a quello oggi adottato in Cassazione.
Per semplicità di esposizione, in tutte le analisi che seguono il numero dei topic è stato fissato pari a $K = 50$. Pur non essendo un valore di K ottimale (almeno secondo criteri formali, come visto nel capitolo precedente) i risultati ottenuti mantengono una validità generale. Ciò che sperimentalmente si osserva all'aumentare del numero dei topic, e cioè una crescente articolazione tematica, non muta infatti la sostanza di ciò che diremo.
Infine, per omogeneità con il sistema ItalGiure, è stato considerato esclusivamente il caso DTM base.

3.1.1 Strumenti

Il gruppo "Strumenti" comprende le funzioni *Carica modello*, *Termini identificativi* e *Ricerca per parole*.
La funzione *Carica modello* provvede a caricare in `Suprema` l'output del modello LDA tra i 20 disponibili, corrispondenti cioè alla sequenza dei valori di K da 50 a 1000 (con passo 50) già incontrata al capitolo precedente. E questo sia per la DTM base che per quella ridotta.
La funzione *Termini identificativi* ordina i termini di un topic per valori decrescenti di probabilità e ne restituisce la lista dei primi n (top terms, con n per default pari a 10). Tipicamente sono i termini sui quali la distribuzione $\widehat{\phi}_k$ tende a concentrare la sua massa e che sintetizzano il "significato" del topic condensandolo in una sorta di etichetta che ne rappresenta il contenuto tematico.
La Figura 3.2 mostra i 10-top terms del topic $T22$ che si riferisce chiaramente a questioni di responsabilità civile per risarcimento danni e come tale potrebbe essere

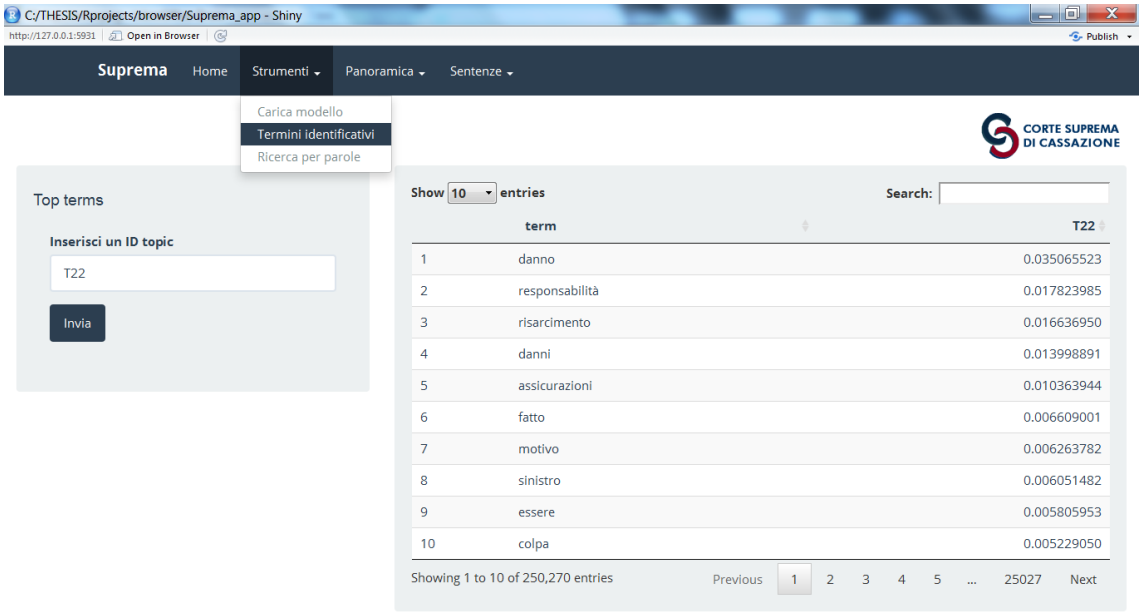


Figura 3.2: I 10 termini identificativi del topic $T22$

etichettato.

`Suprema` implementa anche un'altra modalità di visualizzazione dei termini identificativi di un topic. Essa si basa su una misura di *rilevanza* così come proposto in Sievert & Shirley (2014). La rilevanza di un termine t per un topic k è definita come:

$$r(t|k,\lambda) = \lambda \log(\widehat{\phi}_{kt}) + (1-\lambda)\log\left(\frac{\widehat{\phi}_{kt}}{f_t}\right) \tag{3.1}$$

dove λ è un parametro di bilanciamento compreso tra 0 e 1, $\widehat{\phi}_{kt}$ è la probabilità del termine t condizionata al topic k e f_t è la frequenza relativa del termine t nell'intero corpus.

Questa misura è molto flessibile; ponendo infatti $\lambda = 1$ si ottiene un semplice ordinamento dei termini in ordine decrescente di probabilità, e ponendo invece $\lambda = 0$ se ne ottiene uno basato esclusivamente sul rapporto tra la probabilità del termine t sotto il topic k e quella marginale su tutto il corpus. Fissando opportunamente il valore di λ è allora possibile selezionare termini che siano identificativi del topic e non troppo comuni nel corpus. In Sievert & Shirley (2014) viene suggerito di fissare un valore di λ pari a 0.6.

La funzione *Ricerca per parole* restituisce la lista degli m topic più probabili per un dato termine *query* (con m per default pari a 5), ovvero quelli, tra i topic disponibili, che più probabilmente hanno generato il termine cercato. Si confronti a questo proposito la Figura 3.3.

È questa una funzione in grado di aprire prospettive interessanti se si immagina che

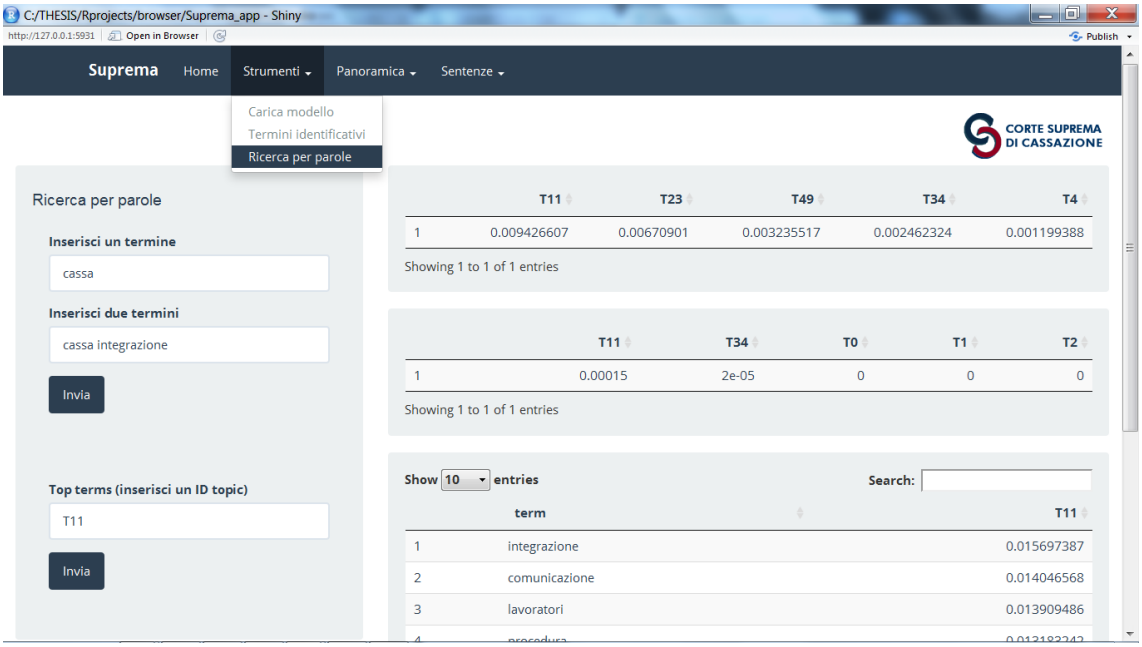


Figura 3.3: Un esempio di ricerca per parole chiave

potrebbe servire come filtro per migliorare le prestazioni di ItalGiure nella ricerca di documenti rilevanti. Un documento rilevante potrebbe infatti non contenere affatto il termine cercato ma molti tematicamente simili, ovvero appartenenti allo stesso (o agli stessi) topic. In questo modo, dopo aver ordinato i topic nel senso sopra descritto, non resterebbe che integrarli all'interno dei sistemi di information retrieval esistenti.

A titolo puramente esemplificativo `Suprema` ammette la possibilità che la ricerca per parole contempli non uno ma due termini, con i topic ordinati in base al prodotto delle probabilità di quest'ultimi. L'assunzione implicita è che essi siano indipendenti. Si osservi che non è corretto parlare in questo caso di bigrammi in quanto essi sono per definizione composti da due termini tra loro dipendenti.

Una seria estensione al caso degli n-grammi richiede più di uno sviluppo futuro.

3.1.2 Temi

Il gruppo "Temi" comprende quattro funzioni: *Topic presence*, *Classi di materia e gruppi*, *Topic discriminanti* e *Topic correlati*[3].

Cominciamo dalla prima che rappresenta un punto di partenza per esplorare il corpus. Essa produce infatti un sommario generale dei topic, ordinati in senso decrescente in base al valore della loro presenza relativa.

[3]Di questa funzione parleremo nel paragrafo successivo insieme con quella che abbiamo chiamato *Sentenze simili*.

Seguendo Chaney & Blei (2012) stabiliamo di misurare la *presenza* relativa nel corpus (topic presence: tp) del k-esimo topic come:

$$tp_k = \frac{\sum_d \hat{\theta}_{dk}}{\sum_{d,k} \hat{\theta}_{dk}}, \forall k = 1, \ldots, K \tag{3.2}$$

ovvero, come somma normalizzata dei valori $\hat{\theta}_{dk}$ rispetto a tutti i documenti del corpus (o di un suo sottoinsieme selezionato per classe di materia).
Le Tabelle 4.9 e 4.10 in appendice elencano i termini identificativi dei 50 topic generati dal modello LDA (a partire dalla DTM base) ordinati in senso decrescente rispetto ai valori definiti in 3.2.
Come era da attendersi, nelle posizioni più elevate ritroviamo topic generalmente imputabili al profilo legale delle sentenze (ciò che nel capitolo precedente abbiamo chiamato "motivi di diritto") anche se non mancano eccezioni come nel caso dei topic $T46$, $T2$, $T41$, $T14$. Si tratta tuttavia di topic poco interessanti dal nostro punto di vista, perchè comuni a tutte le sentenze e dunque con scarso peso informativo.
Appare comunque evidente come la gran parte dei topic sia identificata da una lista di termini tra loro coerenti e rifletta in maniera ragionevole la distribuzione per materia dei documenti. Non sorprendentemente, i topic più presenti appartengono alle classi di materia più numerose, nell'ordine: *Tributi*, *Lavoro*, *Contratti*, *Equa riparazione*, *Responsabilità civile* e *Previdenza*.
Spingendoci oltre proviamo ad associare topic e materie come in Tabella 3.1.
Questa tabella[4] esplicita la relazione tra topic e classi di materia indotta dal contenuto semantico dei primi, cioè dal significato dei termini identificativi e dal loro grado di coerenza.
Da un suo rapido esame emerge chiaramente come siano le classi più numerose a presentare la più ampia articolazione in termini di topic (significativo il caso della classe *Lavoro*) e in generale a essere ben rappresentate; le classi meno numerose invece tendono a essere rappresentate da un numero assai esiguo di topic (uno o due).
Ciò dipende anche dalla scelta del valore di K pari a 50 topic, un valore basso che tende a privilegiare le classi più numerose su quelle meno numerose. È però in linea con quanto discusso nel capitolo precedente a proposito degli errori di classificazione in gran parte concentrati nelle classi meno numerose, quelle più suscettibili di sovrapporsi alle altre.
Il raggruppamento proposto in Tabella 3.1 è ovviamente soggettivo e non mancano casi di topic che avrebbero potuto essere assegnati a più classi diverse. Solo per fare qualche esempio, il topic $T22$ alla classe 30 oltre che a quella 114, il topic $T26$ alla

[4]Realizzata in collaborazione con un magistrato della sezione civile della Corte di Cassazione.

Materia	**ID materia**	**Topic**
Motivi di diritto		T31, T19, T47, T12, T38, T25, T24, T40
Tributi	148	T2, T41, T13, T1, T4, T43
Lavoro	76	T14, T21, T3, T7, T18, T39, T45, T6, T16, T34, T11
Contratti	30	T20, T5, T0, T49
Equa riparazione	50	T46, T27, T44
Responsabilità civile	114	T22
Previdenza	96	T29, T48, T30, T23
Diritti reali	37	T8
Fallimento	55	T28, T42
Vendita, Permuta, Riporto	155	T0
Famiglia	56	T10
Espropriazione	53	T9
Sanzioni amministrative	129	T37
Esecuzione forzata	51	T42
Appalto	9	T32
Comunione e condominio	25	T8
Banca e borsa	18	T49
Ricorsi contro giudici speciali	123	T37
Possesso	93	T8
Edilizia e urbanistica	46	T8
Procedura civile	101	T17

Tabella 3.1: Relazioni tra topic e classi di materia indotte dai *top terms*

classe 50 oltre che ad altre classi (in quanto relativo a questioni comuni di improcedibilità per mancanza dei requisiti essenziali del ricorso).
In generale tuttavia ogni classe sembra avere un nucleo di topic che la caratterizzano. Discorso a parte per la pseudo-classe *Motivi di diritto* aggiunta appositamente e rappresentata dai topic $T31$, $T19$, $T47$, $T12$, $T38$, $T25$, $T24$, $T40$, molto diffusi nel corpus (elevato punteggio tp) ma, come abbiamo già osservato, con scarsa capacità di discriminazione perchè comuni alla gran parte dei documenti.
Una conferma diretta della bontà dello schema in Tabella 3.1 viene dall'esame della Tabella 3.2 che, per ogni classe di materia, restituisce la distribuzione dei documenti raggruppati per topic più probabile.
Questa tabella è un prodotto della funzione *Classi di materia e gruppi* disegnata allo scopo di costruire cluster di documenti sulla base del topic più rappresentativo di ogni documento, quello cioè con il massimo valore di $\hat{\theta}_{dk}$ al variare di k (si ammette che i cluster possano riferirsi a tutto il corpus o anche a un suo sottoinsieme specificato da una particolare classe di materia).
Dal confronto tra le due tabelle emerge un sostanziale accordo. Per esempio, i topic della materia *Tributi* in Tabella 3.1 si ritrovano tutti (e in posizione elevata) in Tabella 3.2. Analogo discorso per le classi *Lavoro*, che si conferma quella più articolata

tematicamente, *Contratti* ed *Equa riparazione*. E così in verità anche per tutte le altre classi.

Materia	ID materia	Topic
Tributi	148	T2 (4212), T41 (3283), T13 (2290), T1 (1316), T19 (935), T43 (666), T4 (548), T17 (531), T37 (389), T31 (372), T39 (317), T25 (269), T24 (246), T9 (175), T38 (120), T47 (112), T16 (83), T12 (75), T15 (70), T26 (61), T0 (54), T49 (53), T18 (26), T28 (22), T10 (21), T27 (15), T35 (15), T33 (14), T32 (13), T21 (10), T8 (10), T42 (8), T46 (8), T7 (8), T29 (7), T40 (5), T36 (4), T23 (3), T44 (2), T45 (2), T6 (2), T20 (1), T48 (1), T5 (1)
Lavoro	76	T14 (2774), T21 (2571), T45 (1265), T7 (1168), T3 (1091), T38 (903), T18 (894), T19 (611), T31 (476), T29 (461), T25 (407), T34 (346), T30 (302), T11 (274), T12 (263), T17 (254), T24 (202), T39 (195), T23 (180), T33 (178), T16 (160), T6 (157), T47 (144), T27 (113), T22 (100), T42 (97), T26 (92), T48 (91), T37 (90), T4 (82), T49 (42), T15 (36), T35 (32), T36 (30), T28 (26), T32 (25), T40 (21), T10 (20), T0 (12), T2 (11), T46 (5), T41 (4), T8 (4), T20 (3), T44 (2), T13 (1), T5 (1), T9 (1)
Contratti	30	T5 (1239), T20 (1197), T0 (1193), T31 (870), T25 (842), T19 (691), T32 (533), T12 (531), T38 (432), T22 (421), T49 (374), T17 (358), T27 (248), T47 (216), T33 (176), T8 (167), T26 (156), T37 (120), T42 (102), T9 (87), T36 (76), T39 (75), T35 (74), T30 (61), T15 (56), T24 (53), T28 (53), T16 (37), T40 (31), T10 (20), T2 (20), T18 (19), T43 (16), T41 (6), T48 (6), T21 (5), T45 (5), T1 (3), T6 (3), T13 (2), T14 (2), T29 (1), T4 (1), T7 (1)
Equa riparazione	50	T46 (5702), T44 (1089), T27 (642), T26 (227), T31 (191), T17 (183), T25 (62), T47 (34), T35 (15), T48 (15), T36 (14), T12 (13), T15 (6), T19 (6), T28 (6), T42 (6), T24 (5), T38 (3), T13 (1), T33 (1), T37 (1), T39 (1), T41 (1), T49 (1)
Responsabilità civile	114	T22 (1302), T25 (422), T31 (408), T19 (299), T38 (250), T12 (222), T15 (135), T17 (124), T33 (117), T9 (106), T47 (105), T27 (100), T32 (100), T30 (87), T8 (62), T37 (59), T36 (47), T26 (46), T49 (46), T35 (28), T0 (27), T16 (22), T40 (15), T24 (12), T39 (11), T42 (10), T28 (9), T10 (7), T20 (6), T21 (4), T18 (2), T29 (2), T48 (2), T43 (1)
Previdenza	96	T29 (1362), T48 (539), T30 (456), T23 (414), T7 (144), T38 (106), T19 (100), T31 (89), T25 (86), T17 (68), T27 (52), T28 (46), T21 (44), T12 (42), T24 (34), T39 (31), T45 (25), T47 (23), T33 (19), T26 (17), T42 (15), T15 (14), T10 (13), T18 (12), T22 (8), T37 (7), T2 (6), T16 (5), T35 (5), T6 (5), T36 (4), T4 (4), T46 (4), T3 (3), T49 (3), T14 (2), T5 (2), T11 (1), T32 (1), T40 (1)
Diritti reali	37	T8 (1321), T12 (142), T0 (117), T38 (92), T19 (76), T25 (71), T17 (63), T31 (60), T35 (60), T20 (54), T9 (40), T36 (35), T47 (31), T32 (24), T26 (17), T10 (13), T42 (12), T49 (9), T27 (7), T18 (6), T30 (5), T5 (5), T22 (4), T37 (4), T15 (3), T33 (2), T39 (2), T40 (2), T1 (1), T2 (1), T24 (1), T28 (1), T44 (1), T46 (1), T48 (1)
Fallimento	55	T28 (1021), T49 (251), T31 (164), T17 (124), T38 (109), T42 (79), T12 (74), T19 (62), T25 (60), T0 (58), T47 (40), T24 (29), T39 (27), T26 (25), T23 (22), T27 (18), T32 (13), T22 (9), T36 (9), T2 (8), T9 (7), T18 (5), T21 (5), T8 (5), T35 (4), T37 (4), T10 (3), T13 (3), T15 (3), T16 (2), T41 (2), T30 (1), T40 (1)
Vendita, Permuta, Riporto	155	T0 (722), T32 (155), T12 (145), T38 (95), T31 (69), T25 (65), T19 (61), T17 (55), T8 (55), T49 (32), T47 (29), T26 (25), T35 (25), T36 (16), T9 (16), T27 (11), T28 (7), T42 (7), T22 (6), T39 (6), T30 (5), T18 (4), T33 (3), T15 (2), T16 (2), T1 (1), T10 (1), T13 (1), T37 (1), T40 (1), T5 (1)
Famiglia	56	T10 (985), T31 (130), T25 (57), T12 (51), T17 (51), T38 (38), T19 (26), T47 (14), T42 (10), T0 (7), T24 (7), T27 (7), T49 (5), T26 (4), T30 (4), T18 (2), T37 (1), T40 (1), T8 (1)
Espropriazione	53	T9 (1130), T12 (48), T25 (34), T31 (29), T19 (27), T17 (19), T8 (15), T36 (9), T42 (9), T47 (9), T26 (7), T27 (7), T38 (7), T37 (6), T24 (5), T28 (3), T49 (3), T0 (2), T32 (2), T35 (2), T18 (1), T2 (1), T39 (1), T48 (1)
Sanzioni amministrative	129	T37 (653), T31 (139), T17 (135), T19 (63), T15 (52), T26 (37), T27 (36), T24 (27), T38 (23), T12 (21), T2 (21), T25 (18), T16 (17), T47 (13), T8 (12), T42 (11), T39 (7), T13 (6), T22 (6), T43 (6), T49 (6), T33 (5), T1 (4), T21 (4), T32 (4), T41 (4), T0 (3), T30 (3), T29 (2), T40 (2), T35 (1), T44 (1), T46 (1)
Esecuzione forzata	51	T42 (395), T25 (155), T31 (134), T19 (81), T17 (58), T49 (58), T12 (44), T47 (36), T27 (25), T26 (23), T0 (18), T24 (15), T8 (12), T36 (11), T2 (10), T32 (9), T38 (8), T28 (6), T33 (5), T9 (5), T10 (4), T39 (4), T35 (3), T18 (2), T22 (2), T23 (2), T40 (2), T37 (1)
Appalto	9	T32 (399), T12 (62), T38 (55), T19 (33), T31 (30), T25 (29), T0 (26), T17 (23), T27 (19), T47 (11), T8 (9), T26 (8), T36 (6), T49 (6), T39 (5), T9 (5), T2 (4), T37 (4), T24 (3), T28 (2), T35 (2), T40 (2), T15 (1), T16 (1), T18 (1), T22 (1), T30 (1)
Comunione e condominio	25	T8 (378), T12 (40), T19 (38), T31 (30), T17 (27), T32 (24), T25 (18), T35 (16), T38 (16), T42 (15), T47 (12), T27 (10), T37 (6), T0 (4), T22 (4), T49 (4), T10 (2), T18 (2), T26 (2), T36 (2), T28 (1), T30 (1), T9 (1)
Banca e borsa	18	T49 (370), T31 (68), T12 (31), T25 (26), T19 (22), T17 (21), T47 (17), T42 (15), T38 (11), T0 (7), T15 (7), T22 (5), T35 (5), T26 (3), T27 (3), T28 (3), T36 (3), T10 (1), T2 (1), T24 (1), T32 (1), T39 (1), T43 (1), T8 (1)
Ricorsi contro giudici speciali	123	T37 (313), T47 (34), T9 (27), T31 (26), T12 (23), T19 (13), T45 (13), T17 (9), T39 (8), T33 (6), T8 (6), T23 (5), T38 (5), T41 (5), T15 (4), T0 (3), T2 (3), T22 (3), T24 (3), T40 (3), T25 (2), T29 (2), T35 (2), T1 (1), T20 (1), T21 (1), T26 (1), T27 (1), T28 (1), T32 (1), T36 (1)
Possesso	93	T8 (270), T38 (56), T12 (33), T25 (26), T31 (21), T19 (19), T0 (17), T9 (15), T35 (14), T17 (10), T42 (8), T36 (7), T26 (5), T10 (4), T47 (4), T18 (3), T37 (3), T27 (2), T28 (2), T2 (1), T39 (1)
Edilizia e urbanistica	46	T8 (367), T12 (20), T19 (19), T31 (16), T25 (15), T32 (12), T38 (10), T17 (9), T9 (5), T0 (4), T24 (4), T36 (4), T30 (3), T35 (3), T22 (2), T47 (2), T18 (1), T20 (1), T27 (1), T40 (1), T42 (1), T49 (1)
Procedura civile	101	T17 (90), T12 (82), T31 (70), T42 (41), T25 (30), T27 (27), T19 (17), T26 (16), T22 (14), T47 (12), T8 (10), T37 (8), T32 (7), T38 (7), T15 (6), T49 (6), T35 (5), T36 (5), T0 (3), T28 (3), T30 (3), T40 (3), T24 (2), T29 (2), T10 (1), T13 (1), T18 (1), T20 (1), T33 (1), T39 (1), T41 (1), T48 (1), T9 (1)

Tabella 3.2: Distribuzione dei documenti di ogni classe per topic più probabile (tra parentesi il numero di documenti)

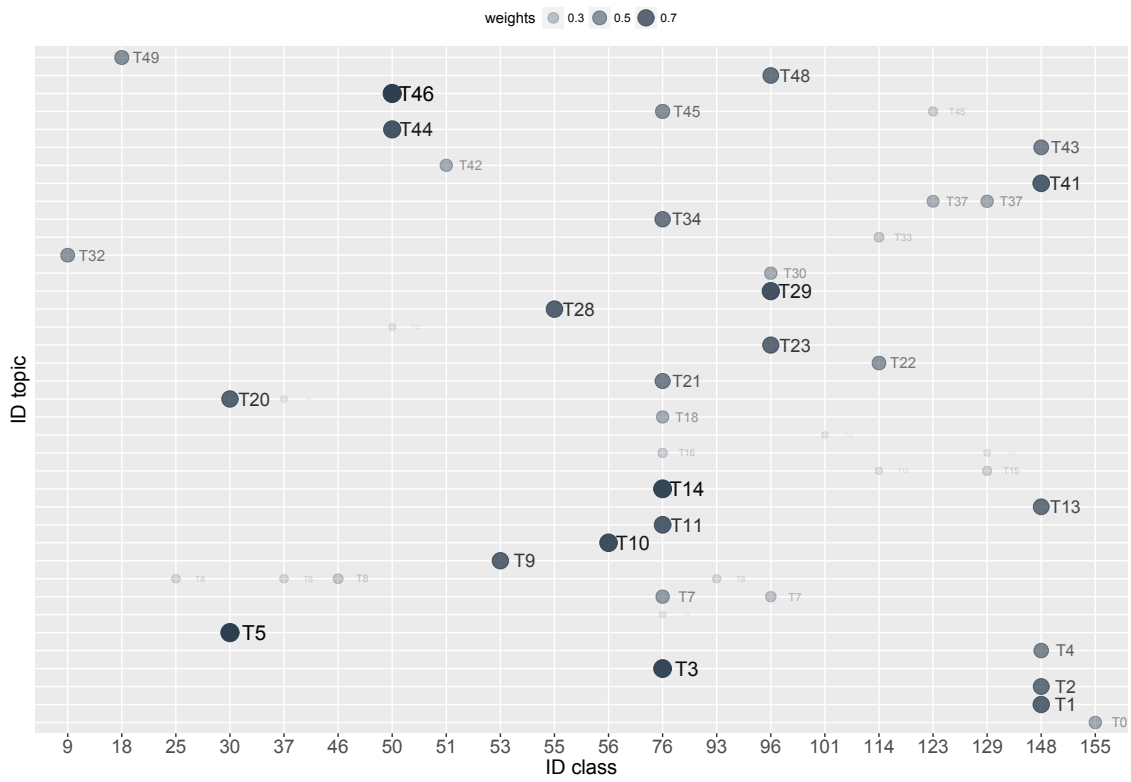


Figura 3.4: I topic caratteristici di ogni classe (soglia pari a 0.15)

In definitiva ci sono certamente casi di topic presenti in modo significativo in più classi ($T0$, $T8$, $T37$, $T42$, $T49$) ma in generale ogni classe risulta essere ben identificata da un nucleo di topic caratteristici.
Cosa quest'ultima che può essere rappresentata anche in forma grafica come nella Figura 3.4 generata dalla funzione *Topic discriminanti* di `Suprema`, una funzione disegnata allo scopo di "tracciare" la lista dei topic caratteristici di ogni classe ottenuti calcolando il peso medio di ogni topic per classe, ovvero il valore:

$$\frac{\sum_{d,k} \hat{\theta}_{dk}}{|C_i|}, \text{ per ogni classe di materia } C_i, \tag{3.3}$$

da normalizzare rispetto alla somma dei pesi medi su tutte le classi per avere una quantità compresa tra 0 e 1.
Si noti che per ragioni di leggibilità nel grafico in Figura 3.4 abbiamo riportato solo i topic con peso medio normalizzato superiore alla soglia 0.15. Come appare evidente, ogni classe, nonostante il valore relativamente basso di K, tende a non condividere i suoi topic più importanti con le altre classi.
Ciò costituisce una ulteriore conferma della capacità dei topic di individuare la struttura tematica propria di ogni classe.

3.1.3 Sentenze

Il gruppo "Sentenze" comprende le funzioni *Sentenze rilevanti*, *Sentenze simili* e *Composizione sentenza*.
Diversamente dal gruppo "Temi" che come abbiamo visto riunisce funzioni utili per esplorare i temi del corpus, qui il focus è sui documenti e sulle relazioni tra documenti e documenti e tra documenti e topic.
Un'esigenza più volte manifestata dagli utenti di ItalGiure (magistrati e avvocati) consiste nel poter disporre facilmente di un elenco di sentenze rilevanti per un dato tema. La funzione *Sentenze rilevanti* è un tentativo iniziale di dare una risposta a questa domanda. Essa cerca infatti le sentenze rilevanti per un dato topic, ovvero, fissato il topic k, restituisce la lista dei documenti che presentano un valore $\hat{\theta}_{dk}$ superiore a una soglia prestabilita. Per default la ricerca è su tutto il corpus ma si ammette la possibilità che possa essere limitata soltanto ai documenti di una determinata classe di materia fra quelle disponibili.
Nella pratica questa funzione si rivela anche un utile strumento per "illuminare" seppur indirettamente il significato di un topic. Dall'esame dei testi delle sentenze rilevanti per quel topic si riesce infatti il più delle volte a comprenderne il contenuto semantico meglio di quanto sia non sia possibile fare direttamente. Più avanti nel paragrafo 3.2 discuteremo un caso concreto che chiarirà la questione (Figura 3.8).
La funzione *Sentenze simili* (Figura 3.5) è a ragione una delle funzioni più importanti di `Suprema` in quanto affronta il problema fondamentale (nella pratica giurisdizionale) della ricerca di sentenze simili a una sentenza assegnata.
In generale due sentenze d_1 e d_2 sono simili se condividono gli stessi topic. Nel contesto di un'analisi di tipo LDA, la similarità tra d_1 e d_2 può dunque essere misurata dalla similarità tra le corrispondenti distribuzioni dei topic $\widehat{\boldsymbol{\theta}}_{d_1}$ e $\widehat{\boldsymbol{\theta}}_{d_2}$.
Cioè, in sostanza, dalla similarità tra due distribuzioni di probabilità discrete. Quest'ultima a sua volta può essere misurata in molti modi diversi.
Seguendo Steyvers & Griffiths (2006), la funzione *Sentenze simili* implementa la classica divergenza (simmetrica) di Kullback-Leiber (KL). Se p e q sono due distribuzioni di probabilità discrete, essa è definita come:

$$\begin{aligned} KL(p,q) &= \frac{1}{2}[D(p,q) + D(q,p)], \text{ dove} \\ D(p,q) &= \sum_{k=1}^{K} p_k \log \frac{p_k}{q_k} \end{aligned}$$

La divergenza KL è non negativa e pari a zero soltanto quando p e q sono identiche, cioè $p_k = q_k$ per tutti i k. Si osservi tuttavia che essa non è propriamente una distanza in quanto non soddisfa la proprietà triangolare. Data la natura vettoriale dei $\widehat{\boldsymbol{\theta}}_d$ si potrebbero applicare misure geometricamente motivate come la distanza

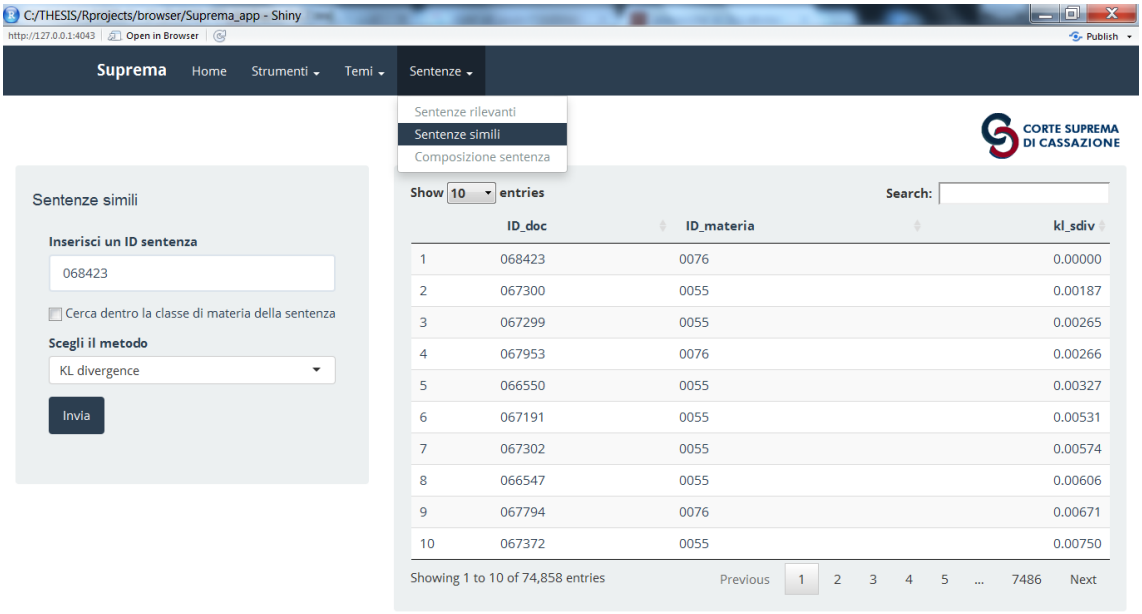


Figura 3.5: Sentenze identiche classificate sotto materie differenti (errori nella base dati di ItalGiure)

euclidea ma in questo caso sarebbe necessario tener conto delle restrizioni imposte dal vincolo della somma unitaria.
Data la sua importanza, il problema della ricerca di sentenze simili a una sentenza data è affrontato anche da ItalGiure che adotta un approccio basato sui punteggi *tfidf* di rilevanza di un termine per un documento: due sentenze sono tanto più simili quanti più termini rilevanti condividono (in Figura 3.6 un esempio estratto da ItalGiure dove, in rosso, sono evidenziati dal sistema i termini rilevanti condivisi da sentenze simili).
A differenza di quello seguito da `Suprema` si tratta di un approccio non probabilistico che non sempre dà buoni risultati al punto che il suo utilizzo viene sconsigliato.
Da un confronto parziale tra i due su casi reali i risultati ottenuti con `Suprema` si sono dimostrati nettamente migliori.
In molti di questi casi, dove i documenti erano identici (o quasi) `Suprema` e ItalGiure hanno avuto lo stesso comportamento, dove il secondo falliva la prima è riuscita a trovare sentenze simili (si veda la Tabella 3.3 riportata a titolo di esempio e nella quale vengono elencati i valori della divergenza KL per `Suprema` e le etichette Si/simile, No/dissimile per ItalGiure).
La questione meriterebbe comunque maggiori approfondimenti che soltanto una serie di esperimenti *ad hoc* condotti da esperti magistrati potrebbero fornire.

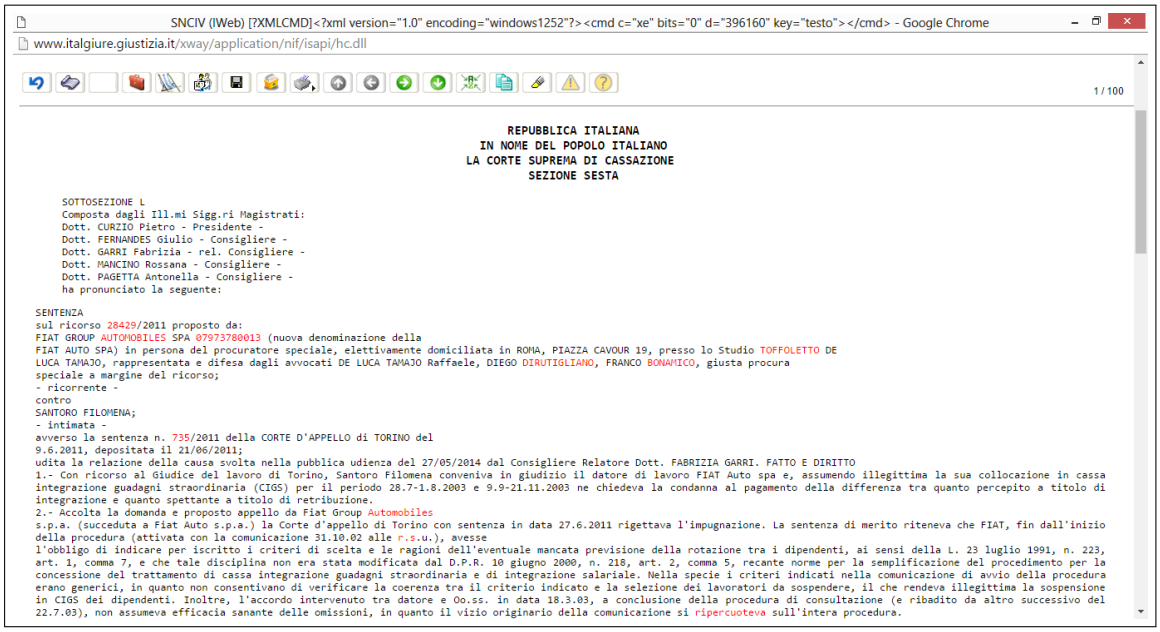


Figura 3.6: La ricerca di sentenze simili (ad una sentenza data) in ItalGiure

Id doc	Suprema	Italgiure
066231	0.00	Si
071382	0.00	Si
042144	0.00	Si
042147	0.00	Si
066230	0.00	Si
043170	0.00	Si
043178	0.00	Si
043179	0.00	Si
043185	0.00	Si
043187	0.00	Si

(a) Le prime 10 sentenze simili alla sentenza ID-066231

Id doc	Suprema	Italgiure
022770	0.00	Si
007589	4.70	No
028698	4.99	No
046252	5.37	No
021430	5.52	No
072605	5.53	No
015498	5.55	No
002820	5.89	No
038540	5.90	No
064179	6.31	No

(b) Le prime 10 sentenze simili alla sentenza ID-022770

Tabella 3.3: `Suprema` *vs* ItalGiure nella ricerca di sentenze simili

Si tenga conto inoltre che grazie a `Suprema` siamo riusciti a correggere facilmente alcuni errori presenti nella base dati di ItalGiure, come ad esempio il caso in Figura 3.5 di sentenze di fatto identiche (KL pressoché nulla) classificate tuttavia sotto materie differenti.

La funzione *Topic correlati* applica lo stesso procedimento della funzione *Sentenze simili* alle distribuzioni di probabilità dei topic $\widehat{\phi}_k$ restituendo questa volta una lista ordinata dei topic più simili (e in questo senso, correlati) a un topic fissato.

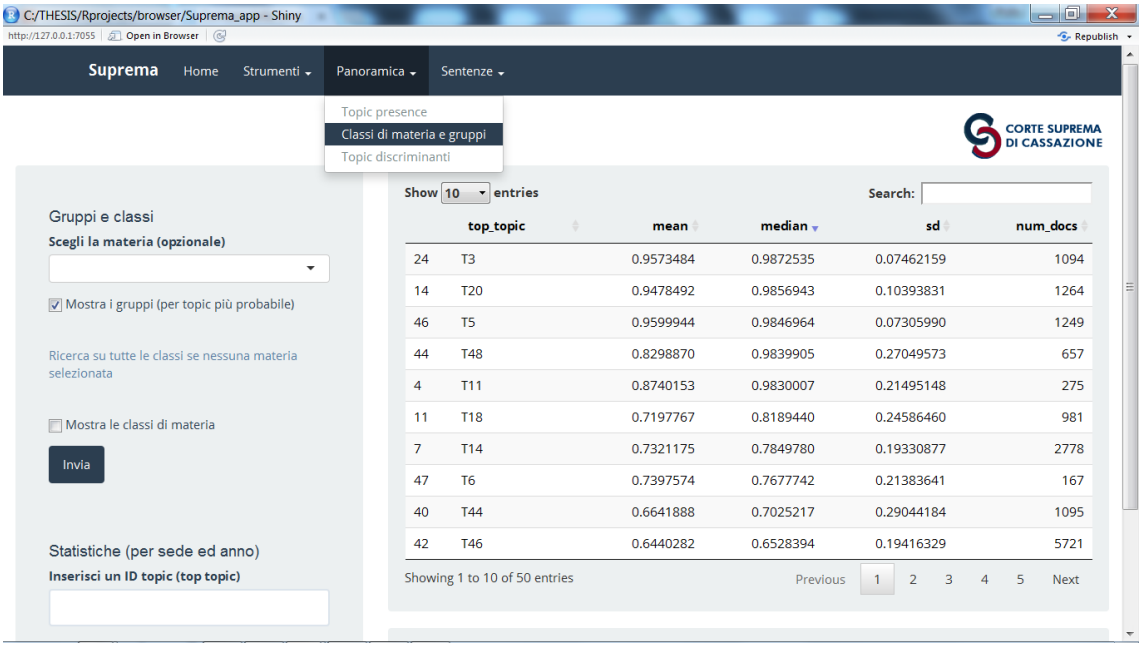


Figura 3.7: Funzione *Classi di materia e gruppi*

La funzione *Composizione sentenza* infine non fa che riprodurre la lista dei topic di ogni sentenza ordinati per valori decrescenti di probabilità.

3.2 Sentenze seriali: un'anomalia svelata da *Suprema*

Come abbiamo visto, la funzione *Classi di materia e gruppi* costruisce gruppi di documenti contrassegnati dall'etichetta del topic più probabile (d'ora in avanti chiamato *top topic*). L'elenco in Figura 3.7 si riferisce ad alcuni di questi gruppi, in particolare quelli che risultano essere caratterizzati da un valore mediano della probabilità del top topic superiore al 65%.
Si tratta nel complesso di 15.281 sentenze (su un totale di 74.858) almeno la metà delle quali molto simili, se non nella forma certamente nel contenuto, perchè "generate" in prevalenza da un solo tema (il top topic, appunto).
Detto altrimenti, `Suprema` ci consente di svelare molto facilmente una *anomalia* del nostro corpus che a sua volta riflette una peculiarità delle sentenze civili della Corte di Cassazione ben nota agli addetti ai lavori.
Molte di queste sentenze infatti appaiono assai simili tra loro, quando non perfettamente identiche (a meno delle generalità degli attori), perchè pronunciate a valle di ricorsi introduttivi, tutti dello stesso tenore, presentati *in serie* da gruppi di avvo-

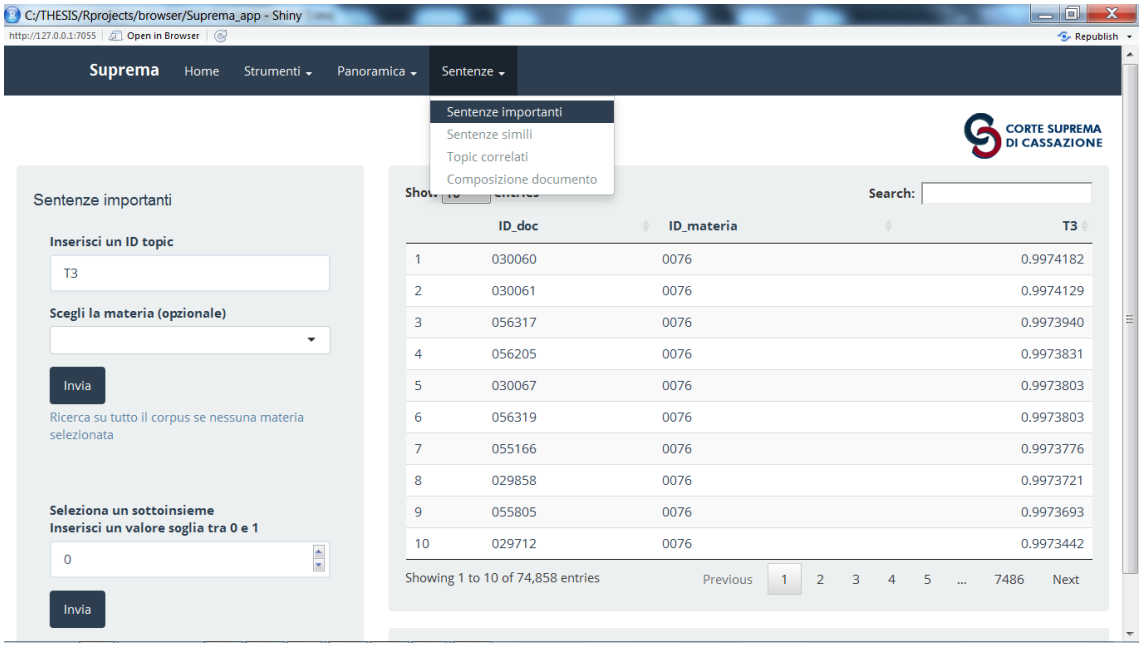


Figura 3.8: Sentenze rilevanti per il topic $T3$

cati organizzati[5] (per esempio quelli delle associazioni dei consumatori).
All'esame di alcuni di questi casi anomali dedicheremo le pagine successive, non senza aver prima sottolineato una volta di più che tutto ciò che diremo in proposito è stato ottenuto in modo automatico senza dover assumere alcuna specifica conoscenza giuridica del contenuto testuale dei documenti.
Rovesciando i termini della questione, sia questa un'occasione per sperimentare un possibile utilizzo di `Suprema` e riflettere su quali vantaggi, in casi come questi, potrebbero derivare all'organizzazione del flusso di lavoro della Corte di Cassazione (e in prospettiva di qualsiasi altro tribunale) dall'adozione di strumenti ispirati a `Suprema` come *filtri* iniziali per la distribuzione automatica dei ricorsi pendenti alle varie sezioni giudicanti.

[5]Si tratta di una pratica cui si è tentato di dare una diversa disciplina negli ultimi anni con l'introduzione anche nel nostro ordinamento dell'istituto della *class action*, molto utilizzato nel mondo anglossasone.

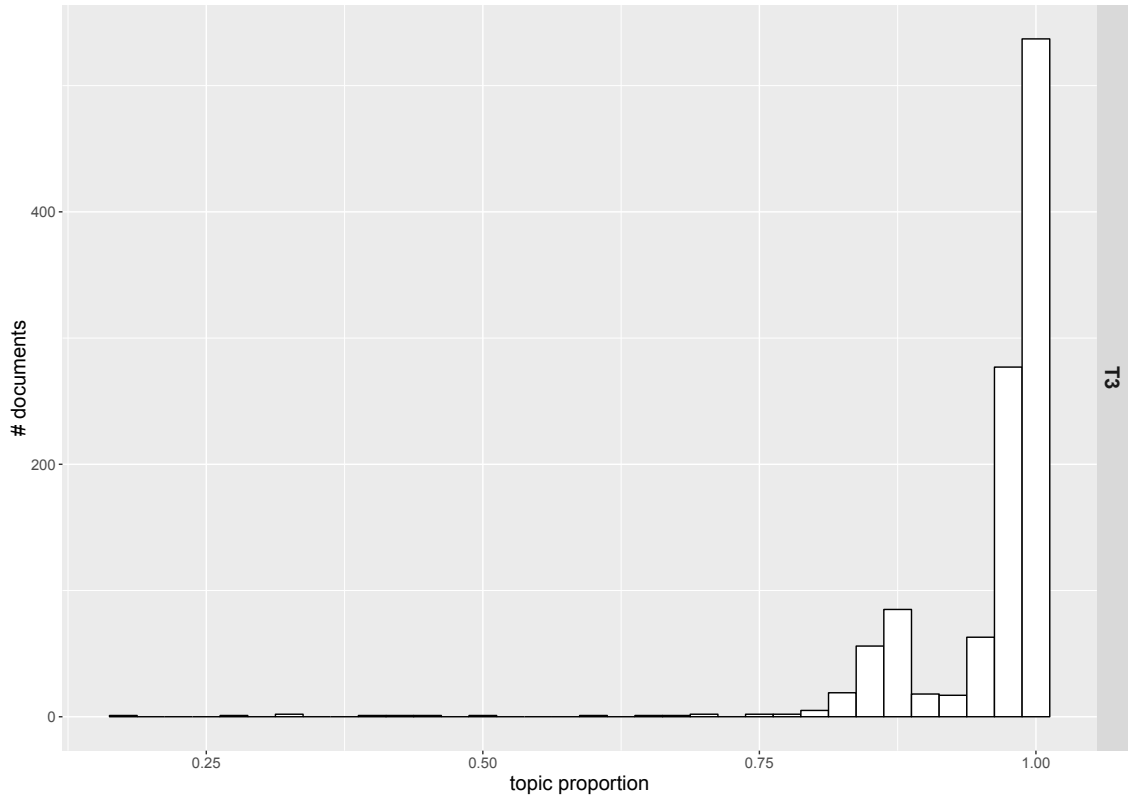


Figura 3.9: Istogramma (delle probabilità) del top topic nel gruppo $T3$

3.2.1 Il caso "Enel" e altri ancora

La Figura 3.9 che riporta l'istogramma (delle probabilità) del top topic nel gruppo di sentenze con etichetta $T3$ autorizza a pensare che tali sentenze siano nella gran parte sostanzialmente identiche: di fatto, una sola sentenza ripetuta a mò di fotocopia avente ad oggetto le stesse questioni di "trasferimento dei lavoratori"[6] come si può evincere facilmente dall'esame congiunto dei termini identificativi del topic $T3$ (in appendice) e dei testi di alcune sentenze rilevanti per quel topic (qui non riprodotti per questioni di spazio e tuttavia corrispondenti agli `ID doc` in Figura 3.8).
Continuando a scorrere l'elenco in Figura 3.7 e svolgendo le medesime argomentazioni i gruppi con etichette $T20$ e $T5$ individuano situazioni, se vogliamo, ancora più nitide e che ci raccontano qualcosa di assai interessante sul funzionamento della giustizia nel nostro paese (è questa ovviamente una mia personale osservazione).

[6]Nello specifico, questioni concernenti il trattamento giuridico ed economico del personale amministrativo, tecnico ed ausiliario (ATA) della scuola trasferito dagli enti locali al Ministero in base alla legge 123/1999.

Essi contengono rispettivamente 1.264 e 1.249 sentenze su controversie contrattuali[7] con il fornitore del servizio di vendita e distribuzione dell'energia elettrica (Enel). Anche qui si tratta nei fatti della stessa sentenza[8] ripetuta molte volte, con notevole dispendio di energie.
Ciò che colpisce tuttavia, a differenza del caso precedente, è la forte concentrazione nel tempo e nello spazio dei ricorsi dai quali successivamente hanno avuto origine le sentenze. Le Tabelle 3.4b e 3.4a ottenute attivando la corrispondente funzionalità di `Suprema` contengono le distribuzioni, per sede di provenienza e anno di iscrizione del ricorso introduttivo, delle sentenze dei gruppi con etichette $T20$ e $T5$ (e probabilità del top topic maggiore del 75%).
Come appare evidente si tratta di sentenze relative a ricorsi concentrati in pochi anni (2011 e 2012) e pochi luoghi ben definiti (comuni della Campania e in minor misura della Calabria).
In aggiunta, analizzando la distribuzione per classe di materia, si scopre che su un totale di 1.202 sentenze in entrambi i casi, rispettivamente 1.196 e 1.140 sono classificate sotto la voce *Contratti*, come del resto era da attendersi, e le restanti, di fatto identiche alle precedenti, sotto voci diverse. Si tratta anche qui di un caso di errore nella base dati di ItalGiure svelato da `Suprema`.
Scorrendo la graduatoria si arriva al gruppo con etichetta $T48$ che comprende 657 documenti. Di questi, poco meno di 500 risultano essere di fatto identici in quanto *coperti* per una quota superiore al 75% dal topic $T48$. Quanto all'appartenenza di classe, si tratta di sentenze nella materia *Previdenza* a seguito di ricorsi contro l'INPS (Istituto Nazionale della Previdenza Sociale) per questioni di indennità di disoccupazione agricola. Quanto alla collocazione spazio-temporale dei ricorsi introduttivi, essa è concentrata nei soli anni 2009 e 2010 esclusivamente negli uffici giudiziari di Bari.
E così il gruppo $T11$ che contiene sentenze relative a questioni di cassa integrazione guadagni presso gli stabilimenti FIAT di Torino; non è casuale dunque che i corrispondenti ricorsi introduttivi provengano nella quasi totalità dagli uffici giudiziari di Torino.
Si potrebbe continuare di questo passo con gli altri gruppi dell'elenco e fare osservazioni analoghe alle precedenti, salvo il caso del gruppo $T46$, il più numeroso tra quelli

[7] Cause contro l'Enel per ottenere il risarcimento delle tasse postali per il pagamento delle bollette di energia elettrica (1 euro). A detta dei ricorrenti infatti, «ai sensi della Delibera 1999/200 dell'Autorità per l'Energia Elettrica ed il Gas (da ritenersi integrata nel contratto di somministrazione), l'Enel avrebbe dovuto offrire ai suoi clienti almeno una modalità gratuita di pagamento della bolletta». Da qui la richiesta di risarcimento.

[8] Su questioni delle quali è lecito dubitare che debbano arrivare fino al terzo grado di giudizio in Cassazione (*NdA*).

Sede	2009	2010	2011	2012	2013
Airola	0	0	135	171	0
Avellino	0	0	6	70	21
Benevento	0	0	26	89	0
Casoria	0	0	0	4	0
Castellammare di Stabia	0	0	19	0	0
Cervinara	0	0	2	0	0
Cinquefrondi	0	0	28	0	0
Crotone	0	3	95	84	5
Gragnano	0	0	0	5	0
Guardia Sanframondi	0	0	0	0	1
Lagonegro	0	0	3	3	0
Lamezia Terme	0	0	1	0	0
Manduria	0	0	0	0	1
Mercato San Severino	0	0	0	47	0
Napoli	0	5	68	185	0
Nola	0	0	1	6	0
Palmi	0	0	1	0	0
Sala Consilina	0	0	0	1	0
Salerno	0	0	0	1	0
Santa Maria Capua Vetere	0	0	0	107	0
TORRE ANNUNZIATA	0	0	0	3	0
VALLO DELLA LUCANIA	3	0	0	2	0
TOTALE: 1202	**3**	**8**	**385**	**778**	**28**

(a) Gruppo con etichetta $T20$

Sede	2010	2011	2012	2013
Airola	0	226	178	0
Avellino	0	69	85	34
Benevento	0	34	122	0
Casoria	0	0	8	0
Castellammare Di Stabia	0	20	22	0
Catanzaro	46	0	0	0
Cervinara	0	7	5	0
Crotone	1	40	29	0
Lagonegro	0	0	11	0
Mercato San Severino	0	0	42	0
Napoli	2	53	120	0
Nola	0	0	7	0
Potenza	0	0	4	0
Salerno	0	0	1	0
Santa Maria Capua Vetere	0	0	26	0
Torre Annunziata	0	0	5	0
Vallo Della Lucania	1	0	4	0
TOTALE: 1202	**50**	**449**	**669**	**34**

(b) Gruppo con etichetta $T5$

Tabella 3.4: Distribuzione delle sentenze per sede di provenienza ed anno di iscrizione del ricorso.

considerati, che si riferisce a sentenze relative all'applicazione della cosiddetta legge Pinto (equa riparazione per irragionevole durata del processo). Un ambito di per sé molto *tipico*, le cui sentenze cioè risultano essere caratterizzate da un numero assai esiguo di temi (si confronti al riguardo il grafico in Figura 3.4). Ciò che spiega la sua presenza nella tabella in Figura 3.7. Non mancano comunque anche qui gruppi di sentenze-fotocopia, facilmente estraibili grazie alle funzionalità di `Suprema`.

Capitolo 4

Gli scenari di sperimentazione: la DTM ridotta

In più occasioni abbiamo visto come il peso eccessivo del lessico dei motivi di diritto possa influenzare i risultati delle nostre analisi. Tanto più se la scelta che si è fatta a monte è quella di preferire nell'esplorazione del corpus i *fatti* al *diritto*. Ciò che imporrebbe l'emersione del lessico dei motivi di fatto su quello dei motivi di diritto. In questo capitolo utilizzeremo un metodo semplice ma efficace per ottenere questo obiettivo. Un metodo che consiste in una opportuna riduzione del numero di colonne della matrice documenti per termini di base.
Accenneremo inoltre in conclusione ad un'analisi condotta sulla sola classe di materia *Tributi*, allo scopo di simulare un possibile uso del modello LDA come filtro per la distribuzione automatica dei ricorsi pendenti in materia tributaria alle varie sezioni giudicanti della Corte Suprema di Cassazione.

4.1 Introduzione

Nei due capitoli precedenti abbiamo trattato esclusivamente il caso della DTM base e abbiamo analizzato le conseguenze che possono derivare dall'essere, quest'ultima, inevitabilmente caratterizzata da una presenza preponderante di termini appartenenti al lessico procedurale-giuridico. Termini, come abbiamo visto, collegati ai *motivi di diritto* piuttosto che ai *motivi di fatto* delle sentenze.
In molte applicazioni pratiche siamo tuttavia interessati più a quest'ultimi che non ai primi; detto altrimenti, più al profilo fattuale che non a quello legale delle sentenze.
Ciò è particolarmente vero quando l'obiettivo dell'analisi per temi sia appunto quel-

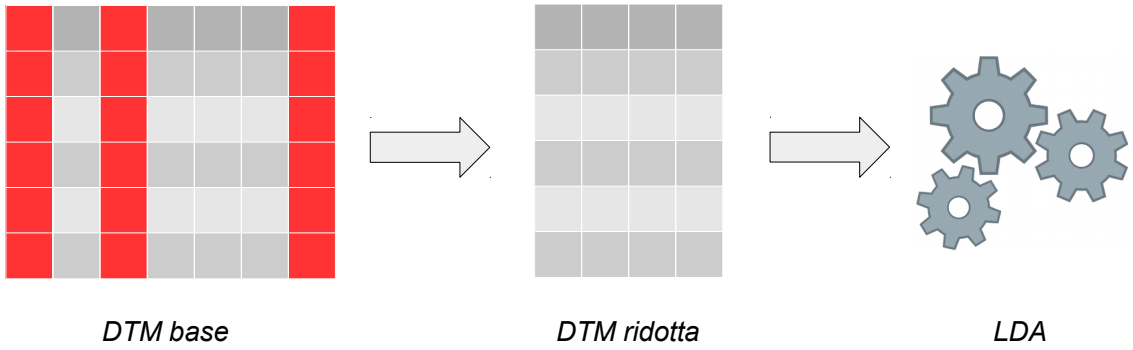


Figura 4.1: Dalla DTM base a quella ridotta come input di LDA

lo di individuare sottoclassi tematiche delle classi di materia con le quali vengono normalmente annotati i documenti dell'archivio delle sentenze civili di ItalGiure. Queste annotazioni riflettono le scelte operate dall'avvocato al momento dell'iscrizione del ricorso in Cassazione (ricorso sul quale, al termine del processo, deciderà la sentenza) e si riferiscono giocoforza a questioni di fatto che descrivono classi di materia molto ampie (si veda al riguardo la Tabella 2.2 del secondo capitolo), le quali, contrariamente a quanto accade per l'archivio delle *massime*, non vengono ulteriormente dettagliate a cura di uffici all'uopo dedicati (come ad esempio l'Ufficio del Massimario).

Ne segue che in questo modo l'archivio delle sentenze resta niente altro che un mero elenco di documenti privo di una adeguata struttura di catalogazione.

A tale proposito, risulta necessario ribadire qui ancora una volta che il nostro punto di vista deve essere quello dell'utente "CED" della Corte di Cassazione e non quello dell'utente "giudice", come invece negli esperimenti dell'ITTIG-CNR dei quali abbiamo già parlato, più interessato alla categorizzazione delle decisioni dal punto di vista legale.

Quanto detto impone di adottare strategie opportune per filtrare, riducendone il numero, le colonne della matrice DTM base con lo scopo di far emergere il lessico dei motivi di fatto su quello dei motivi di diritto, ancor prima di applicare il modello LDA. Come nello schema in Figura 4.1 (in rosso le colonne eliminate dalla DTM base).

Ebbene, abbiamo verificato sperimentalmente che una buona strategia di riduzione della matrice DTM base, nel senso auspicato, consiste nell'adottare lo schema *tfidf* già introdotto in precedenza. Alla sua descrizione sarà dedicato il paragrafo che segue.

Documenti	74.858
Words	**6.876.581**
Terms	125.185
Lunghezza media (terms)	42
Lunghezza media (words)	92
Lunghezza mediana (terms)	37
Lunghezza mediana (words)	73
Deviazione standard (terms)	24
Deviazione standard (words)	73

Tabella 4.1: Statistiche della DTM ridotta

4.2 La DTM ridotta

Un metodo tipico per operare una selezione dei termini-colonna della matrice DTM base consiste nell'attribuire loro un peso in base a qualche criterio, e nel tenere quindi soltanto quelli con peso entro un intervallo prefissato, scartando tutti gli altri.
Già conosciamo il punteggio *tfidf* (term frequency-inverse document frequency) definito come:

$$tfidf(d_i, t_j) = n_{ij} \cdot \log \frac{|\mathcal{D}|}{n_j} \tag{4.1}$$

dove $\mathcal{D}$ è l'insieme dei documenti del corpus, n_{ij} è la frequenza relativa del termine t_j nel documento d_i e n_j il numero di documenti che contengono il termine t_j.
Ora, calcoliamo il punteggio *tfidf medio* di ogni termine, $\bar{tfidf}(t_j)$, ottenuto cioè considerando la media dei punteggi di quel termine rispetto ai documenti che lo contengono:

$$\bar{tfidf}(t_j) = \frac{\sum_{i \in \mathcal{D}_j} tfidf(i, j)}{|\mathcal{D}_j|} \tag{4.2}$$

dove $\mathcal{D}_j$ è l'insieme dei documenti che contengono almeno un'occorrenza del termine t_j.
Nel nostro caso, abbiamo trovato che un'efficace selezione[1] delle colonne della matrice DTM base si ottiene eliminando tutte quelle con punteggio tfidf medio al di fuori dell'intervallo interquartile. Ciò consente infatti al lessico dei motivi di fatto di emergere su quello dei motivi di diritto, in quanto sia i termini rari (come ad esempio i nomi propri) sia quelli presenti in molti documenti (per lo più appartenenti al lessico dei motivi di diritto) vengono omessi.
A sostegno di quanto appena affermato, nelle Tabelle 4.1 e 4.2 e nell'istogramma in

[1] La funzione `reduce.dtm` del pacchetto `R Supreme` fornisce un utile interfaccia per questo compito.

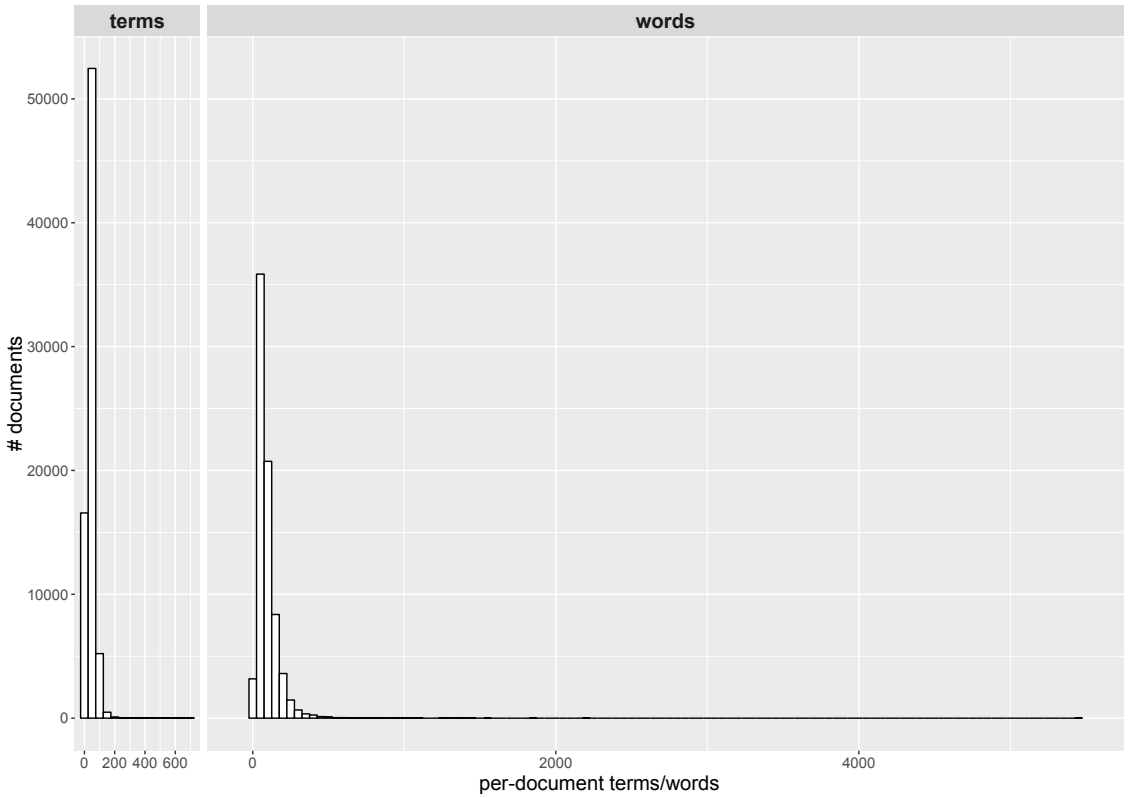


Figura 4.2: Distribuzioni dei documenti per numero di *terms* e *words* (DTM ridotta)

Figura 4.2 riportiamo le statistiche corrispondenti alla DTM ridotta e la lista dei suoi 50 termini più frequenti (da confrontare per curiosità con l'analoga del caso DTM base).
Si osservi comunque che il calcolo del punteggio tfidf medio ha semplicemente lo scopo di selezionare i termini del vocabolario; gli elementi della matrice DTM ridotta restano le frequenze dei termini nei documenti.
Rispetto al caso DTM base, i termini-colonna (terms) risultano ovviamente dimezzati mentre il numero delle loro occorrenze[2] (words) passa da 52.540.175 a 6.876.581. Una forte riduzione, cui segue tra l'altro un notevole miglioramento nelle prestazioni dell'algoritmo di stima (Gibbs sampling) che, come abbiamo visto nel primo capitolo, dipendono essenzialmente da questo numero.
Resta comunque il problema di valutare quanta informazione *utile* si perde nel passaggio dalla DTM base a quella ridotta. Validi suggerimenti al riguardo possono venire dal ripetere, nel caso della DTM ridotta, le medesime analisi condotte nello scenario di base in riferimento alla determinazione del valore di K.

[2]In altre parti di questo lavoro abbiamo usato il termine *token* con lo stesso significato.

Termine	Frequenza	Termine	Frequenza
contribuente	78.783	ccnl	21.028
agenzia	68.868	delib	20.993
durata	64.203	rimborso	20.431
lavoratori	46.507	licenziamento	20.260
tributaria	45.992	reddito	19.712
imposta	45.963	economia	19.600
entrate	43.868	finanze	19.436
indennità	39.597	notificazione	19.412
opposizione	39.491	prezzo	19.230
trasferimento	36.150	milano	19.193
fondo	32.578	direttiva	19.138
ragionevole	31.815	italiane	18.531
giustizia	31.794	lavori	17.798
integrazione	31.788	pensione	17.645
riparazione	29.191	distribuzione	17.632
ctr	29.036	anzianità	17.470
napoli	27.852	penale	17.211
equa	27.190	occupazione	17.154
fallimento	25.808	terreno	17.137
iva	25.304	giurisdizione	16.804
indennizzo	24.148	utenza	16.269
europea	23.926	mansioni	16.235
decadenza	23.332	irragionevole	15.489
inps	23.093	sospensione	15.002
banca	22.872	eredi	14.602

Tabella 4.2: I 50 termini più frequenti nella DTM ridotta

4.3 La selezione del modello e il confronto con lo scenario di base

In questo paragrafo discutiamo i risultati dell'applicazione al caso DTM ridotta della medesima sequenza di operazioni eseguite nel caso DTM base per la determinazione del numero dei topic K. Anche qui affrontiamo la questione rispettivamente come un problema di classificazione e uno di generalizzazione.
Oltre ciò tuttavia nostro precipuo interesse è anche valutare le differenze tra i due casi, come una sorta di maniera indiretta di soppesare quanta informazione utile si perde nel passaggio dall'uno all'altro.
Tutte le analisi che seguono sono state condotte nelle stesse identiche condizioni delle corrispondenti già discusse nel caso DTM base. Di diverso dunque c'è soltanto

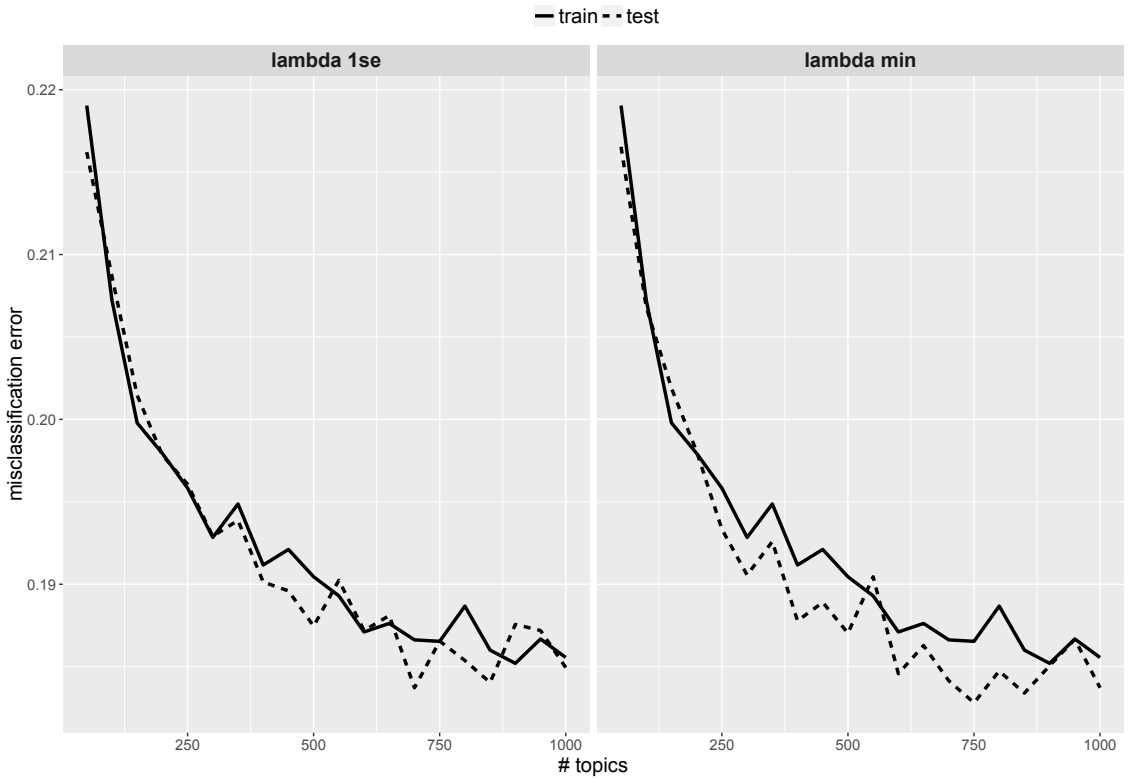


Figura 4.3: Errori di classificazione sul train e sul test set (10-fold cross-validation)

la matrice di input, lì di base e qui ridotta.
I grafici in Figura 4.3 riportano gli errori di classificazione complessivi (overall accuracy) calcolati come percentuale di documenti classificati correttamente sul totale dei documenti nelle due fasi di addestramento e di prova, al variare del numero dei topic K.
Come risulta evidente, l'andamento degli errori di classificazione è molto simile al caso analogo della DTM base. Il modello migliore, $K = 750$, corrisponde a un errore minimo sull'insieme di prova pari a $18,28\%$ (lambda minimo) di poco superiore al $16,68\%$ (lambda minimo) per $K = 950$ ottenuto nel caso DTM base.
Dunque, ciò che si può affermare è che al prezzo di un leggero aumento dell'errore di classificazione si ottiene una riduzione significativa nel numero dei topic del modello ottimale.
Considerazioni simili al caso DTM base si possono fare anche se si guarda alle matrici di confusione in Tabella 4.3 (per omogeneità con il caso DTM base abbiamo riportato anche qui il valore $K = 250$, pari al numero dei livelli di classificazione delle massime) e alla Tabella 4.4 con i valori dell'accuratezza bilanciata per classe.
Le classi relativamente ben classificate continuano a essere quelle più numerose, fatta eccezione per la classe *Contratti*. E questo anche se si considera un numero di

topic assai inferiore a quello ottimale. Esattamente come nel caso DTM base.

pred/truth	9	18	25	30	37	46	50	51	53	55	56	76	93	96	101	114	123	129	148	155
9	69	0	1	35	0	1	0	0	0	2	0	3	0	0	1	9	0	0	0	11
18	0	74	1	29	1	0	0	9	0	5	0	3	0	0	2	4	0	0	2	0
25	0	0	90	17	14	8	0	2	0	0	0	0	7	0	2	5	0	0	0	1
30	70	38	30	1919	105	18	9	70	23	46	12	58	8	9	47	219	11	32	23	173
37	2	1	28	54	316	40	0	4	5	3	0	3	59	0	5	10	0	2	1	20
46	1	0	3	3	31	42	0	0	1	0	0	0	4	0	0	5	0	1	0	0
50	2	2	1	29	6	0	2026	1	2	3	7	23	0	6	6	11	1	3	10	4
51	1	12	1	42	7	0	1	162	6	6	3	13	4	2	11	6	0	2	3	2
53	1	0	0	22	8	0	0	2	281	0	0	1	4	1	2	32	7	0	6	4
55	5	14	0	36	5	0	1	5	5	436	2	10	2	2	5	13	0	1	7	9
56	0	0	0	10	6	0	0	0	1	1	307	3	2	4	1	3	1	0	0	4
76	7	5	2	81	3	1	12	8	1	34	10	3657	0	192	13	56	12	8	20	4
93	0	0	0	7	28	1	0	0	0	1	0	0	36	0	0	1	0	0	1	0
96	1	0	1	11	0	0	1	0	2	2	4	210	0	713	0	3	1	1	2	0
101	0	0	0	9	0	0	0	2	0	0	0	0	0	0	5	0	0	0	0	1
114	11	5	4	167	14	10	2	3	7	2	1	39	1	16	14	633	5	3	0	7
123	2	0	0	6	2	1	0	1	4	0	0	2	1	1	1	11	86	0	2	0
129	4	2	0	26	2	1	1	4	1	0	0	6	0	1	1	9	0	269	3	0
148	0	3	1	54	5	1	3	4	5	13	3	21	1	6	2	13	7	12	4012	11
155	11	0	0	87	18	1	0	5	0	7	1	2	1	0	1	5	0	1	1	155

(a) Modello ottimale con 750 topic

pred/truth	9	18	25	30	37	46	50	51	53	55	56	76	93	96	101	114	123	129	148	155
9	80	0	3	47	1	3	0	0	0	3	0	2	0	0	3	10	0	0	0	9
18	0	64	1	25	0	0	0	1	0	5	0	6	0	0	3	3	0	2	0	3
25	2	0	88	18	20	5	0	1	1	0	0	1	4	0	3	8	1	1	0	1
30	57	53	26	1862	110	13	2	51	24	38	14	87	16	17	50	197	11	32	24	157
37	2	0	21	51	307	40	0	4	5	3	0	3	66	1	5	13	3	1	1	24
46	0	0	4	4	34	46	0	3	0	0	0	0	6	0	1	3	0	0	0	0
50	2	0	0	21	5	1	2036	0	1	3	5	16	0	9	2	15	2	0	12	3
51	0	11	1	46	4	1	0	191	9	4	2	13	3	2	9	5	0	1	3	1
53	1	0	0	27	8	2	0	1	277	0	0	1	3	0	2	32	9	0	5	2
55	8	17	1	49	6	0	0	4	3	451	1	20	1	1	5	16	0	2	11	13
56	1	0	0	17	6	1	1	1	1	3	312	9	0	5	4	6	0	1	1	5
76	5	6	3	99	6	2	3	12	3	21	10	3595	5	229	8	74	18	8	21	3
93	0	0	1	3	23	1	0	0	1	1	1	0	20	0	0	0	1	0	1	1
96	0	0	0	10	2	1	2	1	0	1	2	229	2	675	1	10	0	0	15	0
101	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
114	12	2	9	165	11	8	0	0	5	3	2	25	2	7	14	613	5	6	1	11
123	1	0	1	6	3	1	0	0	4	0	0	8	0	0	2	9	70	2	3	0
129	1	2	0	39	1	0	0	3	1	1	0	9	0	0	5	11	1	254	4	1
148	6	0	3	66	6	0	12	5	7	15	1	29	1	7	2	19	10	24	3989	11
155	9	1	1	87	17	0	0	4	2	9	0	1	1	0	0	4	0	1	2	161

(b) Modello sub-ottimale con 250 topic

Tabella 4.3: Matrici di confusione (lambda minimo) associate a LDA

ID materia	K = 750	K = 250	DTM base
9	0.6828	0.7117	0.6060
18	0.7357	0.7038	0.6977
25	0.7746	0.7682	0.7657
30	0.8317	0.8216	0.8675
37	0.7702	0.7621	0.7561
46	0.6667	0.6825	0.6391
50	0.9892	0.9922	0.9943
51	0.7839	0.8355	0.7201
53	0.9060	0.9001	0.8847
55	0.8852	0.8976	0.8996
56	0.9376	0.9440	0.9539
76	0.9350	0.9251	0.9504
93	0.6374	0.5760	0.6456
96	0.8674	0.8464	0.8856
101	0.5207	0.4999	0.5336
114	0.7932	0.7843	0.7965
123	0.8273	0.7661	0.7669
129	0.8998	0.8770	0.8664
148	0.9845	0.9796	0.9898
155	0.6870	0.6945	0.6669

Tabella 4.4: Valori dell'accuratezza bilanciata per classe

Così la Figura 4.4 che riporta i valori della *perlexity*, la misura della capacità di generalizzazione del modello già definita nel secondo capitolo in Eq. 2.5.
Anche qui l'andamento è del tutto simile a quello del caso DTM base e al crescere del numero dei topic cresce la capacità di generalizzazione del modello, ovvero la capacità di riprodurre documenti *nuovi* non considerati nella fase di addestramento. E anche qui l'andamento del grafico non dà indicazioni precise, che non siano quelle di aumentare sempre più il numero dei topic.
Se vogliamo, ancora una volta la conferma del fatto che dal nostro punto di vista la scelta di K deve essere affidata più a considerazioni di tipo pratico che teorico. Sostanzialmente basate sul soddisfacimento di esigenze di effettiva interpretabilità (e quindi comunicabilità) dei risultati piuttosto che sull'ottimizzazione di qualche criterio formale.
A titolo di esempio, il grafico in Figura 4.5 riporta i topic discriminanti di ogni classe di materia per $K = 50$. Da un suo rapido esame appare evidente come la classe *Equa riparazione* (ID 50) sia, tra le classi più numerose, quella caratterizzata dal minor numero di topic. Questo non deve sorprendere se si tiene conto del fatto che $K = 50$ è un valore relativamente basso e se si ricorda che questa classe si riferisce

a questioni molto tipiche (si potrebbe quasi dire standard).
Tuttavia, se si volesse comunque provare a dettagliarla ulteriormente basterebbe aumentare il numero dei topic. Sempre compatibilmente con esigenze di comunicabilità dei risultati ottenuti. La Figura 4.6 dovrebbe chiarire questo concetto.
In definitiva, se volessimo fare una sintesi di quanto visto in questo paragrafo dovremmo dire che nel passaggio alla DTM ridotta non si rinuncia poi a troppa informazione rispetto al caso base. Certamente si guadagna in migliori prestazioni dell'algoritmo di estrazione dei topic, sia in termini computazionali (tempi ridotti di più di 10 volte) che di qualità dei risultati ottenuti.

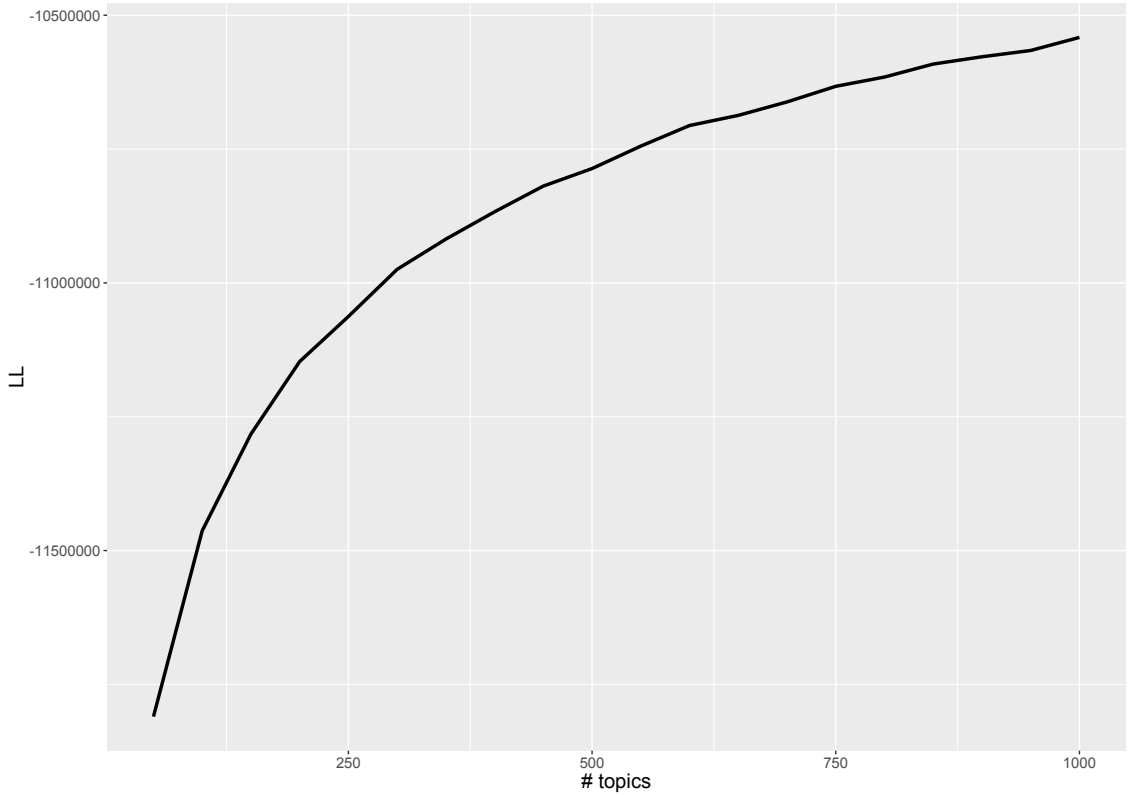


Figura 4.4: Perplexity sull'insieme di prova (DTM ridotta)

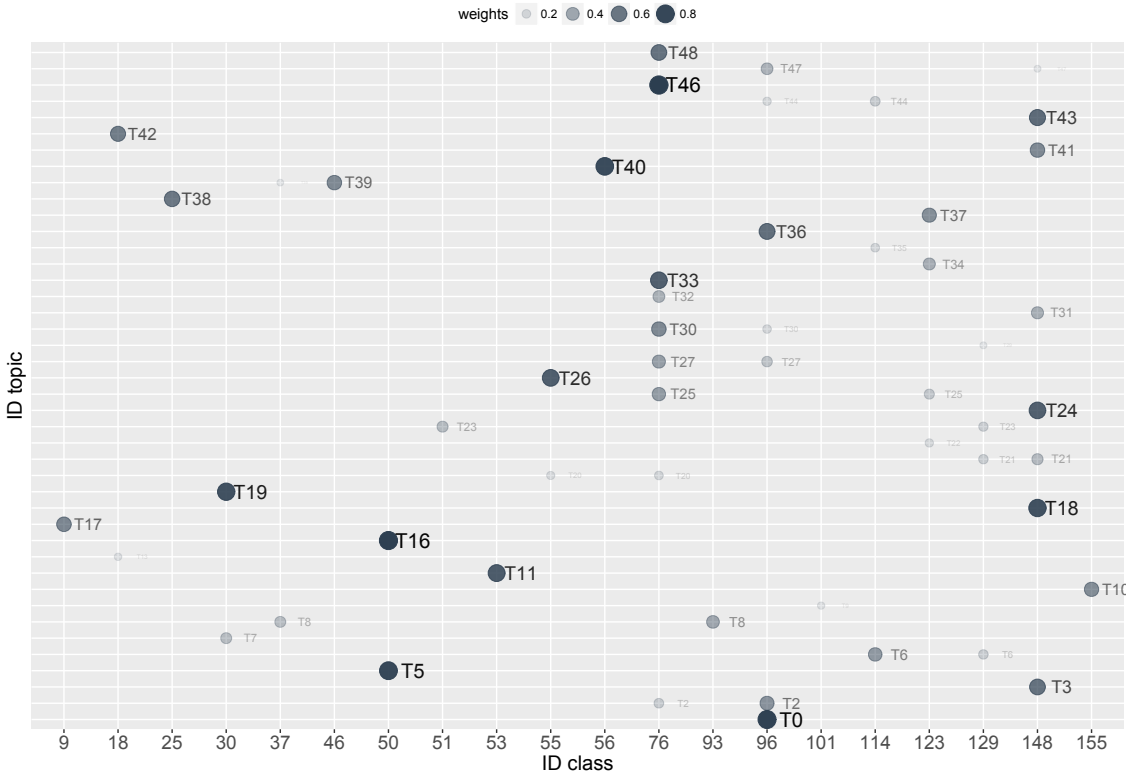


Figura 4.5: I topic caratteristici di ogni classe ($K = 50$ e soglia pari a 0.15)

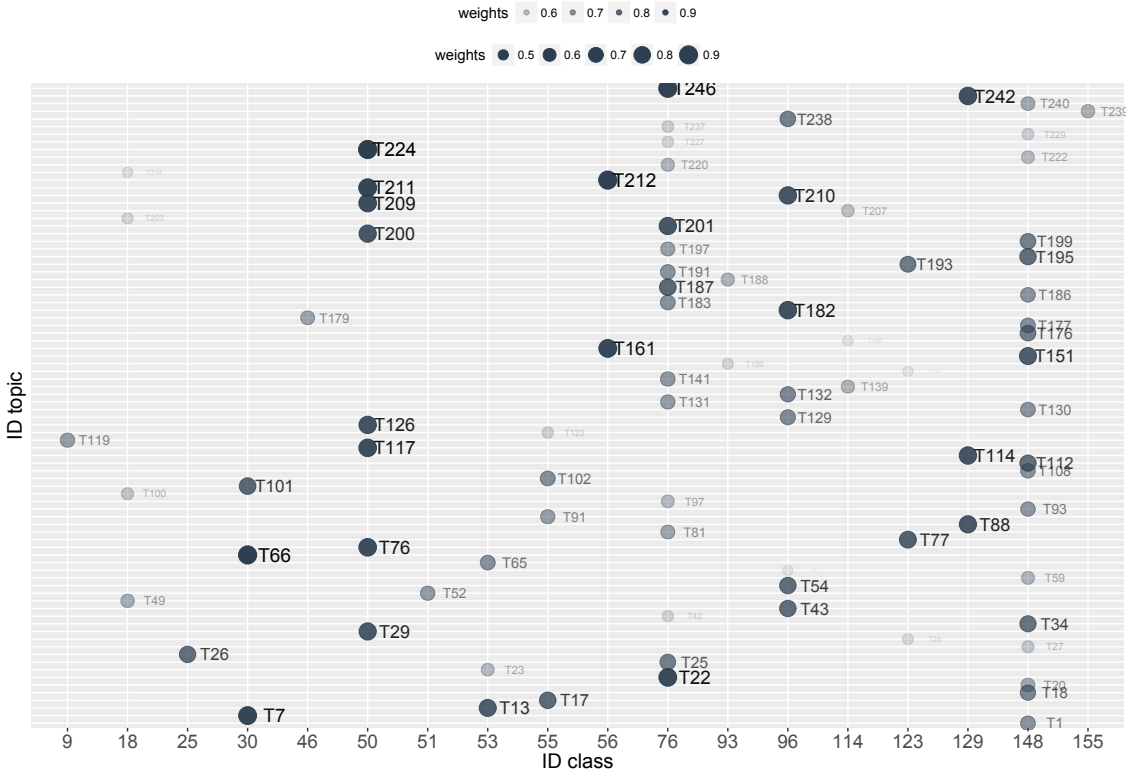


Figura 4.6: I topic caratteristici di ogni classe ($K = 250$ e soglia pari a 0.50)

4.4 Una proposta per la distribuzione dei ricorsi in materia tributaria

Tutte le analisi rese possibili dalle funzionalità di `Suprema`, alle quali abbiamo dedicato il terzo capitolo di questo lavoro, sono state ripetute anche per il caso DTM ridotta[3]. Ottenendo risultati che hanno confermato ancora una volta quanto affermato alla fine del paragrafo precedente.
Qui cambiamo pertanto prospettiva e ci concentriamo su un problema particolare: la distribuzione dei ricorsi pendenti in materia tributaria alle varie sezioni giudicanti della Corte di Cassazione.
Con l'unico obiettivo di mostrare quale contributo alla sua soluzione potrebbe derivare dall'adottare un approccio di tipo topic model.
Il normale corso dei processi davanti alla Corte Suprema di Cassazione prevede, come per qualsiasi altro tribunale, varie fasi: l'iscrizione dei ricorsi da parte degli avvocati ricorrenti, la loro assegnazione alle varie sezioni giudicanti, la discussione (il processo vero e proprio) e la decisione finale (sentenza, decreto od ordinanza) da parte di quest'ultime.
Uno dei fattori che più impattano sul rallentameno del flusso delle attività è l'assegnazione dei ricorsi pendenti[4] alle varie sezioni giudicanti, in genere specializzate per tipologie omogenee di cause.
Ciò è particolarmente vero quando tali ricorsi siano in numero assai elevato e soprattutto siano annotati in maniera molto generica nel registro delle iscrizioni.
È il caso dei ricorsi pendenti in materia tributaria. Ad oggi[5] pari a più di 100.000 e, cosa ancora più importante, annotati semplicemente con l'etichetta "Tributi".
Ne segue necessariamente che l'unico modo di distribuirli alle varie sezioni giudicanti consiste nel leggerne il contenuto e quindi in base a questo deciderne la destinazione. Un'attività in questo momento svolta manualmente da un ufficio dedicato, con enorme dispendio di tempo e di energie.
Come già accennato nelle pagine introduttive di questo lavoro, non appena il processo civile telematico (PCT) andrà a regime, tutte le comunicazioni tra parti e giudici (e dunque anche i ricorsi introduttivi) dovranno essere per legge in formato digitale.
A quel punto sarà interessante sperimentare l'efficacia di un approccio di tipo topic model pensato proprio come *filtro* iniziale per distribuire *automaticamente* i ricorsi pendenti alle varie sezioni giudicanti, sulla base del loro contenuto tematico.

[3] In appendice le Tabelle 4.11 e 4.12 contenenti la lista dei topic ordinati in base al valore della loro presenza nell'intero corpus ($K = 50$).

[4] Cioè, in attesa di giudizio.

[5] Dato aggiornato a Ottobre 2016.

Non disponendo per ora del formato digitale dei ricorsi pendenti in materia tributaria, simuleremo qui una situazione analoga estraendo dal nostro corpus le sole sentenze appartenenti alla classe di materia *Tributi* (la classe più numerosa con 16.375 documenti). E, sfruttando le funzionalità di `Suprema`, proveremo a filtrarle tematicamente come avremmo fatto se invece che sentenze fossero stati ricorsi introduttivi.
Negli esperimenti che seguono abbiamo esaminato due diversi scenari, entrambi a loro volta applicati sia alla DTM base che a quella ridotta: 1) modello addestrato su tutto il corpus e selezione dei risultati relativi alla sola classe *Tributi*, 2) modello addestrato sulle sole sentenze della classe *Tributi*.
Le Tabelle 4.5, 4.6 si riferiscono al primo dei due scenari di analisi e riportano la lista ordinata dei primi 10 topic più presenti nei documenti del corpus appartenenti alla classe *Tributi*. Rispettivamente nel caso DTM base e in quello DTM ridotta.
Così le Tabelle 4.7 e 4.8 per il secondo scenario di analisi (qui si è posto direttamente $K = 10$).
Queste tabelle sono qui riprodotte soltanto a titolo esemplificativo. Si tenga conto tra l'altro che sono state ottenute ipotizzando un numero di topic molto basso, pari a $K = 50$ nel primo scenario e $K = 10$ nel secondo, e che per ragioni di spazio sono stati considerati soltanto i primi 10 topic più presenti.
Si tratta quindi di risultati molto parziali, che hanno almeno il pregio di confermare quanto già sapevamo sulle caratteristiche delle due matrici DTM, base e ridotta.
Tuttavia, ciò che più importa, è la evidente capacità del modello LDA di far emergere la struttura tematica dei documenti.
Molti dei temi che li attraversano sono immediatamente interpretabili (per esempio, i topic $T24$, $T41$, $T43$, $T21$, $T9$, $T34$ nella Tabella 4.6; i topic $T8$, $T6$, $T5$, $T2$ nella Tabella 4.8) e si prestano a poter essere utilizzati come filtri allo scopo di fornire un ausilio nella individuazione di insiemi di documenti omogenei. I quali, nel caso fossero ricorsi introduttivi, potrebbero essere distribuiti alle varie sezioni giudicanti in maniera più efficiente di quanto non si faccia ora.
Come abbiamo visto, l'applicazione Suprema[6] fornisce utili interfacce per compiti come questi, alle quali rimandiamo per un esame più puntuale.
È fin troppo ovvio che la questione meriterebbe maggiori approfondimenti. In particolare, sperimentazioni su più larga scala e validazione dei risultati ottenuti da parte di gruppi di magistrati esperti. In ogni caso, riteniamo di aver predisposto, con il nostro lavoro, almeno gli strumenti essenziali per l'avvio di queste attività. Sulle quali il CED della Corte di Cassazione si è mostrato molto interessato ad investire tempo e risorse.

[6]Raggiungibile all'indirizzo https://cassazione.shinyapps.io/Suprema/.

T2 **tp = 0.159**	**T41** **tp = 0.149**	**T13** **tp = 0.089**	**T19** **tp = 0.078**	**T31** **tp = 0.065**
contribuente	agenzia	contribuente	motivo	roma
imposta	commissione	società	motivazione	ricorrente
agenzia	entrate	iva	violazione	persona
tributaria	accertamento	operazioni	ricorrente	avverso
entrate	contribuente	ufficio	merito	motivi
commissione	tributaria	accertamento	vizio	studio
lgs	regionale	agenzia	motivi	spese
pagamento	ufficio	ctr	parte	giusta
cartella	persona	imposta	censura	controricorso
avviso	generale	prova	impugnata	decisione
T1 **tp = 0.058**	**T47** **tp = 0.044**	**T24** **tp = 0.040**	**T17** **tp = 0.032**	**T43** **tp = 0.028**
comune	caso	lgs	termine	doganale
immobili	diritto	norma	notifica	autorità
classamento	parte	legge	atto	merce
rendita	fatto	disciplina	notificazione	origine
contribuente	essere	norme	parte	essere
immobile	senso	applicazione	udienza	dogane
atto	principio	disposizioni	impugnazione	società
commissione	stessa	vigore	essere	merci
catastale	pur	normativa	notificato	doganali
valore	ipotesi	disposizione	giorni	importazione

Tabella 4.5: Modello addestrato su tutto il corpus: i 10 topic più presenti nella classe *Tributi* (K = 50, DTM base)

T3 **tp = 0.222**	**T18** **tp = 0.196**	**T24** **tp = 0.072**	**T31** **tp = 0.059**	**T41** **tp = 0.055**
contribuente	contribuente	rendita	bologna	iva
agenzia	agenzia	contribuente	organizzazione	imposta
imposta	entrate	catastale	contribuente	fatture
tributaria	reddito	ici	agenzia	contribuente
entrate	tributaria	tributaria	irap	detrazione
cartella	ctr	agenzia	rimborso	fattura
ctr	redditi	imposta	entrate	agenzia
rimborso	soci	fabbricati	emilia	ctr
decadenza	imposta	unità	imposta	entrate
riscossione	ricavi	immobiliare	tributaria	cedente

T43 **tp = 0.033**	**T21** **tp = 0.031**	**T9** **tp = 0.027**	**T15** **tp = 0.022**	**T34** **tp = 0.020**
merce	tariffa	notificazione	milano	cooperativa
merci	tassa	domicilio	firenze	agricola
doganali	pubblicità	revocazione	genova	acque
dogane	tributaria	residenza	rossi	soci
importazione	contribuente	improcedibilità	perugia	fondo
agenzia	imposta	originale	mauro	consortile
certificati	superficie	ricevimento	luciano	acqua
contribuente	direttiva	raccomandata	angelo	coop
dogana	impianti	rinnovazione	bianchi	regione
dazi	urbani	eletto	eredi	cooperative

Tabella 4.6: Modello addestrato su tutto il corpus: i 10 topic più presenti nella classe *Tributi* (K = 50, DTM ridotta)

T9 **tp = 0.297**	**T1** **tp = 0.184**	**T2** **tp = 0.119**	**T6** **tp = 0.082**	**T0** **tp = 0.076**
agenzia	motivo	accertamento	comune	imposta
contribuente	motivazione	società	classamento	società
entrate	ricorrente	contribuente	rendita	valore
persona	violazione	ufficio	immobile	beni
tributaria	fatto	agenzia	immobili	fini
commissione	parte	entrate	contribuente	contratto
motivo	atto	reddito	atto	agenzia
roma	merito	essere	commissione	reddito
ricorrente	contribuente	soci	essere	base
avverso	giudizio	persona	catastale	essere

T3 **tp = 0.070**	**T8** **tp = 0.065**	**T7** **tp = 0.042**	**T5** **tp = 0.040**	**T4** **tp = 0.025**
lgs	attività	iva	consorzio	doganale
norma	contribuente	società	lavoro	autorità
diritto	organizzazione	operazioni	capitale	origine
essere	agenzia	imposta	fondo	merce
termine	irap	contribuente	bonifica	essere
legge	imposta	operazione	rapporto	merci
imposta	persona	fatture	lgs	doganali
amministrazione	lgs	detrazione	tassazione	società
applicazione	esercizio	diritto	contribuente	importazione
materia	generale	essere	parte	dogane

Tabella 4.7: Modello addestrato sui documenti di classe *Tributi*: topic ordinati per valori di presenza (K = 10, DTM base)

T9 **tp = 0.227**	**T1** **tp = 0.198**	**T3** **tp = 0.147**	**T4** **tp = 0.089**	**T8** **tp = 0.081**
contribuente	contribuente	imposta	agenzia	contribuente
agenzia	agenzia	contribuente	entrate	rendita
entrate	imposta	agenzia	notificazione	catastale
reddito	tributaria	tributaria	tributaria	tributaria
tributaria	entrate	entrate	contribuente	agenzia
ctr	cartella	organizzazione	societa	unità
soci	rimborso	registro	domicilio	ici
redditi	decadenza	ctr	fallimento	immobiliare
ricavi	ctr	irap	finanze	fabbricati
rettifica	riscossione	terreno	ctr	ctr

T6 **tp = 0.081**	**T0** **tp = 0.066**	**T7** **tp = 0.059**	**T5** **tp = 0.029**	**T2** **tp = 0.023**
iva	imposta	capitale	merce	dogane
contribuente	contribuente	tassazione	doganali	agenzia
imposta	esenzione	fondo	importazione	merci
agenzia	tributaria	agenzia	merci	dogana
fatture	pubblicità	redditi	certificati	energia
detrazione	tariffa	contribuente	agenzia	contribuente
ctr	aree	imposta	dogane	doganali
entrate	aiuto	tributaria	contribuente	merce
fattura	superficie	reddito	dazi	elettrica
cessione	agevolazione	entrate	certificato	consumo

Tabella 4.8: Modello addestrato sui documenti di classe *Tributi*: topic ordinati per valori di presenza (K = 10, DTM ridotta)

Appendice

T31 **tp = 0.076**	**T19** **tp = 0.060**	**T46** **tp = 0.057**	**T47** **tp = 0.047**	**T12** **tp = 0.038**
roma	motivo	durata	caso	giudizio
ricorrente	motivazione	processo	diritto	domanda
persona	violazione	euro	parte	grado
avverso	ricorrente	decreto	fatto	parte
motivi	merito	giudizio	essere	causa
studio	vizio	anni	senso	confronti
spese	motivi	ragionevole	principio	stata
giusta	parte	ministero	stessa	diritto
controricorso	censura	riparazione	pur	atto
decisione	impugnata	equa	ipotesi	motivo

T38 **tp = 0.038**	**T2** **tp = 0.037**	**T41** **tp = 0.034**	**T14** **tp = 0.031**	**T25** **tp = 0.030**
prova	contribuente	agenzia	termine	quesito
merito	imposta	commissione	contratto	diritto
motivazione	agenzia	entrate	lavoro	motivo
valutazione	tributaria	accertamento	contratti	motivazione
giudizio	entrate	contribuente	parti	bis
fatto	commissione	tributaria	rapporto	fatto
parte	lgs	regionale	poste	violazione
elementi	pagamento	ufficio	accordo	formulazione
fatti	cartella	persona	italiane	relazione
ricorrente	avviso	generale	motivo	sintesi

T17 **tp = 0.027**	**T24** **tp = 0.027**	**T21** **tp = 0.024**	**T27** **tp = 0.024**	**T0** **tp = 0.021**
termine	lgs	lavoro	spese	contratto
notifica	norma	rapporto	euro	immobile
atto	legge	lavoratore	interessi	vendita
notificazione	disciplina	licenziamento	somma	prezzo
parte	norme	ricorrente	liquidazione	parti
udienza	applicazione	motivo	giudizio	preliminare
impugnazione	disposizioni	società	pagamento	parte
essere	vigore	datore	motivo	locazione
notificato	normativa	violazione	importo	domanda
giorni	disposizione	giusta	parte	atto

T8 **tp = 0.021**	**T13** **tp = 0.021**	**T22** **tp = 0.018**	**T37** **tp = 0.017**	**T20** **tp = 0.017**
proprietà	contribuente	danno	comune	enel
condominio	società	responsabilità	giurisdizione	contratto
fondo	iva	risarcimento	consorzio	distribuzione
servitù	operazioni	danni	ente	servizio
parte	ufficio	assicurazioni	regione	energia
diritto	accertamento	fatto	violazione	elettrica
possesso	agenzia	motivo	amministrazione	sensi
motivo	ctr	sinistro	amministrativo	delib
essere	imposta	essere	essere	essere
domanda	prova	colpa	ricorrente	integrazione

T5 **tp = 0.017**	**T29** **tp = 0.016**	**T44** **tp = 0.015**	**T3** **tp = 0.015**	**T36** **tp = 0.015**
enel	inps	durata	trasferimento	napoli
servizio	istituto	giudizio	lavoratori	maria
utenza	lavoro	ragionevole	unione	giuseppe
pagamento	contributi	riparazione	posizione	ricorrenti
integrazione	diritto	equa	questione	antonio
contratto	lavoratori	decreto	direttiva	salvatore
contenuto	pensione	procedimento	giustizia	francesco
lett	previdenza	depositato	europea	palermo
modalità	sociale	termine	diritti	de
delib	delega	essere	cessionario	catania

Tabella 4.9: Topic ordinati per valori di presenza nell'intero corpus (K = 50, DTM base)

T7 **tp = 0.015**	**T18** **tp = 0.015**	**T35** **tp = 0.014**	**T9** **tp = 0.014**	**T26** **tp = 0.014**
lavoro	poste	ricorrenti	comune	procura
indennità	parti	maria	occupazione	atto
retribuzione	italiane	eredi	indennità	giudizio
contratto	conciliazione	domiciliati	valore	parte
rapporto	interesse	controricorrenti	area	atti
accordo	accordo	qualità	terreno	documenti
base	cessazione	de	espropriazione	documento
straordinario	contendere	attori	aree	difensore
ccnl	materia	giovanni	pubblica	produzione
compenso	roma	giuseppe	mq	fascicolo

T39 **tp = 0.014**	**T1** **tp = 0.013**	**T32** **tp = 0.013**	**T49** **tp = 0.013**	**T48** **tp = 0.013**
società	comune	lavori	banca	prescrizione
attività	immobili	contratto	credito	decadenza
azienda	classamento	società	pagamento	termine
impresa	rendita	vizi	conto	diritto
servizi	contribuente	opera	società	domanda
cooperativa	immobile	ricorrente	interessi	azione
gestione	atto	appalto	somma	parte
soci	commissione	pagamento	garanzia	prestazione
cessione	catastale	esecuzione	motivo	ricorrente
imprese	valore	opere	contratto	convertito

T40 **tp = 0.013**	**T45** **tp = 0.012**	**T28** **tp = 0.012**	**T42** **tp = 0.012**	**T10** **tp = 0.010**
incidentale	personale	fallimento	opposizione	assegno
principale	qualifica	fall	esecuzione	ricorrente
motivo	inquadramento	curatore	decreto	separazione
ricorsi	ccnl	fallimentare	cod	casa
roma	amministrazione	passivo	proc	coniugi
controricorso	profilo	procedura	provvedimento	coniuge
condizionato	economico	credito	ingiuntivo	mantenimento
motivi	trattamento	società	esecutivo	figlio
ricorrente	ministero	liquidazione	ordinanza	madre
violazione	posizione	curatela	titolo	figli

T30 **tp = 0.009**	**T23** **tp = 0.008**	**T4** **tp = 0.007**	**T43** **tp = 0.007**	**T6** **tp = 0.006**
inail	pensione	capitale	doganale	societa
consulenza	fondo	fondo	autorità	puo
ctu	trattamento	lavoro	merce	lavoro
consulente	legge	lgs	origine	gia
malattia	cassa	rapporto	essere	attivita
tecnica	diritto	tassazione	dogane	perche
ufficio	previdenza	somme	società	legittimita
grado	sistema	redditi	merci	nonche
lavoro	norma	reddito	doganali	impresa
conclusioni	pensionistico	fondazione	importazione	contratto

T15 **tp = 0.006**	**T33** **tp = 0.006**	**T16** **tp = 0.004**	**T34** **tp = 0.004**	**T11** **tp = 0.004**
penale	diritto	lavoro	lavoratori	integrazione
procedimento	azienda	rifiuti	criteri	comunicazione
commissione	università	motivo	procedura	lavoratori
decisione	ministero	parte	comunicazione	procedura
reato	medici	ricorrente	accordo	rotazione
aiuti	sanitaria	essere	scelta	criteri
diritto	prescrizione	lavoratore	sindacali	fiat
recupero	danno	obbligo	esame	accordo
aiuto	essere	datore	essere	cassa
civile	usl	motivazione	sindacale	esame

Tabella 4.10: Topic ordinati per valori di presenza nell'intero corpus (K = 50, DTM base)

T3 **tp = 0.053**	**T5** **tp = 0.052**	**T18** **tp = 0.045**	**T46** **tp = 0.043**	**T15** **tp = 0.040**
contribuente	durata	contribuente	italiane	milano
agenzia	riparazione	agenzia	indeterminato	firenze
imposta	indennizzo	entrate	lavoratori	genova
tributaria	equa	reddito	assunzioni	rossi
entrate	ragionevole	tributaria	conciliazione	perugia
cartella	economia	ctr	ccnl	mauro
ctr	finanze	redditi	fiorillo	luciano
rimborso	irragionevole	soci	mutuo	angelo
decadenza	europea	imposta	lavoratrice	bianchi
riscossione	onorari	ricavi	ferie	eredi

T16 **tp = 0.039**	**T19** **tp = 0.035**	**T28** **tp = 0.033**	**T4** **tp = 0.030**	**T6** **tp = 0.029**
durata	utenza	calabria	napoli	assicurazioni
ragionevole	integrazione	transazione	eredi	sinistro
riparazione	delib	catanzaro	salerno	stradale
equa	distribuzione	eredi	erede	compagnia
giustizia	energia	rosa	campania	strada
irragionevole	elettrica	reggio	morte	veicolo
indennizzo	utente	rinuncia	decesso	assicuratrice
europea	elettrico	angelo	esposito	conducente
mese	gratuita	paola	cuius	assicuratore
cedu	gas	teresa	gennaro	polizza

T23 **tp = 0.027**	**T0** **tp = 0.026**	**T9** **tp = 0.024**	**T48** **tp = 0.023**	**T26** **tp = 0.021**
opposizione	pensione	notificazione	licenziamento	fallimento
ingiuntivo	inps	domicilio	recesso	fall
pace	decadenza	revocazione	mansioni	passivo
esecutivo	previdenziale	residenza	lavoratori	curatore
sospensione	pensionistico	improcedibilità	lavoratrice	fallimentare
ingiunzione	disoccupazione	originale	indennità	curatela
pignoramento	anzianità	ricevimento	preavviso	creditori
esecutivi	indennità	raccomandata	mobilità	concordato
forzata	riccio	rinnovazione	ccnl	opposizione
procedente	pensioni	eletto	licenziamenti	fallita

T40 **tp = 0.020**	**T11** **tp = 0.019**	**T49** **tp = 0.019**	**T1** **tp = 0.018**	**T42** **tp = 0.017**
assegno	occupazione	bari	catania	banca
coniugi	indennità	lecce	messina	banco
separazione	terreno	potenza	brescia	mutuo
coniuge	espropriazione	cancellazione	ancona	ipoteca
figlio	aree	socio	cagliari	popolare
casa	mq	soci	sassari	fideiussione
figli	esproprio	puglia	siracusa	finanziamento
madre	terreni	taranto	bergamo	tasso
familiare	stima	vito	sicilia	cassa
moglie	fondo	registro	marche	risparmio

T8 **tp = 0.017**	**T24** **tp = 0.017**	**T17** **tp = 0.017**	**T10** **tp = 0.017**	**T31** **tp = 0.017**
fondo	rendita	lavori	prezzo	bologna
usucapione	contribuente	appalto	scrittura	organizzazione
terreno	catastale	progetto	trasferimento	contribuente
strada	ici	committente	immobiliare	agenzia
attori	tributaria	compenso	promittente	irap
divisione	agenzia	appaltatore	mutuo	rimborso
particella	imposta	incarico	simulazione	entrate
mappale	fabbricati	ditta	venditrice	emilia
confine	unità	ctu	acquirenti	imposta
terreni	immobiliare	appaltatrice	caparra	tributaria

Tabella 4.11: Topic ordinati per valori di presenza nell'intero corpus (K = 50, DTM ridotta)

T37 **tp = 0.016**	**T33** **tp = 0.016**	**T7** **tp = 0.015**	**T14** **tp = 0.015**	**T36** **tp = 0.014**
giurisdizione	trasferimento	locazione	venezia	inps
regione	direttiva	conduttore	trento	contributi
palermo	giustizia	locatore	provincia	iscrizione
associazione	lavoratori	affitto	trieste	lavoratori
sicilia	europea	prelazione	veneto	aiuto
co	unione	riscatto	bolzano	previdenziale
arbitrale	cessionario	fondo	verona	contribuzione
tar	anzianità	conduttrice	padova	contributivo
siciliana	cedente	comodato	manzi	cartella
caltanissetta	peggioramento	agraria	costa	opposizione

T25 **tp = 0.014**	**T41** **tp = 0.014**	**T44** **tp = 0.013**	**T13** **tp = 0.013**	**T2** **tp = 0.013**
mansioni	iva	medico	torino	malattia
inquadramento	imposta	sanitaria	agente	infortunio
dirigente	fatture	salute	agenzia	lavoratori
incarico	contribuente	sanitario	nato	mansioni
dirigenziale	detrazione	usl	piemonte	ferrovie
ccnl	fattura	asl	provvigioni	rete
dirigenti	agenzia	medici	incarico	ferroviaria
graduatoria	ctr	regione	provvigione	rendita
bando	entrate	sanitarie	preponente	infortuni
incarichi	cedente	ctu	investimento	ctu

T45 **tp = 0.013**	**T39** **tp = 0.013**	**T29** **tp = 0.012**	**T35** **tp = 0.012**	**T34** **tp = 0.012**
societa	confine	trasporto	penale	cooperativa
puo	muro	merce	reato	agricola
gia	fabbricato	consumo	cautelare	acque
attivita	distanze	industriale	sequestro	soci
perche	metri	trasporti	sospensione	fondo
legittimita	edificio	gas	querela	consortile
nonche	demolizione	accisa	falso	acqua
inammissibilita	costruzioni	porto	assoluzione	coop
nullita	ctu	dogane	stampa	regione
indennita	fondo	impianti	giornalista	cooperative

T20 **tp = 0.012**	**T30** **tp = 0.011**	**T32** **tp = 0.011**	**T47** **tp = 0.011**	**T38** **tp = 0.010**
straordinario	indennità	orario	fondo	condomini
aquila	ccnl	università	capitale	condominiale
ccnl	anzianità	ore	previdenziale	assemblea
compenso	comparto	riposo	contributi	appartamento
mensilità	inquadramento	compenso	contributo	edificio
operai	scuola	specializzazione	tassazione	delibera
tfr	stipendio	ferie	complementare	unità
tredicesima	temporizzazione	medici	pensione	parcheggio
indennità	stipendiale	corsi	capitalizzazione	condomino
orario	retributivo	turno	premi	immobiliare

T22 **tp = 0.010**	**T21** **tp = 0.009**	**T27** **tp = 0.009**	**T43** **tp = 0.008**	**T12** **tp = 0.007**
conti	tariffa	lavoratori	merci	cessione
bancari	tassa	cassa	merce	trasferimento
to	pubblicità	integrazione	doganali	ramo
dom	tributaria	fiat	dogane	cedente
elett	contribuente	avvio	importazione	cessionario
assegni	imposta	salariale	agenzia	fusione
versamenti	superficie	oo	certificati	avviamento
intestati	direttiva	sospensione	contribuente	somministrazione
amministratori	impianti	sospendere	dogana	cessioni
bancarie	urbani	crisi	dazi	lavoratori

Tabella 4.12: Topic ordinati per valori di presenza nell'intero corpus (K = 50, DTM ridotta)

Bibliografia

Agnoloni, T., Bacci, L. & Sagri, M. (2014), 'Tecniche di estrazione terminologica e classificazione automatica di corpora giurisprudenziali', *Informatica e diritto* **XXIII**(1), 41–64.

Bishop, C. M. (2007), *Pattern Recognition and Machine Learning*, 1st edn, Springer, chapter 9.

Blei, D. & Lafferty, J. (2009), *Text Mining: Classification, Clustering, and Applications*, Chapman & Hall/CRC Press, chapter Topic Models.

Blei, D., Ng, A. & Jordan, M. (2003), 'Latent dirichlet allocation', *Journal of Machine Learning Research* **3**, 993–1022.

Chaney, A. J. B. & Blei, D. M. (2012), Visualizing topic models, *in* 'ICWSM'.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. & Harshman, R. (1990), 'Indexing by latent semantic analysis', *Journal of the American Society for Information Science* **41**(6), 391–407.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977), 'Maximum likelihood from incomplete data via the em algorithm', *Journal of the Royal Statistical Society. Series B (Methodological)* **39**(1), 1–38.

Dumais, S. T. (1995), Latent semantic indexing: Trec-3 report, *in* 'Proceedings of the Text REtrieval Conference (TREC-3)', D. Harman, Ed., pp. 219–30.

Feinerer, I., Hornik, K. & Meyer, D. (2008), 'Text mining infrastructure in r', *Journal of Statistical Software* **25**(1), 1–54.

Friedman, J., Hastie, T. & Tibshirani, R. (2010), 'Regularization paths for generalized linear models via coordinate descent', *Journal of Statistical Software* **33**(1), 1–22.

Griffiths, T. & Steyvers, M. (2004), Finding scientific topics, *in* 'Proceedings of the National Academy of Sciences of the United States of America', Vol. 101, National Academy of Sciences, pp. 5228–5235. Suppl. 1.

Grün, B. & Hornik, K. (2011), '**topicmodels**: An r package for fitting topic models', *Journal of Statistical Software* **40**(13).

Hastie, T., Tibshirani, R. & Friedman, J. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edn, Springer-Verlag.

Hofmann, T. (1999), Probabilistic latent semantic indexing, *in* 'Proceedings of the Twenty-Second Annual International SIGIR Conference'.

McLachlan, G. J. & Krishnan, T. (1997), *The EM Algorithm and Extensions*, Wiley, New York.

Peruginelli, G. & Ragona, M. (2014), *L'informatica giuridica in Italia. Cinquant'anni di studi, ricerche ed esperienze*, number 12 *in* 'Collana: ITTIG. Serie Studi e documenti', Edizioni Scientifiche Italiane, Napoli.

Popesul, A., Ungar, L., Pennock, D. & Lawrence, S. (2001), Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments, *in* 'Uncertainty in Artificial Intelligence, Proceedings of the Seventeenth Conference'.

Salton, G. & Bucley, C. (1988), 'Term-weighting approaches in automatic text retrieval', *Inform. Process. Man.* **24**(5), 513–523.

Sebastiani, F. (2002), 'Machine learning in automated text categorization', *ACM Computing Surveys* **34**(1), 1–47.

Sievert, C. & Shirley, K. (2014), Ldavis: A method for visualizing and interpreting topics, *in* 'Proceedings of the Workshop on Interactive Language Learning, Visualization and Interfaces', Association for Computational Linguistics.

Steyvers, M. & Griffiths, T. (2006), *Latent Semantic Analysis: A Road to Meaning*, Laurence Erlbaum, chapter Probabilistic topic models, pp. 424–440.

Teh, Y., Jordan, M., Beal, M. & Blei, D. (2006), 'Hierarchical dirichlet processes', *Journal of American Statistical Association* **101**(476), 1566–1581.

Van den Boogaart, K. G. & Tolosana-Delgado, R. (2013), *Analyzing Compositional Data with R*, 1st edn, Springer-Verlag.

Wallach, H., Murray, I., Salakhutdinov, R. & Mimno, D. (2009), Evaluation methods for topic models, *in* 'ICML09: Proceedings of the 26th International Conference on Machine Learning', Association for Computing Machinery, ACM press, pp. 1105–1112.

Yao, L., Mimno, D. & McCallum, A. (2009), Efficient methods for topic model inference on streaming document collections, *in* 'Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '09, ACM, New York, NY, USA, pp. 937–946.