Tenzor link analysis

Bruno Fabulić, Helena Marciuš, Dora Parmać

6. ožujka 2021.

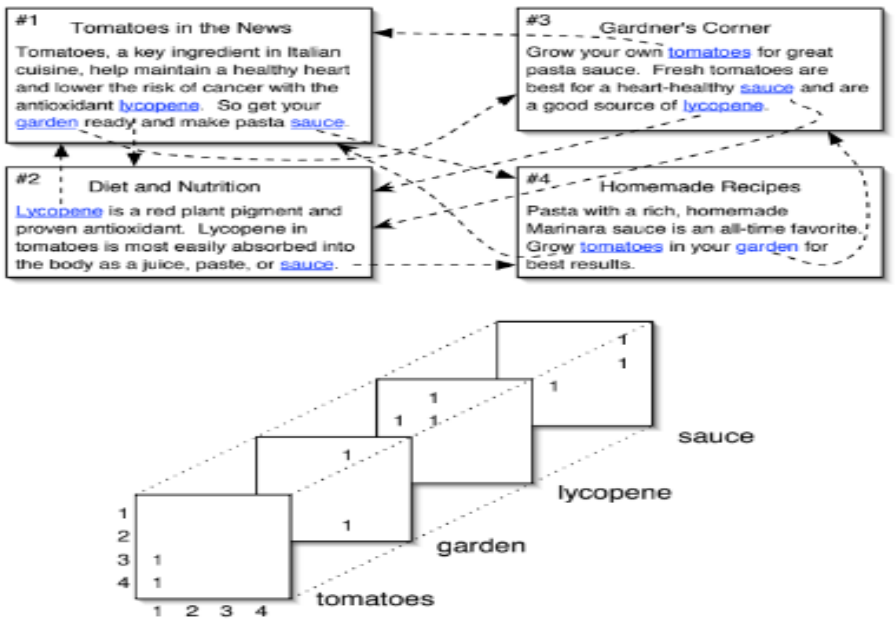
Sadržaj

1 Uvod

Kako se veličina Weba povećava, postaje sve važnije analizirati strukturu linkova uzimajući u obzir i kontekst. Multilinearna algebra pruža novi alat za obuhvaćanje teksta poveznice i drugim informacija koji se koriste u drugih metodama analize poveznica kao što je HITS algoritam.

Algoritam kojeg proučavamo naziva se Topical HITS ili TOPHITS. On je generalizacija Kleinbergovog HITS modela. TOPHITS dodaje 3. dimenziju za oblikovanje tenzora susjedstva koji uključuje i tekst poveznice. Ova dodatna informacija pruža način za obuhvaćanje konteksta uz izračunavanje autoriteta i hubova, što je postignuto 3D Parallel Factors (PARAFAC) dekompozicijom, više-dimenzionalni analogon SVD dekompozicije. Dodavanje 3. dimenzije znatno poboljšava primjenjivost HITS-a, zato što se TOPHITS analiza može izvoditi unaprijed i offline. Također, TOPHITS otkriva latentno grupiranje stranice i informacija.

U ovom seminaru, opisat ćemo brži matematički algoritam za izračunavanje TOPHITS algoritma na podatcima.



Slika 1: TOPHITS analizira 3D tenzor koji reprezentira kolekciju web stranica

2 Usporedba HITS i TOPHITS algoritma

Mnoge metode za analizu weba, poput PageRanka i HITS-a bazirane su na matrici susjedstva grafa kolekcije web stranica. PageRank rezultati dani su ulazima glavnog svojstvenog vektora Markovljeve matrice vjerojatnosti. S druge strane, HITS računa hubove i autoritete za svaki vrh i oni odgovaraju glavnom lijevom i desnom singularnom vektoru matrice susjedstva. Zanimljivo svojstvo HITS-a, što ne dijeli s PageRankom jest da se višestruki parovi singularnih vektora mogu razmatrati.

2.1 HITS

Neka je I broj stranica u našem podgrafu. Svaka stranica ima hub-score h i autoritet-score a koje računamo iterativno na sljedeći način:

$$\boldsymbol{h}_i^{(t+1)} = \sum_{i \to j} \boldsymbol{a}_j^t \tag{1}$$

$$\boldsymbol{a}_j^{(t+1)} = \sum_{i \to j} \boldsymbol{h}_i^{t+1} \tag{2}$$

Nakon svake iteracije, normaliziramo h i a. To znači da je hub-score stranice i jednak sumi autoritet-scorova svih stranica na koje pokazuje. Slično, autoritet-score stranice i jednak je sumi hub-scorova svih stranica koje pokazuju na nju.

Zapišimo to matrično. Neka je $I \times I$ matrica susjedstva $\boldsymbol{X}$ definirana kao:

$$X_{i,j} = \begin{cases} 1 & \text{ako stranica i pokazuje na stranicu j} \\ 0 & \text{inače} \end{cases} \tag{3}$$

Tada jednadžbe 1 i 2 postaju:

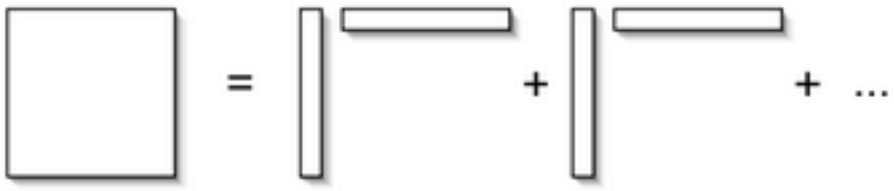
$$\boldsymbol{h}^{(t+1)} = \boldsymbol{X}\boldsymbol{a}^{(t)}, \boldsymbol{a}^{(t+1)} = \boldsymbol{X}^T\boldsymbol{h}^{(t+1)} \tag{4}$$

Prvih R faktora singularne dekompozicije of $\boldsymbol{X}$ proizvodi najbolju R-rang aproksimaciju, odnosno možemo aproksimirati $\boldsymbol{X}$ kao:

$$\boldsymbol{X} \approx \boldsymbol{H\Sigma A^T} \equiv \sum_{\boldsymbol{r=1}}^{\boldsymbol{R}} \boldsymbol{\sigma_r h_r} \circ \boldsymbol{a_r} \tag{5}$$

Vrijedi $\Sigma = diag\{\sigma_1 \dots \sigma_R\}$ i za singularne vrijednosti pretpostavljamo $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_R > 0$ i gdje su $\boldsymbol{h_r}$ i $\boldsymbol{a_r}$ odgovarajući singularni vektori. Matrice

$\boldsymbol{H}i\boldsymbol{A}$ su veličine $I \times R$ i imaju ortonormalna polja. To možemo zamisliti kao aproksimaciju matrice $\boldsymbol{X}$ sumom R 1 ranga vanjskog produkta, kako je prikazano na (2) Glavni par singularnih vektora $\boldsymbol{h_1}$ i $\boldsymbol{a_1}$ pruža hub i aurotitet rezultate za dominantnu temu u kolekciji web stranica. Drugim riječima, stranice koje imaju najveće rezultate $\boldsymbol{h_1}$ su najbolji hubovi za dominantnu temu, a stranice koje imaju najveće rezultate u $\boldsymbol{a_1}$ su najbolji autoriteti.



Slika 2: SVD dekompozicija

2.2 TOPHITS

Ilustrirajmo sada TOPHITS algoritam. On proizvede trojke $\boldsymbol{h_1}$, $\boldsymbol{a_1}$, $\boldsymbol{t_1}$ gdje $\boldsymbol{u}$ i $\boldsymbol{v}$ predstavljaju hub i aurotitet rezultate, a $\boldsymbol{t}$ term rezultate.

Slično kao HITS, rezultati se mogu dobiti iterativno. Neka je I broj stranica, a K broj termova. Tada je:

$$\boldsymbol{h}_i^{(t+1)} = \sum_{i \xrightarrow{\text{k}} j} \boldsymbol{a}_j^t \boldsymbol{t}_k^{(t)} \qquad \text{za} \quad i = 1, \ldots, I \tag{6}$$

$$\boldsymbol{a}_j^{(t+1)} = \sum_{i \xrightarrow{\text{k}} j} \boldsymbol{h}_i^{t+1} \boldsymbol{t}_k^{(t)} \qquad \text{za} \quad j = 1, \ldots, I \tag{7}$$

$$\boldsymbol{t}_k^{(t+1)} = \sum_{i \xrightarrow{\text{k}} j} \boldsymbol{a}_j^{t+1} \boldsymbol{h}_i^{t+1} \qquad \text{za} \quad k = 1, \ldots, I \tag{8}$$

Kao i kod HITS-a, normaliziramo vrijednosti nakon svake iteracije.

Zapišimo to u tenzor formi. Neka je $\boldsymbol{X}$ $I \times I \times K$ tenzor susjedstva podgrafa definiran kao:

$$X_{i,j,k} = \begin{cases} 1 & \text{ako stranica i pokazuje na stranicu j koristeći term k} \\ 0 & \text{inače} \end{cases} \tag{9}$$

Tada se jednadžbe 6, 7 i 8 mogu zapisati kao:

$$\boldsymbol{h}^{(t+1)} = \boldsymbol{X} \,\bar{\times}_2\, \boldsymbol{a}^{(t)} \,\bar{\times}_3\, \boldsymbol{t}^{(t)} \tag{10}$$

$$\boldsymbol{a}^{(t+1)} = \boldsymbol{X}^T \,\bar{\times}\, \boldsymbol{h}^{(t+1)} \,\bar{\times}_3\, \boldsymbol{t}^{(t)} \tag{11}$$

$$\boldsymbol{t}^{(t+1)} = \boldsymbol{X}^T \,\bar{\times}_1\, \boldsymbol{h}^{(t+1)} \,\bar{\times}_2\, \boldsymbol{a}^{(t)} \tag{12}$$

Ovdje notacija $\boldsymbol{X} \,\bar{\times}_i\, \boldsymbol{x}$ znači množenje tenzora $\boldsymbol{X}$ vektorom x u dimenziji i.

Tada se aproksimacija tenzora $\boldsymbol{X}$ PARAFAC dekompozicijom može dobiti kao:

Slika 3: PARAFAC dekompozicija

$$\boldsymbol{\mathcal{X}} \approx \boldsymbol{\lambda}[\![\boldsymbol{H}, \boldsymbol{A}, \boldsymbol{T}]\!] \equiv \sum_{\boldsymbol{r=1}}^{\boldsymbol{R}} \boldsymbol{\lambda}_r \boldsymbol{h}_r \circ \boldsymbol{a}_r \circ \boldsymbol{t}_r \tag{13}$$

Ovdje pretpostavljamo $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_R$. Matrice $\boldsymbol{H}, \boldsymbol{A}, \boldsymbol{T}$ imaju polja duljine 1, ali suprotno od SVD-a, nisu generalno ortonormalne. PARAFAC dekompozicija aproksimira tenzor $\boldsymbol{\mathcal{X}}$ sumom , kao što je prikazano na (3) .

Glavna trojka PARAFAC vektora $\boldsymbol{h_1}$, $\boldsymbol{a_1}$, $\boldsymbol{t_1}$ pruža hub i aurotitet rezultate za dominantnu temu u kolekciji web stranica (na isti način kao što je prethodno opisano kod HITS-a), te term rezultate. Drugim riječima, termi koji imaju najveće rezultate $\boldsymbol{t_1}$ su najdeskriptivniji termi.

Suprotno SVD-u, nema garancije da će PARAFAC aproksimacija biti optimalna, te vektori nisu ortogonalni.

2.2.1 TOPHITS ukratko

Ideja iza TOPHITSa je sljedeća: pretpostavimo da analiziramo kolekciju I web stranica imajući ukupno K termova u tekstu poveznica svih poveznica. Tada je $I \times I \times K$ tenzor susjedstva $\boldsymbol{\mathcal{X}}$ definiran kao u (6). Primijetimo da je tenzor $\boldsymbol{\mathcal{X}}$ rijedak zato što većina stranica pokazuje samo na na nekoliko drugih stranica u kolekciji, a svaki link koristi samo nekoliko termova. Stoga je realno za očekivati da je broj ne-nul elemenata u $\boldsymbol{\mathcal{X}}$ jednak $O(I)$.

Uzimajući u obzir $R > 0$ (broj različitih grupiranja podataka), TOPHITS algoritam pronađe matrice $\boldsymbol{H}$**i**$\boldsymbol{A}$, veličina $I \times R$, i matricu $\boldsymbol{T}$ veličine $K \times R$. Svaka trojka $\{\boldsymbol{h_r}, \boldsymbol{a_r}, \boldsymbol{t_r}\}$, za $r = 1 \ldots R$ definira grupiranje hubova, autoriteta i terma uzimajući u obzir ulaze s najvećom vrijednosti u svakom vektoru; vrijednost λ_r definira težinu grupiranja. (Bez smanjenja općenitosti, pretpostavljamo da su stupci naših matrica normalizirani.)

TOPHITS je zapravo ekstenzija Kleinbergovog HITS algortma koja je koristila singularni vektor matrice hiperlinkova (2D tenzor) da bi proizvela skup hubova i autoriteta. Dodatak topic-vektora znači da je određivanje skupa singularnih vektora koji sadrže odgovor na upit zapravo promatranje koji topic-vektori imaju najveći score uključujući njihove autoritete i hubove. Slično kao PageRank, TOPHITS je neovisan o upitima zato što se računanje značajnih vektora može obaviti unaprijed i offline.

3 Metodologija

3.1 PARAFAC model

3D dekompozicija simultano je predložena sa strane Harshmana, koristeći ime "Parallel Factors ili PARAFAC" te sa strane Carroll i Chang, koristeći ime "Canonical Decomposition ili CANDECOMP". Cilj je prikazati vektor reda N kao sumu vektora vanjskog produkta kao što je prikazano na slici 3.

Matematički, problem je zadan na sljedeći način. Pretpostavimo da imamo tenzor $\boldsymbol{\mathcal{X}}$ veličine $I_1 \times I_2 \times \ldots \times I_n$ i željeni rang aproksimacije R. Tada želimo naći matrice $\boldsymbol{U}^n$ veličine $I_n \times R$, za svaki $N = 1, \ldots, N$ i težinski vektor $\boldsymbol{\lambda}$ duljine R tako da vrijedi:

$$\boldsymbol{\mathcal{X}} \approx \boldsymbol{\lambda}[\![\boldsymbol{U^{(1)}}, \boldsymbol{U^{(2)}}, \ldots, \boldsymbol{U^{(N)}}]\!] \tag{14}$$

Kruskalov operator $[\![.]\!]$ je skraćeni zapis sume ranga svih vanjskih produkata stupaca. Drugim riječima,

$$\lambda[[\mathrm{U}^{(1)}, \mathrm{U}^{(2)}, \ldots, \mathrm{U}^{(N)}]] \equiv \sum_{r=1}^{R} \lambda_r \mathrm{u}_r^{(1)} \circ \mathrm{u}_r^{(2)} \circ \cdots \circ \mathrm{u}_r^{(N)} \tag{15}$$

Bez smanjenja općenitosti, prepostavljamo da je $\left\|\boldsymbol{u}_r^{(n)}\right\| = 1$ za sve $r = 1, \ldots, R$ i $n = 1, \ldots, N$. Povrh toga, preuredimo završno rješenje tako da vrijedi $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_R$.

Cilj nam je riješiti problem minimizacije:

$$\min \left\| \mathrm{X} - \lambda[[\mathrm{U}^{(1)}, \mathrm{U}^{(2)}, \ldots, \mathrm{U}^{(N)}]] \right\|^2 \tag{16}$$

s obzirom na $\lambda \in \mathbb{R}^R,\ \mathrm{U}^{(n)} \in \mathbb{R}^{I_n \times R},\ n = 1, \ldots, N$.

U slučaju TOPHITSA, $\boldsymbol{\lambda}$ je tenzor reda 3, tj $N = 3$ i vrijedi:

$$\boldsymbol{H} \equiv \boldsymbol{U^{(1)}}, \boldsymbol{A} \equiv \boldsymbol{U^{(2)}}, \boldsymbol{T} \equiv \boldsymbol{U^{(3)}} \tag{17}$$

3.2 Pohlepni PARAFAC

Notacija $\boldsymbol{X}_{(n)}$ reprezentira n-to unfolding tenzora $\boldsymbol{\mathcal{X}}$. Drugim riječima, $\boldsymbol{X}_{(n)}$ predstavlja reorganizirane ulaze $\boldsymbol{\mathcal{X}}$ u matricu veličine $I_n \times J$, gdje je $J = \prod_{k=1, k \neq n}^{N} I_k$ tako da su "vlakna" u dimenziji n uređena kao stupci matrice. Matematički imamo:

$$[\boldsymbol{X}_{(n)}]_{i,j} = x_{i_1,i_2,\ldots,i_N}$$

$$i = i_n, j = 1 + \sum_{k=1,k\neq n}^{N} (i_n - 1) \prod l = 1, l \neq n^{k-1} I_l$$

$$i \leq i \leq I_n, 1 \leq j \leq J \quad (18)$$

3.2.1 Pohlepni PARAFAC - pseudokod

Algorithm 1 Greedy PARAFAC

in: Tensor $\mathcal{X}$ of size $I_1 \times I_2 \times \cdots \times I_N$.
in: Desired rank $R > 0$.
for $r = 1, \ldots, R$ **do** {outer loop}
  Set $\mathbf{v}^{(n)}$ to be a vector of all ones of length I_n for $n = 1, \ldots, N$.
  repeat {middle loop}
    for $n = 1, \ldots, N$ **do** {inner loop}
      Set $\mathbf{w} = \mathbf{X}_{(n)}\mathbf{z}^{(n)} - \sum_{i=1}^{r-1} \left(\mathbf{u}_i^{(n)} \prod_{\substack{m=1 \\ m\neq n}}^{N} (\mathbf{v}^{(n)})^{\mathsf{T}} \mathbf{u}_i^{(m)} \right)$ where $\mathbf{z}^{(n)} \equiv \mathbf{v}^{(1)} \otimes \cdots \otimes \mathbf{v}^{(n-1)} \otimes \mathbf{v}^{(n+1)} \otimes \cdots \otimes \mathbf{v}^{(N)}$.
      Set $\lambda_r = \|\mathbf{w}\|$.
      Set $\mathbf{v}^{(n)} = \mathbf{w}/\lambda_r$.
    end for
  until the fit ceases to improve or the maximum number of middle-loop iterations has been exceeded.
  Set $\mathbf{u}_r^{(n)} = \mathbf{v}^{(n)}$ for $n = 1, \ldots, N$.
end for
out: $\boldsymbol{\lambda} \in \mathbb{R}^R$ and $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$.

1. Vanjska petlja (po r) računa $\{\mathrm{u}_r^{(1)}, \ldots, \mathrm{u}_r^{(N)}\}$
   Za svaki $r, r = 1, \ldots, N$, unutarnja petlja metodom alternirajućih najmanjih kvadrata minimizira

2. $$\left\| \left(\mathrm{X} - \sum_{i=1}^{r-1} \lambda_i \mathrm{u}_i^{(1)} \circ \cdots \circ \mathrm{u}_i^{(N)} \right) - \left(\mathrm{v}^{(1)} \circ \cdots \circ \mathrm{v}^{(N)} \right) \right\| \tag{19}$$

   po vektorima $\mathrm{v}^{(1)}, \ldots, \mathrm{v}^{(N)}$
   Tada je $\mathrm{u}_r^{(n)} = \mathrm{v}^{(n)},\ n = 1, \ldots, N$

3.3 Alternirajući najmanji kvadrati za PARAFAC

Češći pristup rješavanja PARAFAC dekompozicija je metoda alternirajućih najmanjih kvadrata. U svakoj unutarnjoj iteraciji, računamo cijelu ntu matricu $\boldsymbol{U^{(n)}}$ dok sve ostale matrice držimo fiksne. V koji se računa nakon svake unutarnje iteracije je rješenje danog optimizacijskog problema:

$$\min_V \left\| \mathrm{X} - [[\mathrm{U}^{(1)}, \ldots, \mathrm{U}^{(n-1)}, \mathrm{V}, \mathrm{U}^{(n+1)}, \ldots, \mathrm{U}^{(N)}]] \right\|^2 \tag{20}$$

što se matrično može zapisati kao

$$\min_V \left\| \mathrm{X}_n - \mathrm{V}\mathrm{Z}^{(n)T} \right\|^2 \tag{21}$$

Matrica $\boldsymbol{Z}^{(n)}$ je veličine $J \times R$ i definirana kao (2.6) Rješenje najmanjih kvadrata za 21 uključuje pseudo inverz matrice $\boldsymbol{Z}^{(n)}$:

$$\boldsymbol{V} = \boldsymbol{X}_{(n)} (Z^{(n)T})^{\dagger} \tag{22}$$

Prikladno, pseudo inverz $\boldsymbol{Z}^{(n)}$ ima specijalnu strukturu. Neka je $\boldsymbol{Y}^{(n)}$ simetrična $R \times R$ matrica kao u (2.7). Tada se može pokazati da je:

$$(Z^{(n)T})^{\dagger} = \boldsymbol{Z}^{(n)} \boldsymbol{Y}^{(n)T} \tag{23}$$

Dakle rješenje za 21 je dano (2.5), pa računanje $\boldsymbol{U}_n$ zapravo svodimo na računanje inverza specijalne $R \times R$ matrice $\boldsymbol{Y}^{(n)}$.

3.3.1 Alternirajući najmanji kvadrati za PARAFAC - pseudokod

Algorithm 2 Alternating Least Squares (ALS) for N-way arrays

in: Tensor $\mathcal{X}$ of size $I_1 \times I_2 \times \cdots \times I_N$.
in: Desired rank $R > 0$.
Initialize $\mathbf{U}^{(n)}$ for $n = 1, \ldots, N$ (see §2.4).
repeat {outer loop}
  for $n = 1, \ldots, N$ **do** {inner loop}

(2.5) Set $\mathbf{V} = \mathbf{X}_{(n)}\mathbf{Z}^{(n)}\mathbf{Y}^{(n)}$,

(2.6) where $\mathbf{Z}^{(n)} \equiv \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \otimes \ldots \otimes \mathbf{u}_r^{(n-1)} \otimes \mathbf{u}_r^{(n+1)} \otimes \ldots \otimes \mathbf{u}_r^{(N)}$,

(2.7) and $\mathbf{Y}^{(n)} \equiv \left(\mathbf{U}^{(1)\mathsf{T}}\mathbf{U}^{(1)} * \cdots * \mathbf{U}^{(n-1)\mathsf{T}}\mathbf{U}^{(n-1)} * \mathbf{U}^{(n+1)\mathsf{T}}\mathbf{U}^{(n+1)} * \cdots * \mathbf{U}^{(N)\mathsf{T}}\mathbf{U}^{(N)}\right)^{-1}$.

    for r=1,...,R **do** {Assign $\mathbf{U}^{(n)}$}
      Set $\lambda_r = \|\mathbf{v}_r\|$
      Set $\mathbf{u}_r^{(n)} = \mathbf{v}_r/\lambda_r$.
    end for
  end for
until the fit ceases to improve or the maximum number of outer iterations is exceeded.
out: $\boldsymbol{\lambda} \in \mathbb{R}^R$ and $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$.

3.3.2 Inicijalizacija

- Pohlepni PARAFAC - inicijaliziramo matrice $\mathrm{U}^{(n)}$ kao matrice dobivene pohlepnim PARAFAC algoritmom
- Nasumična inicijalizacija - matrice $\mathrm{U}^{(n)}$ inicijaliziramo nasumičnim vrijednostima
- HOSVD inicijalizacija - za svaki mod $\mathrm{X}_{(n)}$ izračunamo SVD dekompoziciju, $\mathrm{X}_{(n)} = \mathrm{U}_n\Sigma_n\mathrm{V}_n^T$, i inicijaliziramo matrice $\mathrm{U}^{(n)} = \mathrm{U}_n,\ n = 1, \ldots, N$

3.4 Specijalna konsideracija za razbacane podatke

Kao što smo već spomenuli, tenzor $\boldsymbol{\mathcal{X}}$ je veoma rijedak. Posljedično, njegova reprezentacija $\boldsymbol{X}_n$ je rijetka matrica (ima iste ne nul elemente, ali preoblikovane). Matrica $\boldsymbol{Z}^n$ iz (2.6) se ne bi trebala formirati eksplicitno jer bi onda bila gusta matrice veličine $I_n \times J$ gdje je $J = \prod_{k=1,k\neq n}^{N} I_N$. Umjesto toga, računanje

$$\boldsymbol{X}_{(n)}\boldsymbol{Z}^{(n)} \tag{24}$$

potrebno za (2.5) mora biti izračunato posebno, iskorištavanjem svojstava Kroneckerove strukture produkta u $\boldsymbol{Z}^{(n)}$ da bi se očuvala rijetkost. Završni rezultat je veličine $I_n \times R$ stoga sse može pohraniti u gustu matricu. Jedna od metoda prikazana je sljedećim algoritmom:

Algorithm 3 Computing the sparse product $\mathbf{X}_{(n)}\mathbf{Z}^{(n)}$

in: Tensor $\mathcal{X}$ of size $I_1 \times I_2 \times \cdots \times I_N$ with Q nonzeros. Let the index of the qth nonzero be $(k_{1_q}, k_{2_q}, \ldots, k_{N_q})$ and its value be given by v_q.
in: Index n and matrices $\mathbf{U}^{(m)}$ for $1 \leq m \leq N, m \neq n$.
for $r = 1 \ldots, R$ **do**
  for $q = 1, \ldots, Q$ **do**
    Compute $w_q = v_q \prod_{\substack{m=1 \\ m \neq n}}^{N} u^{(m)}_{k_{m_q}, r}$
  end for
  for $i = 1, \ldots, I_n$ **do** {Compute rth column of $\mathbf{P}$}
    Set $p_{ir} = \sum_{\substack{q=1 \\ k_{n_q} = i}}^{Q} w_q$.
  end for
end for
out: $\mathbf{P} = \mathbf{X}_{(n)}\mathbf{Z}^{(n)}$

4 TOPHITS i upiti

Jednom kada smo izračunali TOPHITS model ranga R

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\lambda}[\![\boldsymbol{H}, \boldsymbol{A}, \boldsymbol{T}]\!] \tag{25}$$

možemo ga iskoristiti za razumijevanje podataka na razne načine. Gledajući najveću vrijednost svake trojke $\{\boldsymbol{h_r}, \boldsymbol{a_r}, \boldsymbol{t_r}\}$ pruža grupiranje hubova, autoriteta, termova web stranice, a λ_r pruža relativnu težinu grupiranja.

Jedno pitanje koje smatramo osnovnim pitanjem web pretraživanja: pronaći sve stranice povezane s određenim termom ili skupom termova. Promotrimo vektor upita g duljine K (gdje je K broj termova) kao:

$$q_k = \begin{cases} 1 & \text{ako je term k u upitu} \\ 0 & \text{inače} \end{cases} \tag{26}$$

Primijetimo da nema razloga restringirati se na upite o termovima. Također se možemo baviti problem pronalaženja web stranica i/ili termova povezanih s određenom stranicom ili skupom stranica.

4.1 Pronalaženje podudarne grupe

Umjesto vraćanja liste rangiranih stranica, TOPHITS omogućuje opciju identificiranja grupiranja relevantnih za dani upit. Možemo kreirati vektor grupe $\boldsymbol{s}$ duljine R koji sadrži vrijednosti svakog grupiranja, na temelju matrice $\boldsymbol{T}$ iz PARAFAC modela:

$$\boldsymbol{s} = \boldsymbol{\Lambda T^T q}, \boldsymbol{\Lambda} = diag(\boldsymbol{\lambda}) \tag{27}$$

Ulaz s_r daje rezultat-vrijednost r-te grupe, a grupiranja s većim rezultatima smatraju se više relevantnima.

Alternativno, možemo konstruirati vektora upita na temelju web stranica, $\hat{\boldsymbol{q}} \in R^l$ te izračunati rezultate grupe kao:

$$\hat{\boldsymbol{s}} - \boldsymbol{\Lambda A^T \hat{q}}, \boldsymbol{\Lambda} = diag(\boldsymbol{\lambda}) \tag{28}$$

4.2 Pronalaženje skupa autoriteta

Također je moguće vratit tradicionalnu rangiranu listu vjerojatnosti. Možemo kombinirati sve informacije TOPHITS modela da bi vratili skup rangiranih autoriteta i/ili hubova. Definirajmo $\boldsymbol{s}$ kao u 27. Tada su kombinirani autoriteti dani s:

$$a^* = As = \sum_{r=1}^{R} s_r a_r \tag{29}$$

Sortiranje ulaza $\boldsymbol{a}^*$ vraća rangiranu listu autoriteta. Na isti način, kombinirani hubovi dani su:

$$h^* = Hs = \sum_{r=1}^{R} s_r h_r \tag{30}$$

5 Implementacija

Testirali smo našu tehniku na skupu podataka, generiranima koristeći web crawler koji uključuje i tekst poveznice.