Compact data structures: Broder's set sketches

Luca Becchetti

"Sapienza" Università di Roma – Rome, Italy

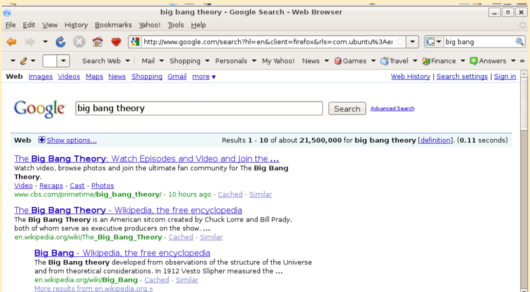
November 11, 2013

Playing with sets

How “similar”?

User 1: {Murray Gell-Mann, Sheldon Cooper, Leonard Susskind, Rajesh Kuthrapalli}
User 2: {Sheldon Cooper, Murray Gell-Mann, Howard Wolowitz, Rajesh Kuthrapalli, Leonard Hofstadter}

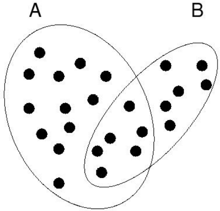
Why bother? Find similar users, filter out “similar” results etc.



Jaccard similarity coefficient

Given two *discrete* sets A and B:

- $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$ (Jaccard coefficient)
- $c(A,B) = \frac{|A \cap B|}{|B|}$ (containment of A in B)



Estimation

$J(A,B)$ (also known as *resemblance*) measures the extent to which A and B overlap
$c(A,B)$ measures extent to which A is a subset of B

Estimating similarity coefficients

Expensive to do exactly...

- Linear if we use a hash table (needs the hash table)
- Can be overly expensive in many cases
  - E.g.: detecting almost duplicates in the Web

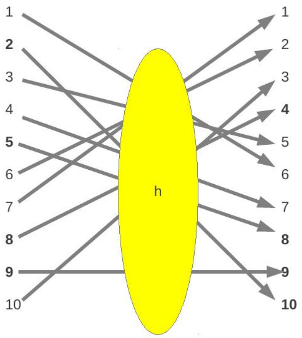
How to proceed...

Use compact set representations

References: [Broder et al., 1997, Broder, 2000] for (Web) documents, [Broder, 2000], [Broder et al., 2000],

Estimating similarity coefficients - Idea

- Assume we have a family $\mathcal{H}$ of hash functions that permute $[n] = \{0, \ldots n-1\}$
- Example ($n = 10$): for a particular $h \in \mathcal{H}$ and the set (2 5 8 9) we might have:



... hence applying $h(\cdot)$ we obtain the set (4 8 9 10) in this case

Estimating similarity coefficients - cont.

Assume we extract h uniformly at random from $\mathcal{H}$...

What does it mean to extract u. a. r?

Depends on how you define $\mathcal{H}$
Example: if $\mathcal{H} = \{(ax + b \mod p) \mod n\}$, with $a, b \in \{0, \ldots, p-1\}$ for p prime, $p > n$ this means choosing a, b u.a.r. in $\{0, \ldots, p-1\}$

- For every h and $X \subseteq [n]$, let $h(X)$ denote the image of X under h and $\min\{h(X)\} = \min_{x \in X} h(x)$
- $\mathcal{H}$ is *min-wise independent* if, once h is chosen u.a.r. from $\mathcal{H}$ we have:

$$\mathbf{P}[\min\{h(X)\} = h(z)] = \frac{1}{|X|}, \forall z \in X$$

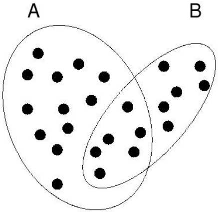
Estimating similarity coefficients - cont.

Assume $\mathcal{H}$ is min-wise independent
Consider two sets $A, B \subseteq [n]$

Theorem ([Broder et al., 2000])

Assume h is chosen u.a.r. from $\mathcal{H}$. Then:

$$\mathbf{P}[\min\{h(A)\} = \min\{h(B)\}] = J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

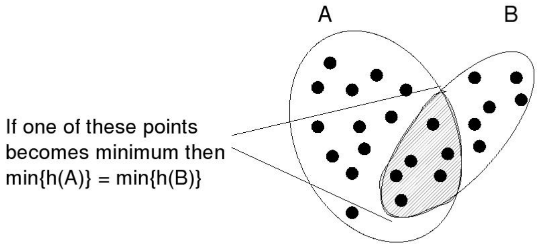


In the picture: $|A \cap B| = 6$, $|A \cup B| = 24$, $J(A, B) = 0.25$

Estimating similarity coefficients - cont.

Proof idea

- $\min\{h(A)\} = \min\{h(B)\}$ if and only if $\min\{h(A \cup B)\} = h(z)$, with $z \in A \cap B$
- All items in $A \cup B$ have equal chances of being the minimum
- $\mathbf{P}[\arg\min\{h(A \cup B)\} \in A \cap B] = \frac{|A \cap B|}{|A \cup B|}$



How to proceed in practice?

Given:

A collection of sets. E.g.: each set is a handy's address book containing a set of names, such as: ("S. Cooper", "L. Hofstadter", "R. Kuthrapalli", "H. Wolowitz")

Step 0: Pick m hash functions $h_1 \ldots, h_m$ u.a.r. from a min-wise independent family $\mathcal{H}$, with $h_i : [n] \rightarrow [n]$
For each set X:

1. (If necessary) map X's items to integers in $[n]$ (use same encoding for all sets)
2. Compute $M_i(X) = \min\{h_i(X)\}$, $i = 1, \ldots, m$

$(M_1(X), \ldots, M_m(X))$ is X's *fingerprint*. To estimate $J(A, B)$:

$$J(A, B) \simeq \frac{\sum_{i=1}^{m}(M_i(A) == M_i(B))}{m}$$

Q.: can you figure out why?

Applying to user-user recommendations/1

Users and their profiles:

Each user is represented by an incidence vector $\vec{u} \in \{0, 1\}^n : \vec{u}_x = 1$ if u purchased (browsed, considered ...) item x, 0 otherwise

Representing a user's profile in compact form

User u, i.e., $\vec{u}$, is represented by a fingerprint $F(u) = (M_1^u(X_u), \ldots, M_m^m(X_u))$, where X_u is the set of items purchased (browsed, considered ...) by u

Computing fingerprints

For $i = 1, \ldots, m$, $M_i^u(X_u)$ is computed by hashing the set X_u using the i-th hash function, as shown before

Applying to user-user recommendations/2

User similarity

The similarity between two users u and v is given by their Jaccard coefficient:

$$J(u,v) = \frac{\vec{u} \cdot \vec{v}}{\sum_{x=1}^{n}(\vec{u}_x \text{ OR } \vec{v}_x)}$$

Estimating user similarity

$$J(u,v) \simeq \frac{\sum_{i=1}^{m}(M_i^u(X_u) == M_i^u(X_v))}{m}$$

Neighbourhood

Neighbourhood can be of fixed size or threshold based, as before

Applying to user-user recommendations/3

Estimating z-scores

$$\hat{z}_{Joe}(x) = \sum_{v \in Pool} \frac{J(Joe, v)}{J_{Joe}} z_v(x),$$

where $J_{Joe} = \sum_{v \in Pool} J(Joe, v)$ and the $z_v(x)$ as usual is computed from v's past ratings

Making a prediction

$$\hat{r}_{Joe}(x) = \mu_{Joe} + \sigma_{Joe} \hat{z}_{Joe}(x)$$

as before

Note

Ratings are not used to compute user pairwise similarities but they are used to compute z-scores

What else?

Perfect, min-wise independent hash functions can be very expensive

- $\Omega(n \log n)$ truly random bits necessary [Broder et al., 2000]
- In practice: use pairwise independent hash functions [Carter and Wegman, 1979] or approximate min-wise independent families of small (roughly logarithmic) size [Indyk, 1999]

Pairwise independent hash functions

- $\mathcal{H} = \{(ax + b \mod p) \mod n\}$, for p prime, $p > n$
- with a chosen u.a.r. $\{1, \ldots, p-1\}$, b chosen u.a.r. $\{0, \ldots, p-1\}$
- In practice, p might be the Mersenne prime $2^{31} - 1$ in many cases

Broder, A. Z. (2000).
Identifying and filtering near-duplicate documents.
In *11th Annual Symposium on Combinatorial Pattern Matching (CPM)*, pages 1–10.

Broder, A. Z., Charikar, M., Frieze, A. M., and Mitzenmacher, M. (2000).
Min-wise independent permutations.
J. Comput. Syst. Sci., 60(3):630–659.

Broder, A. Z., Glassman, S. C., Manasse, M. S., and Zweig, G. (1997).
Syntactic clustering of the web.
In *Selected papers from the sixth international conference on World Wide Web*, pages 1157–1166, Essex, UK. Elsevier Science Publishers Ltd.

Carter, J. L. and Wegman, M. N. (1979).
Universal classes of hash functions.
Journal of Computer and System Sciences, 18(2):143–154.

Indyk, P. (1999).

A small approximately min-wise independent family of hash functions.

In *SODA '99: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 454–456, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.