Hash-alapú keresések az információ-visszakeresésben

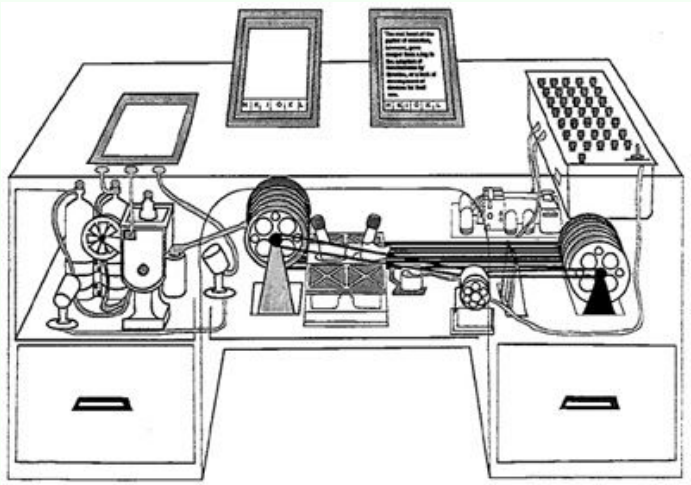
Bodó Zalán

Babeș–Bolyai Tudományegyetem
Magyar Matematika és Informatika Intézet

A Magyar Tudomány Napja Erdélyben
Kolozsvár, 2012

Tartalom

Információ-visszakeresés



- **információ-visszakeresés** (*information retrieval*) = információ visszatérítése egy adathalmazból, amely releváns az adott információs igény/kérés szempontjából
- alapötlet: Vannevar Bush: *As We May Think* (1945)

> *"It consists of a desk, and while it can presumably be operated from a distance, it is primarily the piece of furniture at which he works. On the top are slanting translucent screens, on which material can be projected for convenient reading. There is a keyboard, and sets of buttons and levers. Otherwise it looks like an ordinary desk."*

`http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/`

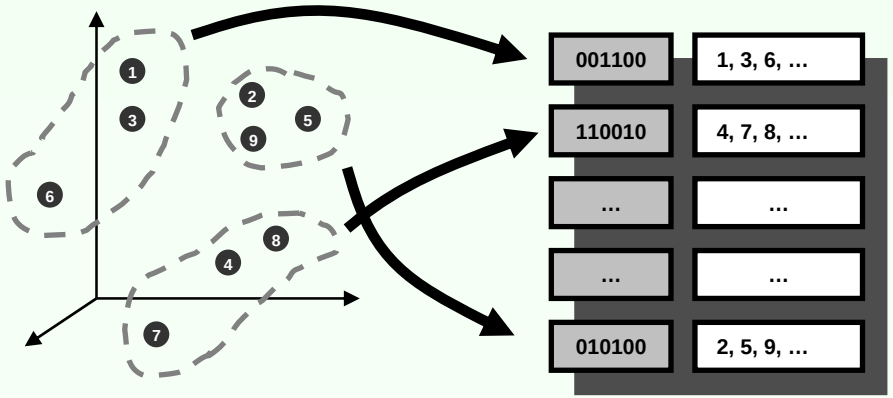
- az adatok nem strukturáltak; nem adatbázis-lekérdezést hajtunk végre
- általában nem *egyetlen* találatot térítünk vissza, hanem **az első k legközelebbit**, vagy akár az **összes illeszkedőt**
- a találatokat **rangsoroljuk**
- az adatok legtöbbször szövegek és képek
- **probléma**: be kell járni az **egész** adatbázist és össze kell hasonlítani a kéréssel

Példa: *Vektortér modell*

A szövegeket tárolhatjuk *bag-of-words*-ként, azaz szavak bag-jeként (multihalmazként) vagy ekvivalens módon vektorokként. A tér dimenzióit a (valamilyen speciális módon kiválasztott) indexelő szavak adják.
Két dokumentum hasonlóságát kiszámíthatjuk a vektorok közötti szög koszinuszaként:

$$s(\boldsymbol{d}_i, \boldsymbol{d}_j) = \frac{\boldsymbol{d}_i' \boldsymbol{d}_j}{\|\boldsymbol{d}_i\| \|\boldsymbol{d}_j\|}$$

Hash-alapú keresések



- alapötlet: a pontokat tegyük egy **hash táblába** úgy, hogy *közeli* pontok *azonos* vagy egymáshoz *közeli kulcsú* rekeszbe kerüljenek
- kulcs = bináris szekvencia
- a Hamming-távolságot "egyszerű" kiszámítani
- legyen $\boldsymbol{u}, \boldsymbol{v} \in \{0,1\}^n$; ekkor

$$d_{\mathrm{H}}(\boldsymbol{u}, \boldsymbol{v}) = \sum_{i=1}^{n} (\boldsymbol{u} \,\mathrm{XOR}\, \boldsymbol{v})$$

Locality-Sensitive Hashing (LSH)

ALG 1 LSH

1: Generáljunk r darab egységnyi hosszúságú (egyenletes eloszlású) d-dimenziós vektort: $\{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_r\}$

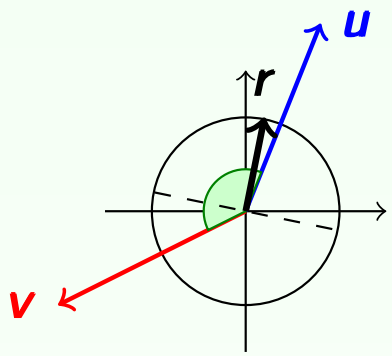
2: Az r darab hash függvényünk legyen

$$h_i(\boldsymbol{u}) = \begin{cases} 1, & \boldsymbol{r}_i'\boldsymbol{u} \geq 0 \\ 0, & \boldsymbol{r}_i'\boldsymbol{u} < 0 \end{cases}, \quad i = 1, 2, \ldots, r$$

3: Egy $\boldsymbol{x}$ ponthoz rendelt bináris hash szekvencia

$$(h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots, h_r(\boldsymbol{x}))$$

Miért működik az LSH?



- $\boldsymbol{u}, \boldsymbol{v}$ – vektorok $\mathbb{R}^d$-ben
- $\boldsymbol{r}$ – véletlen vektor $\mathbb{R}^d$-ben, $\|\boldsymbol{r}\| = 1$

Ha a hash függvény

$$h_{\boldsymbol{r}}(\boldsymbol{u}) = \begin{cases} 1, & \boldsymbol{r}'\boldsymbol{u} \geq 0 \\ 0, & \boldsymbol{r}'\boldsymbol{u} < 0 \end{cases}$$

akkor

$$P(h_{\boldsymbol{r}}(u) = h_{\boldsymbol{r}}(v)) = 1 - \frac{\theta(\boldsymbol{u}, \boldsymbol{v})}{\pi}$$

Biz.

$$\begin{aligned} & P(h_{\boldsymbol{r}}(\boldsymbol{u}) \neq h_{\boldsymbol{r}}(\boldsymbol{v})) \\ & = P(\boldsymbol{r}'\boldsymbol{u} \geq 0, \boldsymbol{r}'\boldsymbol{v} < 0) \\ & + P(\boldsymbol{r}'\boldsymbol{u} < 0, \boldsymbol{r}'bv \geq 0) \\ & = \frac{\theta(\boldsymbol{u}, \boldsymbol{v})}{2\pi} + \frac{\theta(\boldsymbol{u}, \boldsymbol{v})}{2\pi} \\ & = \frac{\theta(\boldsymbol{u}, \boldsymbol{v})}{\pi} \end{aligned}$$

□

Spektrális hashing

- $\boldsymbol{y}_i \in \{-1, 1\}^r$, $i = 1, \ldots, N$ kódszavakat keresünk a pontokhoz úgy, hogy egymáshoz hasonló pontok esetén ezen kódszavak távolsága kicsi legyen:

$$\min_{\{\boldsymbol{y}_i\}_{i=1,\ldots,N}} \quad \sum_{i,j=1}^{N} W_{ij} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2$$

$$\text{ú.h.} \quad \boldsymbol{y}_i \in \{-1, 1\}^r, \quad \sum_{i=1}^{N} \boldsymbol{y}_i = \boldsymbol{0}, \quad \frac{1}{N} \sum_{i=1}^{N} y_i y_i' = \boldsymbol{I}$$

  - 2. feltétel: a bitek "kiegyensúlyozottak" legyenek
  - 3. feltétel: a bitek ne korreláljanak egymással

- a feladatot relaxáljuk:

$$\min_{\{\boldsymbol{y}_i\}_{i=1,\ldots,N}} \quad tr(\boldsymbol{Y}'\boldsymbol{L}\boldsymbol{Y})$$

$$\textbf{ú.h.} \quad \boldsymbol{Y}'\boldsymbol{1} = \boldsymbol{0}, \quad \boldsymbol{Y}'\boldsymbol{Y} = \boldsymbol{I}$$

- a feladat **megoldása**: $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ első r darab (legkisebb sajátértékű) sajátvektora, $\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{v}_2 & \boldsymbol{v}_3 & \dots & \boldsymbol{v}_{r+1} \end{pmatrix}$
- az i-edik kódszó ekkor : $\boldsymbol{y}'_i = \begin{pmatrix} v_{2i} & v_{3i} & \dots & v_{(r+1)i} \end{pmatrix}$
- az eredeti cikkben az általánosítást úgy oldották meg, hogy feltételezték a Gauss-kernel használatát, a többdimenziós egyenletes eloszlást, és felhasználták a Laplace–Beltrami operátorok sajátfüggvényeire vonatkozó eredményeket (approximáció)

Spektrális klaszterezés

- **normalizált minimális vágás** gráfokban:

$$\frac{\sum_{i\in A, j\in \bar{A}} W_{ij}}{\sum_{i\in A, j\in V} W_{ij}} + \frac{\sum_{i\in A, j\in \bar{A}} W_{ij}}{\sum_{i\in \bar{A}, j\in V} W_{ij}}$$

- átalakítható az alábbi numerikus optimalizálási feladatra (relaxáció):

$$\begin{aligned} \max_{\boldsymbol{z}} \quad & \frac{\boldsymbol{z}' \boldsymbol{D}^{-1/2} \boldsymbol{W} \boldsymbol{D}^{-1/2} \boldsymbol{z}}{\boldsymbol{z}'\boldsymbol{z}} \\ \text{ú.h.} \quad & \boldsymbol{z}' \boldsymbol{D}^{1/2} \mathbf{1} = 0 \end{aligned}$$

- kimutatható, hogy a spektrális klaszterezés egy **hipersíkkal** "vágja" el az adatokat ($\boldsymbol{W} = \boldsymbol{\Phi}'\boldsymbol{\Phi}$)

$$\begin{aligned} \max_{\boldsymbol{w}} \quad & \|\boldsymbol{w}' \boldsymbol{\Phi} \boldsymbol{D}^{-1/2}\| \\ \text{ú.h.} \quad & \sum_{i=1}^{N} \boldsymbol{w}' \phi(\boldsymbol{x}_i) = 0, \quad \|\boldsymbol{w}\| = 1 \end{aligned}$$

- ez felírható a következőképpen:

$$\max_{\boldsymbol{w}} \quad \frac{\boldsymbol{w}'\boldsymbol{\Phi}\boldsymbol{D}^{-1}\boldsymbol{\Phi}'\boldsymbol{w}}{\boldsymbol{w}'\boldsymbol{w}}$$

$$\text{ú.h.} \quad \sum_{i=1}^{N} \boldsymbol{w}'\phi(\boldsymbol{x}_i) = 0$$

- ennek megoldása a $\boldsymbol{\Phi}\boldsymbol{D}^{-1}\boldsymbol{\Phi}'$ második legnagyobb sajátértékű sajátvektora ($\boldsymbol{u}_2$)
- $\boldsymbol{D}^{-1/2}\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{D}^{-1/2}$ 2. legnagyobb sajátértékű sajátvektora $\boldsymbol{v}_2$; a kapcsolat a kettő között: $\boldsymbol{u}_2 = \boldsymbol{\Phi}\boldsymbol{D}^{-1/2}\boldsymbol{v}_2 e_2^{-1}$
- ezáltal egy induktív klaszterezőt nyerünk $\boldsymbol{w}$ normálissal:

$$\boldsymbol{w} = \boldsymbol{u}_2 = \boldsymbol{\Phi}\boldsymbol{D}^{-1/2}\boldsymbol{v}_2 e_2^{-1}$$

- ekkor egy új pont klasztercímkéjét a következő módon határozzuk meg:

$$f_2(\boldsymbol{x}) = \phi(\boldsymbol{x})'\boldsymbol{w} = \phi(\boldsymbol{x})'\boldsymbol{u}_2 = \phi(\boldsymbol{x})'\boldsymbol{\Phi}\boldsymbol{D}^{-1/2}\boldsymbol{v}_2 e_2^{-1}$$

- megmutatható, hogy

$$\begin{aligned} \operatorname{sgn}\left(\boldsymbol{\Phi}'\boldsymbol{w}\right) &= \operatorname{sgn}\left(\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{D}^{-1/2}\boldsymbol{v}_2 e_2^{-1}\right) = \operatorname{sgn}\left(\boldsymbol{D}^{-1/2}\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{D}^{-1/2}\boldsymbol{v}_2 e_2^{-1}\right) \\ &= \operatorname{sgn}\left(\boldsymbol{v}_2 e_2 e_2^{-1}\right) = \operatorname{sgn}(\boldsymbol{v}_2) \end{aligned}$$

Lineáris spektrális hashing

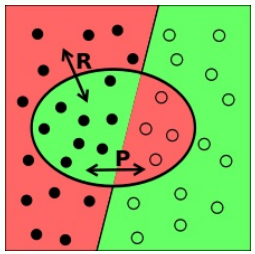
- a gond az **általánosítás** új pontokra
- **alapötlet**: használjuk fel az hipersíkkal való szétválasztást a spektrális klaszterezéstől
- cseréljük ki a Laplace-mátrixot a normalizált Laplace-mátrixra a spektrális hashingben: $\boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}$
- felhasználva az általánosítási eredményeket a spektrális klaszterezéstől kapjuk, hogy

$$f(\boldsymbol{x}) = \begin{pmatrix} f_2(\boldsymbol{x}) & f_3(\boldsymbol{x}) & \dots & f_{r+1}(\boldsymbol{x}) \end{pmatrix}'$$

- **probléma**: $\phi(\boldsymbol{x})'\boldsymbol{\Phi}$ kiszámításához N kernel számolásra van szükség
- **megoldás**: használjuk a pontokat az input térben $\Rightarrow$ kernel $=$ skalárszorzat
- ekkor kiszámítjuk az első r sajátvektort, $\boldsymbol{g}_k := \boldsymbol{u}_k$ vagy $\boldsymbol{g}_k := \boldsymbol{X}\boldsymbol{D}^{-1/2}\boldsymbol{v}_k$, $k = 2, \dots, r+1$
- új pont esetén a ponthoz rendelt kódszót a $\boldsymbol{x}'\boldsymbol{g}_k$, $k = 2, \dots, r+1$ skalárszorzatokkal számítjuk ki

Kísérletek, eredmények

- tesztelés a Reuters-21578 [1] és 20Newsgroups [2] szövegkorpuszokon
- szövegek feldolgozása:
  - stopszavak kiszűrése (199 db.)[3]
  - az első 5000 leggyakoribb szó használata dimenziókként
  - **bag-of-words** reprezentáció + **tf-idf** súlyozás
  - dokumentumvektorok normalizálása
- nem használtuk a címkéket
- a teszthalmaz minden dokumentumához megkerestük az első 50 legközelebbit a tanuló adatokból
- ezek megtalálásának sikerességét mértük **precision** és **recall** segítségével (pl. 32 bit esetén $\{0, 1, 2, 4, \ldots, 32\}$ Hamming-távolságokra számítottunk recallt és precisiont)



[1] `http://disi.unitn.it/moschitti/corpora.htm`
[2] `http://qwone.com/~jason/20Newsgroups/20news-bydate.tar.gz`
[3] WordNet::Similarity 2.05, `http://wn-similarity.sourceforge.net/`

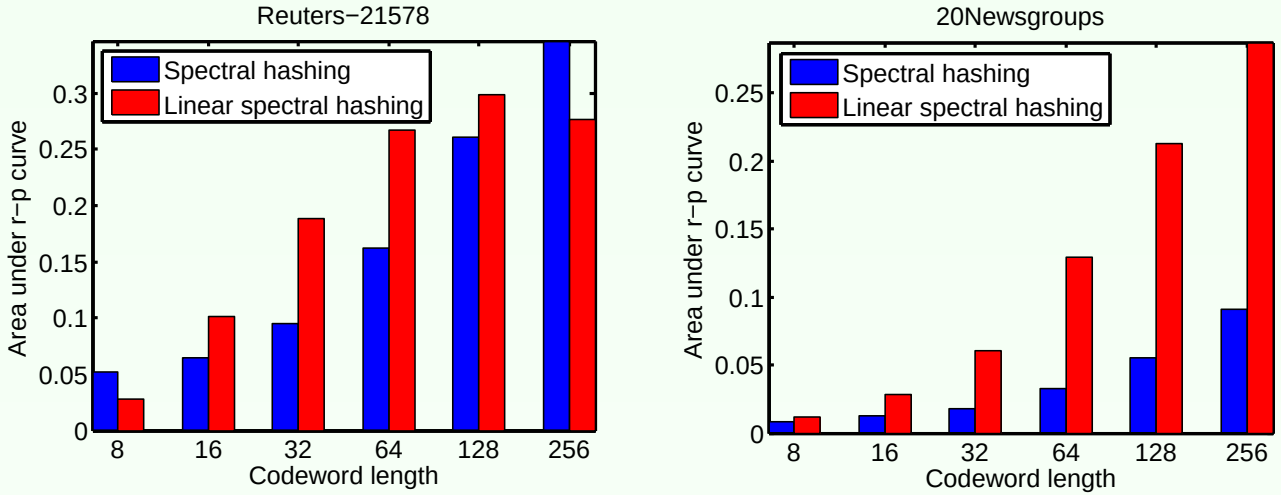


Figure: A recall-precision görbe alatti terület a hash szekvencia hosszának függvényében.

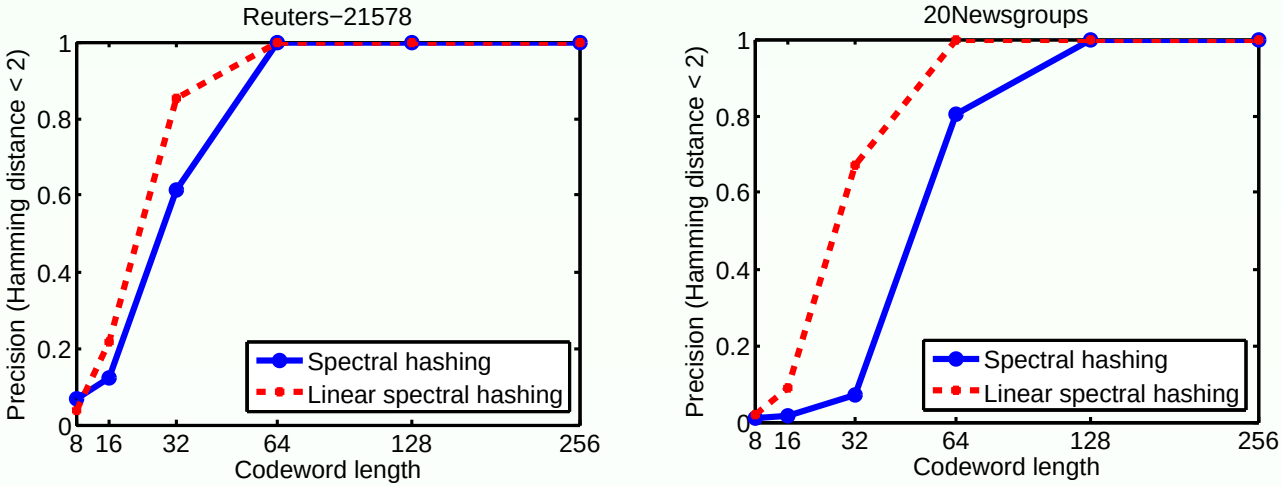


Figure: Precison eredmények Hamming-távolság < 2 esetén.

Bibliográfia

[Weiss, Torralba, Fergus 2008] Yair Weiss, Antonio B. Torralba, and Robert Fergus. Spectral hashing. In NIPS, pages 1753–1760. MIT Press, 2008.

[Rahimi, Recht 2004] Ali Rahimi and Ben Recht. Clustering with normalized cuts is clustering with a hyperplane. In Statistical Learning in Computer Vision, 2004.

Köszönöm a figyelmet!