Modelli

Esistono diversi modelli di IRS ed ognuno di essi è costruito a partire da un particolare modello matematico. I principali modelli di IRS sono: il modello booleano, il modello vettoriale e il modello probabilistico. A partire da questi sono poi stati sviluppati diversi modelli più specifici.

Il modello booleano

Funzionamento, query ed RSV

Il modello booleano si basa sulla teoria degli insiemi e rappresenta l'importanza, o significatività, dei termini indice utilizzando dei pesi binari $w_{kj} \in \{0, 1\}$.

Un documento d_j è rappresentato dall'insieme dei termini indice t_k che occorrono all'interno di questo. Formalmente:

$$R(d_j) := \{t_k | w_{kj} = 1\}$$

Una query è definita come un'espressione costruita utilizzando gli operatori booleani (AND, OR e NOT) e consente di definire in modo preciso l'insieme dei documenti da selezionare. "Definire in maniera precisa" significa che, per un dato documento e una data query, il RSV vale o 0 o 1: un documento è rilevante oppure non lo è, non ci sono vie di mezzo. Formalmente, questo significa che il modello booleano modella la rilevanza di un documento come una proprietà binaria.

Il fatto che la significatività dei termini assume valori in $\{0, 1\}$ e il fatto che le query sono espressioni booleane, consentono di rappresentare il funzionamento del meccanismo di matching per questo modello con una sequenza di applicazioni di *operazioni insiemistiche* di unione e intersezione.

Una query booleana può sempre essere riscritta in *forma normale disgiuntiva*, o CNF, in cui ogni disgiunto rappresenta un insieme di documenti ideali. Un documento soddisfa una query se appartiene ad uno degli insiemi descritti dai disgiunti. È importante ricordare che l'*ordine di valutazione* di una query booleana è significativo e per questa ragione deve essere specificato.

Valutazione di query booleane

Il processo di valutazione di una query booleana può essere riassunto come segue:

1. si accede al file dizionario, ad esempio, nel caso di indici lineari, utilizzando la ricerca binaria;
2. si recupera la lista di posting corrispondente al termine;
3. il meccanismo di matching costruisce un albero binario di valutazione della query booleana;
4. valutazione lazy della query;
5. reperimento di tutti i documenti, vengono reperiti tutti i documenti che contengono una delle keyword

che corrispondono ad una delle foglie dell'albero, utilizzando il file inverted;
6. costruzione della lista dei documenti da ritornare.

Come detto in precedenza il meccanismo di matching può essere visto come l'applicazione di operazioni insiemistiche, in particolare:

- OR, rappresenta la costruzione di una lista costituita dall'unione delle liste dei sottoalberi destro e sinistro;
- AND, rappresenta la costruizione di una lista costituita dall'intersezione delle liste dei sottoalberi destro e sinistro;
- $BUT = AND\ NOT$, rappresenta la costruzione di una lista costituita dalla differenza tra le liste dei sottoalberi destro e sinistro.

Una query booleana può essere valutata utilizzando diverse modalità:

- full evalutation mode (visita ricorsiva in post-ordine), in cui si allocano in memoria le liste intermedie per i risultati dei nodi;
- lazy evalutation mode, in cui non vengono allocate le liste per i risultati parziali dei nodi.

Limiti

In questo paragrafo vengono elencati i principali limiti del modello booleano.

1. Essendo il meccanismo di matching basato su un criterio decisionale binario, il modello booleano non è in grado di produrre un ordinamento dei risultati. Questo perché tutti i documenti reperiti avranno lo stesso RSV, ossia 1*.
2. Essendo che l'RSV di ognuno dei documenti reperiti è uguale a 1, risulta impossibile limitare in maniera sensata il numero di risultati ritornati da una particolare query, nella pratica questo significa che molto spesso vengono ritornati o troppi risultati o troppo pochi risultati.
3. Le query booleane formulate dagli utenti sono spesso ambigue. Questo è fortemente legato alle difficoltà degli utenti in fase di formulazione delle query, causate ad esempio dal fatto che l'operatore di AND linguistico corrisponde all'operatore di OR logico.

Il modello vettoriale

Rilevanza e modalità di rappresentazione

Il modello vettoriale modella la rilevanza come una proprietà graduale dei documenti. Questo consente ai sistemi che adottano questo modello di operare un ordinamento dei risultati in funzione decrescente di rilevanza rispetto alla query, cosa impossibile per i sistemi che adottano il modello booleano.

Il modello vettoriale si basa sull'algebra lineare e rappresenta documenti e query all'interno di uno spazio vettoriale n-dimensionale, in cui n è il numero totale dei termini indice: un documento è rappresentato da un vettore di pesi w che assumono valori nell'intervallo $[0, \infty)$:

$$R(d_j) := \vec{d_j} = (w_{1j}, w_{2j} \dots w_{Ni})$$

Il modello vettoriale consente la formulazione di query espresse come una lista di parole. Questa lista di parole viene poi convertita dal sistema in un vettore di pesi: un peso assume valore 1 nel caso in cui il corrispondente termine dell'indice sia contenuto della query, 0 altrimenti.

Assunzioni

Il modello vettoriale si basa due diverse assunzioni molto importanti:

1. la rilevanza è un concetto graduale ed è proporzionale alla similarità tra il vettore che identifica un documento e il vettore che identifica la query, in termini formali $Rilevanza(d) \approx dim(d, q)$;
2. i termini sono reciprocamente indipendenti, questo significa che la presenza contemporanea di coppie o di più termini nei documenti non è correlata in alcun modo.

Versori e combinazioni lineari

Nel modello vettoriale, i termini indice vengono rappresentati come una delle coordinate dello spazio n-dimensionale considerato (come un versore). I vettori che rappresentano i diversi termini indice saranno quindi tutti linearmente indipendenti e formeranno una base ortonormale per lo spazio. Questo è diretta conseguenza dell'assunzione di indipendenza tra termini, che come detto in precedenza, è alla base di questo modello. Questo significa anche che ogni vettore dello spazio è una combinazione lineare degli n vettori associati ai termini indice ossia: ogni query e ogni documento può essere visto come una combinazione lineare dei vettori associati ai termini indice. Formalmente, l'r-esimo documento d_r può essere rappresentato come un vettore documento

$$\vec{d_r} = \sum_{i=1}^{N} w_{ir} \vec{t_i}$$

Spazio dei documenti e densità

Un termine viene definito *buon separatore* dei documenti che lo contengono da a quelli che non lo contengono se la sua selezione porta ad un aumento della distanza media tra i documenti della collezione, ossia alla produzione di uno spazio dei documenti meno denso. Viceversa, un termine con frequenza alta in tutti i documenti della collezione non è un buon indice per lo spazio dei documenti in in quanto la sua selezione come nuova coordinata dello spazio rende tutti i documenti più vicini tra loro, aumentando la densità dello spazio. Questo principio viene formalizzato dalla definizione di potere discriminante dei termini indice:

$$D_j = Den - Den_j$$

dove Den e Den_j sono le densità dello spazio prima e dopo l'assegnamento del termine t_j, che a loro volta sono definite come

$$Den = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{k=1,\, i \neq k}^{N} sim(d_i, d_k)$$

Un termine j viene considerato *buon indice* per lo spazio se $D_j > 0$ o cattivo indice se $D_j < 0$.

Pesatura delle query

Per quanto riguarda invece i pesi associati ai termini contenuti da una query, esistono diverse strategie per il loro calcolo:

1. si associa un peso uguale a 1 a tutti i termini della query;
2. si calcola la norma del vettore query come $q = w_{iq}\sqrt{L}$;
3. si utilizza la definizione di Salton e Buckley:

$$w_{i,q} = (0.5 + \frac{0.5\, freq_{i,q}}{max_l\, freq_{i,q}}) \times log\frac{N}{n_i}$$

Cosine-similarity

Esistono diverse definizioni alternative di similarità, di cui la più nota è la cosine-similarity*, definita come

$$sim(d_i, q) = \frac{\sum_{i=1}^{N} w_{ki} w_{kq}}{\sqrt{\sum_{k=1}^{N} w_{ki}^2 \sum_{k=1}^{N} w_{kq}^2}}$$

* il coseno è definito come $cos\ \alpha = (x * y)/(|x| * |y|)$

Vantaggi e svantaggi

Il modello vettoriale porta numerosi miglioramenti rispetto al modello booleano grazie alla pesatura dei termini, consentendo il reperimento anche documenti che approssimano le condizioni espresse dalla query grazie al confronto parziale e permettendo di definire un ordinamento dei documenti in funzione del grado di similarità alla query. Tuttavia anche questo modello presenta delle significative limitazioni, causate principalmente dalla mancanza di fondamento dell'assunzione di indipendenza tra termini, dal fatto che i termini non presenti all'interno di una query possono comunque andare a influenzare il retrieval e, infine, dalla insufficiente espressività del linguaggio di query adottato.

Per superare quest'ultima limitazione e in generale per ottenere dei risultati migliori, alcuni sistemi cercano di unire i due modelli utilizzando le query booleane e applicando ai risultati ottenuti da queste un ordinamento definito da un criterio di ranking basato sul modello vettoriale (es. SIRE).

Ottimizzazioni

In fase di implementazione del modello vettoriale, al fine di evitare di applicare la funzione di similarità prescelta a tutti i documenti presenti nella collezione, sono state definite alcune regole euristiche che

consentono di compire delle ottimizzazioni:

1. calcola il valore della funzione di similarità solo per i documenti che contengono molti o tutti termini termini della query (almeno uno);
2. calcola il valore della funzione di similarità solo per i documenti che contengono un termine con $IDF > s$, questo consente di evitare posting list particolarmente lunghe;
3. precalcola la lista dei documenti in cui un termine ha peso alto nel dizionario, detta champion list, e considera l'unione delle champion list per ogni termine della query;
4. effettua un clustering dei vettori documento e il calcolo dei centroidi: la valutazione di una query partirà dai centroidi e successivamente verranno considerati solamente i documenti appartenenti ai cluster migliori.

Il modello probabilistico

Probability Ranking Principle

Il modello probabilistico è invece costruito a partire dal Probability Ranking Principle:

"The best retrieval effectiveness can be achieved when documents are ranked in decreasing order of their probabilities of being judged relevant to the user. The above probabilities should be estimated as accurately as possible on the basis of whatever data has been made available for this purpose.".

In altri termini, data una query utente esiste un insieme di documenti rilevanti che la soddisfano e la query è una descrizione delle proprietà di questo insieme.

Funzionamento

Nel modello probabilistico la rilevanza è rappresentata come un concetto binario, un documento può essere rilevante o non rilevante, esattamente come visto per il modello booleano, e viene stimata a rispetto alle query.

Il modo in cui il modello determina l'insieme dei documenti rilevanti per una query può essere scomposto in due fasi principali:

1. viene ipotizzato l'insieme ideale di documenti;
2. questo insieme-ipotesi viene raffinato in modo iterativo.

In passato, gli IRS basati sul modello probabilistico selezionavano in maniera casuale l'insieme iniziale dei documenti; i metodi probabilistici più recenti stimano invece un peso per ogni termine nella query e utilizzano i termini con peso maggiore per definire l'insieme iniziale. Il peso rappresenta la probabilità del termine di reperire un documento rilevante: cresce con il numero di occorrenze del termine nei documenti rilevanti e decresce con l'occorrenza nei documenti irrilevanti.

Binary Independence Model

Il Binary Independence Model è un modello probabilistico che utilizza le due fasi presentate in precedenza:

1. ipotesi iniziale, si stima la probabilità che un documento sia rilevante, dove l'occorrenza nel documento di un termine della query indica che quel documento è rilevante rispetto alla query;
2. Apprendimento, la stima iniziale della probabilità di rilevanza di un documento viene migliorata a partire dalle informazioni acquisite (feedback di rilevanza), ossia alla luce dei documenti che l'utente stesso ha identificato come rilevanti, andando a verificare se il termine compare all'interno di questi.

Questo modello si basa sull'assunzione di indipendenza dei termini, il che permette di semplificare notevolmente i calcoli necessari per stimare la probabilità di rilevanza dei documenti (*Binary Independence Assumption*).

Rilevanza, ranking e RSV

Il modello probabilistico rappresenta un documento o una query come un vettore di pesi indicanti la presenza/assenza dei termini indice, come anche il modello vettoriale. Un sistema che utilizza il modello probabilistico deve stimare la probabilità di rilevanza di ogni documento nella collezione rispetto ad una query: un documento d è rilevante se

$$P(R|q, d_j) > P(\overline{R}|q, d_j)$$

La rilevanza non può però essere stimata direttamente. Per poterla stimare è prima necessario stimare $P(R|q, d_j)$.

Per stimare $P(R|q, d_j)$ si applica il teorema di Bayes:

$$P(R|d_j) = \frac{P(R) * P(d_j|R)}{P(d_j)}$$

e si considera il ranking definito come

$$\frac{P(R|q, d_j)}{P(!R|q, d_j)}$$

Il Retrieval Status Value è quindi definito come

$$RSV = log \prod_{i=1, t_i \in q} \frac{P(d_j|R)}{P(d_j|\overline{R})} \approx \sum_{i=1, t_i \in q}^{t} w_{ij}(log \frac{P(w_{ij} = 1|R) * (1 - P(w_{ij} = 1|\overline{R}))}{P(w_{ij} = 1|\overline{R}) * (1 - P(w_{ij} = 1|R))})$$

Il valore di $P(w_{ij} = 1|R),\ P(t_i|R)$ e $P(w_{ij} = 1|\overline{R}),\ P(t_i|\overline{R})$ può essere calcolato basandosi sull'assunzione iniziale: - $P(t_i|R) = 0.5$, uguale per tutti i termini o determinato in fase di training; - $P(t_i|\overline{R}) = n_i/N$, distribuzione analoga a quella nella collezione.

Processo iterativo

Le operazioni svolte a fronte di una query da un IRS che adotta il modello probabilistico sono le seguenti:

1. considerando i pesi binari, si reperiscono i documenti che contengono i termini nella query;
2. si calcola un ordinamento calcolando il RSV;
3. si applica un processo iterativo di miglioramento dell'ordinamento iniziale.

Il processo iterativo consiste in una revisione delle stime $P(t_i|R)$ e $P(t_i|\overline{R})$:

$$P(t_i|\overline{R}) = \frac{|V_i|}{|V|}$$

$$P(t_i|R) = \frac{n_i - |V_i|}{N - |V|}$$

dove V è il sottoinsieme dei documenti inizialmente reperiti, ossia quelli più rilevanti secondo il sistema, e V_i è il sottoinsieme di V di documenti che contengono t_i.

Vantaggi e svantaggi

I due principali vantaggi del modello probabilistico sono rappresentati dal fatto che consente di produrre un ordinamento dei documenti basato sulla probabilità di rilevanza, ottimo dal punto di vista dell'utente, e dal fatto che incorpora un meccanismo di feedback di rilevanza, ottimo dal punto di vista del sistema.

Lo svantaggio fondamentale di questo modello è rappresentato dal fatto che, affinché questo possa funzionare, è necessario stimare le probabilità a priori $P(t_i|R)$.

Inoltre è importante sottolineare che il BIM opera senza tenere in considerazione i fattori tf e idf.

Linguaggi di query

Espressioni regolari

Le espressioni regoli sono un linguaggio che consente la composizione di pattern complessi a partire dalla combinazione di pattern semplici:

- un singolo carattere è un'espressione regolare;
- un'espressione regolare più complessa può essere costruita in maniera induttiva utilizzando gli operatori binari di unione e concatenazione o l'operatore unario di ripetizione.

Alcune delle query costruite utilizzando questo linguaggio possono richiedere calcoli particolarmente costosi. Ad esempio, nel caso di dizionari che utilizzano l'ordinamento alfabetico, la query "comp*" può essere valutata in maniera molto efficiente a differenza della query "*comp", il cui calcolo risulta significativamente costoso.

Adiacenza

Alcuni linguaggi di query supportano l'utilizzo degli operatori di adiacenza:

- $x\ adj\ y$, si richiede che x e y siano adiacenti tra loro;
- $x\ with\ y$, si richiede che x e y appartengano alla stessa frase;
- $x\ same\ y$, si richiede che x e y appartengano allo stesso paragrafo.

Al fine di poter valutare query che utilizzano operatori di adiacenza, è necessario che le posizioni di ogni keyword facente parte del documento vengano memorizzate all'interno del file inverted.

La valutazione di una query che utilizza un'operatore di adiacenza prevede che vengano reperiti i documenti e le posizioni dei singoli termini nella fase e, una volta identificate le occorrenze dei termini nella frase, viene verificata la contiguità delle posizioni.

Per ottimizzare la valutazione di query che utilizzando operatori di adiacenza è meglio iniziare il controllo dei termini meno frequenti nella frase.

Prossimità

Gli operatori di prossimità sono simili a quelli di adiacenza. Questi operatori consentono di specificare una lista di termini e una distanza d: un documento soddisferà l'operatore se tutti i termini appartenenti alla lista compaiono con distanza massima d tra di essi all'interno del documento.

Ad esempio, il documento "I cani iniziano la parte finale della gara" soddisfa un query che utilizza un operatore di adiacenza del tipo $([cani, gara], 5)$ ma non del tipo $([cani, gara], 2)$.

Struttura dei documenti

Assumendo che tutti i documenti della collezione abbiano una struttura comune, a prescindere dal fatto che questa struttura sia un insieme o una gerarchia, alcuni sistemi permettono la formulazione di query che esprimono:

- condizioni sul contenuto, espresse tramite keywords o espressioni regolari e combinazioni con operatori booleani;
- condizioni sulla struttura, come l'esistenza di specifici campi, contenimento delle condizioni sul contenuto in specifici campi, contenimento e prossimità tra campi.

La valutazione di query di questo tipo richiede di estendere il file inverted in maniera tale da includere l'ID delle sezioni o dei campi che compongono un documento.