Index (parte 2)

- reti sociali e popolarità dei link:
  - scienza sociale e bibliometria:
    - tecniche iniziali per web search prese da IR classica (occorrenze keyworks nel testo)
    - ma ci sono differenze che le rendono meno effettive: dinamicità, abbondanza, utenti malevoli, hyperlinks
    - web search = IR classica + analisi spettrale + segreto
      - analisi spettrale: importanza pagina determinata da struttura grafo di pagine
      - segreto: strategie motore di ricerca (per evitare che utenti si adattino per prendere la rilevanza)
    - hyperlinks: forniscono informazioni supplementari al testo normale che come qualità spesso supera il testo normale
    - web è esempio di rete sociale
    - teoria reti sociali applicabile a epidemiologia, spionaggio, citazione:
      - interessata a determinare proprietà legate a connettività grafo
      - proprietà più importanti:
        - centralità
        - co-citazione
        - prestigio

- centralità:
  - distanza $d(u, v)$ tra 2 nodi è il più piccolo numero di links per andare da u a v
  - raggio nodo u: $r(u) = max_v\{d(u, v)\}$
  - **centralità sfruttando raggio**: $u = min_u\{r(u)\}$
  - **in-degree centrality**: $u = \{max_v c(v)\}$ con $c(v) = inDeg(v)$
  - **out-degree centrality**: $u = \{max_v c(v)\}$ con $c(v) = outDeg(v)$
  - **closeness centrality**: $u = \{max_v c(v)\}$ con $c(v) = \frac{1}{\sum_{t \in V | t \neq v} d(v,t)}$
  - **betweenness centrality**: $u = \{max_v c(v)\}$ con $c(v) = \sum_{s,t \in V | s \neq t \neq v} \frac{\sigma_{st}(v)}{\sigma_{st}}$

- co-citazione:
  - se u cita v, w, v, w sono co-citati da u
  - se v, w sono co-citati da molti documenti, v, w sono correlati
  - calcolare E: matrice di adiacenza grafo citazione
  - calcolare E^T matrice trasposta
  - calcolare $E^T E$
  - possono essere usati per costruire clusters di pagine web

- prestigio:
  - ciascun nodo ha misura associata a prestigio: $p[u]$, il prestigio di tutti i nodi è vettore p
  - pagina v ha alto prestigio se tasso di visita è alto, cioè se ci sono molti nodi con tassi di visita alto che linkano v
  - ciascun nodo v ha la somma totale del prestigio di tutti i nodi u che hanno un collegamento a v: si computa nuovo mettore prestigio p'
  - $p' = E^T p$
  - metodo delle potenze (power iterations):
    - $p_0 = (1, \dots, 1)^T$
    - $p_{i+1} = E^T p_i$ **con** normalizzazione
  - il metodo tende a valore convergente per p, cioè l'autovettore principale della matrice E^T
  - **convergenza**: se grafo è non diretto, E^T è simmetrica e diagonalizzabile, è sufficiente trovare un unico autovettore principale
  - **miglioramenti**: fattore di attenuazione $p_{i+1} = \alpha E^T p_i$
  - nel grafo del web (grafo diretto), metodo difficilmente converge (non abbiamo di solito un unico autovettore principale)

- autovettori ed autovalori:
  - $Ap = \lambda p$
  - p: autovettore di A
  - λ: autovalore di A rispetto a p
  - autovettore di valore assoluto massimo è **autovettore principale**
  - computazione autovettore principale: metodo delle potenze (power iterations):
    - $p_0 = (1, \dots, 1)^T$

    - $p_{i+1} = Ap_i$ **con** normalizzazione
  - converge se c'è solo 1 autovettore principale e se A è diagonalizzabile
  - **teorema**: la sequenza p_i converge all'autovettore principale
  - **dimostrazione**

- web e popolarità dei link:
  - motori di ricerca ordinano rispetto a informazione che chiede utente e popolarità pagina
  - popolarità pagina secondo analisi dei link
  - passi:
    1. collezione pagine web
    2. estrazione grafo hyperlinks
    3. esecuzione algoritmo analisi link
    4. peso popolarità su ogni nodo del grafo
  - 2 categorie:
    - query independent
    - query dependent

- PageRank:
  - qualità pagina u determinata da mumero di link entranti in u e da qualità delle pagine che linkano u
  - concetto di **surfer random**: web surfer che clicca eternamente sui hyperlinks, in modo randomico e uniforme sceglie un link per andare alla prossima pagina
  - assunzione grafo diretto fortemente connesso
  - p_0 è probabilità di iniziare da nodo randomico, $\sum_u p_0[u] = 1$
  - calcolo probabilità di trovarsi in un nodo v dopo aver cliccato 1 volta $p_1[v]$
  - per raggiungere v il surfer doveva trovarsi in u che ha link uscente verso v: da E derivazione di L, ogni elemento della riga di E viene diviso per la somma degli elementi non nulli della riga di E (il numero di link uscenti del nodo preso in considerazione)
  - $p_0 = (\frac{1}{N}, \dots, \frac{1}{N})^T$
  - $p_{i+1} = L^T p_i$
  - se L è irriducibile ed aperiodica allora la sequenza p_i converge all'autovettore principale della matrice L^T
  - $p[u]$ si dice PageRank del nodo u
  - usiamo il metodo delle potenze per trovare l'autovettore principale
  - **teorema**: la sequenza $L^T p_0, L_2^T p_0, \dots$ si avvicina all'autovettore principale di L^T
  - **dimostrazione**
  - per viaggio infinitamente lungo del surfer, p è probabilità che surfer tocca ogni pagina
  - corrispondenza tra prestigio e PageRank, differenza sul fatto che il PageRank di un nodo è diviso tra i suoi archi uscenti, PageRank come flusso
  - **problemi**:
    - **dead ends**: nodi pozzo, perdita importanza
    - **spider traps**: tutti i link uscenti sono all'interno di gruppo, assorbono importanza
  - grafo del web non è fortemente connesso ed aperiodico: inseriamo finte transizioni a probabilità bassa
  - a ciascun nodo il surfer farà una scelta:
    - con probabilità d: salta ad una pagina randomicamente
    - con probabilità $1 - d$: salta alla prossima pagina dalla pagina corrente scegliendo un link uscente in modo randomico ed uniforme
  - d è damping factor

$$p_{i+1} = (1-d)L^T p_i + d \begin{bmatrix} \frac{1}{N} \cdots \frac{1}{N} \\ \vdots \ddots \vdots \\ \frac{1}{N} \cdots \frac{1}{N} \end{bmatrix} p_i = ((1-d)L^T + \frac{d}{N} 1_N) p_i$$

  - dato grande numero links non è possibile trovare soluzione diretta all'autovettore principale ma dopo limitato numero di iterazioni, i valori di rankings convergono a quelli stabili, quindi andare avanti non cambia i rankings
  - il metodo delle potenza (power iterations) può essere usato per computazione, nessuna normalizzazione è necessaria perchè i vettori sono probabilistici:
    - $p_0 = (\frac{1}{N}, \dots, \frac{1}{N})^T$
    - $p_{i+1} = L^T p_i$ **senza** normalizzazione
  - quando query è inserita, un meccanismo nascosto combina PageRank con rilevanza rispetto a query (IR)
  - problemi:

    - misura popolarità generica pagina -> soluzione: topic-specific PageRank
    - usa singola misura di importanza -> soluzione: hubs e authorities
    - suscettibile a link spam -> soluzione: TrustRank

- topic-specific PageRank:
  - invece di popolarità generica vogliamo popolarità all'interno di topic
  - risponde alla query dell'utente in base agli interessi dell'utente
  - surfer randomico si teletrasporta con probabilità secondo regola:
    - seleziona categoria in base alla query e alla distribuzione delle categorie utente
    - si teletrasporta ad una pagina dentro la categoria scelta
  - non si può computare PageRank a tempo di query
  - **offline**: PageRank computato per singole categorie: query independent, ciascuna pagina ha diversi punteggi PageRank (1 per categoria)
  - **online**: distribuzione pesi su categorie computate dal contesto della query: punteggio PageRank dinamico: somma pesata dei PageRank della categoria:
    - **input**:
      - grafo web W
      - vettore influenza v: (pagina: grado influenza)
    - **output**:
      - vettore ranking r: (pagina: importanza rispetto a v)
    - $r = PR(W, v)$
    - per insieme vettori influenza v_j

      $$\sum_j w_j \cdot PR(W, v_j) = PR(W, \sum_j w_j \cdot v_j)$$

    - w_j dovrebbe essere il peso sulla categoria computato dalla query: probabilità teletrasporto su quella categoria

- HITS:
  - alcune pagine web sono autorità su topic, ed altre sono hubs per trovare le pagine autorevoli
  - è query dependent a differenza di PageRank
  - **pagine autorities**: contengono informazioni buone
  - **pagine hubs**: contengono links a pagine autorities
  - buone autorities linkate da buoni hubs, buoni hubs linkano a buone autorities: definizione mutualmente ricorsiva
  - ogni pagina è sia hub sia authority: 2 punteggi per ciascun nodo:
    - $h[u] = \sum_{(u,v) \in E} a[v]$
    - $a[u] = \sum_{(v,u) \in E} h[v]$
  - matricialmente:
    - $a = E^T h$
    - $h = Ea$
  - i punteggi di tutti i nodi danno luogo a 2 vettori a,h
  - si può usare metodo delle potenze (power iterations) per risolvere iterativamente:
    - **inizio**:
      - $a_0 = (1, \ldots, 1)^T$
      - $h_0 = (1, \ldots, 1)^T$
    - **iterativamente**:
      - $a_{i+1} = E^T E a_i$ **con** normalizzazione
      - $h_{i+1} = E E^T h_i$ **con** normalizzazione
  - convergenza in 20, 30 iterazioni per migliaia di nodi e links
  - **passi HITS** (topic distillation):
    - query presentata a sistema IR e creazione root set
    - espansione root set e quindi creazione expanded set (base set = root set + expanded set)
    - esecuzione metodo delle potenze (power iterations) su hub e authorities simultaneamente
    - ritorno hub e authorities con punteggi più alti
  - **pro e contro**:
    - non precalcola hub e authorities perchè grafo G_q può essere ottenuto solo dopo che la query è conosciuta (query dependent)
    - HITS ha bisogno di meno espedienti rispetto a PageRank per ottenere ranking pagine (una volta che punteggi sono conosciuti)(è già rispetto ad una query, mentre PageRank descrive popolarità generica)
    - HITS deve calcolare autovettori per ogni query a differenza di PageRank (contro)

- **problemi**:
  - HITS colpito da effetto TKC: piccolo sottoinsieme di pagine in cui ogni hub ha link a ciascuna pagina authority (grafo bipartito):
    - questo piccolo sottoinsieme ha punteggi alti anche se pagine non sono autorevoli in materia
    - effetto sfruttato da spammers per incrementare peso delle loro pagine
    - TKC più grande nasconde quelli più piccoli perchè durante la normalizzazione dell'autovettore principale, i TKC piccolo diventano trascurabili
  - espansione root set aumenta recall ma diminuisce precision:
    - problemi di contaminazione con il topic specificato in query: possono cadere nel expanded set nodi che non c'entrano niente con la query
  - piccoli cambi del grafo del web hanno effetti drammatici sui punteggi di hub e authorities

- SALSA:
  - estensione probabilistica HITS
  - cerca di rimuovere anomalie di HITS
  - funzionamento: surfer randomico durante camminata randomica effettua scelte:
    - in dato nodo v, scelto in modo randomico ed uniforme un link entrante proveniente da nodo u, ci muoviamo su nodo u seguendo link all'indietro
    - al nodo u, scelto in modo randomico ed uniforme un link uscente verso dato nodo w, ci muoviamo su nodo w seguendo link in avanti
  - probabilità di transizione da v a w:

$$p(v,w) = \frac{1}{inDeg(v)} \sum_{(u,v),(u,w)\in E} \frac{1}{outDeg(u)}$$

  - viene mantenuta natura bipartita di HITS (effetto reciproco rafforzamento dei nodi nei grafi bipartiti è mantenuta)
  - persi di SALSA più robusti di pesi di HITS in presenza dell'effetto TKC:
    - anche perchè punteggio authority in SALSA è proporzionale al grado entrante
  - PageRank è più stabile rispetto ai cambi nel grafo del web (a causa edi salti randomici)
  - miglioramento algoritmo:
    - data probabilità fissata d, ad ogni step il surfer:
      - **con probabilità** d: salta ad un nodo random del base set
      - **con probabilità** $1-d$:
        - se step è **pari**: sceglie arco uscente random
        - se step è **dispari**: sceglie arco entrante random
  - algoritmo con salti randomici è molto più stabile rispetto ai cambiamenti del grafo del web
  - stabilità cresce quando d cresce
  - settare $d = 1$ inutile per ranking (punteggi tutti uguali)
  - non esiste ricetta per settare d sono vedendo struttura di grafo, d deve prendere in considerazione anche contenuto pagine

- spamming:
  - qualsiasi azione che ha fine di aumentare posizione pagina nei risultati **in modo spropositato** rispetto a vero valore pagina
  - **spam**: pagine web risultato dello spamming
  - 10-15% delle pagine web sono spam
  - **spammers**: persone con interessi commerciali che sfruttano motore di ricerca per portare persone su loro sito
  - tecniche spam: **term spam**, ripetizione parola target molte volte nella pagina web
  - **link spam**: strutture di link che aumentano PageRank su pagina
  - **spam farms**: strutture sviluppate per concentrare PageRank in una pagina
  - **link spamming**:
    - 3 tipi di pagine per lo spammer:
      - **pagine inaccessibili**
      - **pagine accessibili**: lo spammer può postare link a sue pagine
      - **pagine di proprietà**: pagine controllate da spammer
  - **link farms**:
    - obiettivo spammers: massimizzare PageRank di pagina target t
    - tecnica:
      - prendere più link possibili da pagine accessibili per mirare alla pagina t
      - costruire link farm per prendere l'effetto moltiplicatore del PageRank

- TrustRank:
  - **combattere link spam**:
    - rilevamento e blacklisting strutture che sembrano link farms
    - **TrustRank**: topic-specific PageRank con teleport set = trusted pages
  - **principio base**:
    - è raro per una buona pagina linkare una cattiva (spam)
    - campioniamo insieme seed pages dal web
    - oracolo identifica pagine buone da quelle spam in seed set
  - **trust propagation**:
    - **trusted pages**: set di pagine nel seed set identificate come buone
    - eseguiamo topic-specific PageRank con teleport set = trusted pages:
      - propagazione trust attraverso i link (ciascuna pagina ha $0 < trust < 1$)
      - soluzione: usiamo un valore soglia per marchiare tutte le pagine al di sotto ($<$) del valore di soglia come spam
    - settiamo trust di ciascuna pagina trusted ad 1
    - settiamo trust di ciascuna pagina untrusted a 0
    - trust di pagina p è t_p
    - pagina p ha insieme di link uscenti O_p
    - per ogni pagina $q \in O_p$, p conferisce trust a q
    - $$t_q = \sum_{(p,q) \in E} \frac{\beta \cdot t_p}{|O_p|}, 0 < \beta < 1$$
    - **trust è additivo**: il trust di p è la somma dei trust conferiti su p da tutte le pagine che linkano p(link entranti)
  - **trust attenuation**: il grado di trust conferito da trusted page descresce con distanza
  - **trust splitting**: trust è diviso tra link uscenti (più grande è il numero link uscenti meno controllo l'autore ha sui link)
  - **seed set**:
    - deve essere più piccolo possibile
    - deve garantire che ogni pagina prenda adeguato TrustRank (tutte le pagine buone devono essere raggiungibili da cammini corti)
    - come sceglierlo:
      1. **PageRank**: prendiamo le k pagine più in alto da PageRank
      2. usiamo domini trusted

- grafo del web:
  - interessante per:
    - analisi dei link per: **data mining**, determinazione di comunità
    - determinazione di modelli per: dimostrazioni di proprietà di algoritmi, **predizione evoluzione fenomeni**
  - struttura web:
    - tasso crescita: raddoppia ogni 15 mesi
    - interesse nell'identificare **non solo proprietà quantitative** ma anche proprietà legate a **connettività grafo**
  - **componente debolmente connessa**: insieme di pagine in cui ogni pagine è accessibile da tutte le altre seguendo gli hyperlinks in avanti e in indietro
  - **componente fortemente connessa**: insieme di pagine in cui per ogni coppia di pagine (u, v) esiste un path diretto da u a v
  - circa il 90% del web è una componente debolmente connessa
  - struttura:
    - **core centrale fortemente connesso** (SCC):
      - 30%: portali, motori di ricerca, siti di grandi compagnie
    - **sottografo IN** con rotte dirette verso SCC:
      - 24%, pagine che permettono di raggiungere nucleo ma non sono raggiungibili da esso: pagine personali, siti più piccoli
    - **sottografo OUT** con rotte che vanno all'esterno di SCC:
      - 24%, pagine che sono accessibili dal nucleo ma non sono connesse ad esso nella direzione opposta: pagine di università, compagnie, centri di ricerca
    - **tendrils** isolati attaccati ad uno dei 3 grandi sottografi
    - rimanente 22% pagine linkate tra loro ma completamente disconnesse dal resto del web