Today’s topics

• FAQs
• Minhash
• Minhash signature
• Calculating Minhash with MapReduce
• Locality Sensitive Hashing

FAQs

• Compute the similarity between the two strings s1=Farmers Insurance, s2 = Liberty Insurance

\[ \text{CosSimilarity}(V,W) = \cos(\alpha) = \frac{V \cdot W}{\|V\| \times \|W\|} = \frac{0.55 \times 0.55}{\sqrt{0.55^2 + 0.72^2}} \approx 0.38 \]

• What is the Jaccard similarity for the same case?

How can we compute TF/IDF based similarity analysis?

• Programming Assignment 2

Using n-Grams
Using n-GRAMS

- A string is divided into smaller tokens of size $n$.
  - q-grams or n-grams
  - Size of n-gram is string with a length $n$
    - Size of n-gram words is string with a length $n$ words
- Generating n-grams
  - Slide a window of size $n$ over the string

Generating n-grams

- Generate 3-grams
  - Q-grams of $s_1 = \{##H, #He, Hen, enr, ri, i, W, Wa, Wat, ate, ter, ern, rno, noo, oos, ose, self, edef\}$
  - Q-grams of $s_2 = \{##H, #He, Hen, enr, rny, ry, y, W, Wa, Wat, ate, ter, ern, rno, nos, ose, se#, e##\}$

n-gram based token similarity

- 13 overlaps among total 22 distinct q-grams.
  - Overlaps: number of two item pairs having overlap
- Jaccard similarity
  \[ \text{StringJaccard}(s_1, s_2) = \frac{13}{22} = 0.59 \]

n-gram based token similarity

- Using the cosine similarity with tf-idf weights
  \[ \text{CosSimilarity}(V, W) = ? \]

Using n-grams

- Using the same similarity measures used in other token based similarity computations
  - Less sensitive to typographical errors
    - “discretization” Vs. “discretisation”
  - What if we change the size of $n$?

How large should $n$ be? [1/2]

- $n$ should be picked large enough that the probability of any given n-gram appearing in any given document is low

  - Example
    - Our corpus of documents is emails
    - Picking $n = 5$
    - Printable ASCII characters $27^5 = 14,348,907$ possible 5-grams
    - Is it really true with real emails?
How large should $n$ be? [2/2]

- All characters do not appear with equal probability
- Imagine that there are only 20 characters and estimate the number of $n$-grams as $20^n$
- For research articles, a choice of $n = 9$ is considered safe

Applications of Near-neighbor search

Near Neighbor(NN) Search

- Also known as proximity search, similarity search
- Finding closest or most similar points
- The post-office problem
  - Assigning to a residence the nearest post office

Hashing $n$-grams

- Creating buckets and use the bucket numbers as the shingles
- For the set of 9-grams,
  - Map each of those 9-grams to a bucket number in the range $0$ to $2^{32} - 1$
  - 9 bytes of data is compacted to 4 bytes
- Can hash-based approaches provide NN search results?

Similarity-preserving summaries of set

- Signatures
  - Replacing large sets of $n$-grams by much smaller representations
- We should be able to compare the signatures of two sets and estimate the Jaccard similarity
  - Of the underlying sets from the signatures alone

Applications of Near-neighbor search: MinHashing
### Matrix representations of sets

<table>
<thead>
<tr>
<th>Element</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>e</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

* (a,b,c,d,e)  
  S1 = {a,d}  
  S2 = {c}  
  S3 = {b,d,e}  
  S4 = {a,c,d}  

### MinHashing

- **Signature generating algorithm**
  - Minhash of the characteristic matrix
  - Minhash$(\pi)$ of a set is the number of the row (element) with first non-zero in the permuted order $\pi$
    - $\pi = (b,e,a,d,c)$

#### Permuted order 1

<table>
<thead>
<tr>
<th>Element</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>e</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>a</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Permuted order 2

<table>
<thead>
<tr>
<th>Element</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>e</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>a</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

### MinHash and Jaccard Similarity

- **Theorem:**

  \[ P(\text{minhash}(S) = \text{minhash}(T)) = \text{JaccardSIM}(S,T) \]

#### Proof:

- Let $X$ be the number of rows with 1 for both $S$ and $T$ (e.g., $S = \{a,d\}$, $T = \{b,d,e\}$)
- Let $Y$ be the number of rows with either $S$ or $T$ have 1, but not both (e.g., $S = \{a,d\}$, $T = \{b,d,e\}$)
- Let $Z$ be the number of rows with 0 (e.g., $S = \{a,d\}$, $T = \{b,d,e\}$)

\[ P(\text{minhash}(S) = \text{minhash}(T)) = \frac{X}{X+Y} = \text{JaccardSIM}(S,T) \]

### MinHash and permuting the order

- What if we change the order of Elements?
- Can we simulate the effect of a random permutation?
- It is NOT feasible to permute a large characteristic matrix explicitly
  - N element will need N! permutations!
Representing the MinHash Signatures

- Form a signature matrix
  - The $i^{th}$ column of $M$ is replaced by the min hash signature of the $i^{th}$ column

- Compressed form for a sparse matrix

<table>
<thead>
<tr>
<th>Element</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>a</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>d</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

> 10^9 elements?

Using a random hash function

- Pick $n$ randomly chosen hash functions $h_1, h_2, \ldots, h_n$ on the rows
- In the signature matrix, let $\text{SIG}(i, c)$ be the element of the signature matrix for the $i^{th}$ hash function and column $c$

- Initially, the set $\text{SIG}(i, c)$ to $\infty$ for all $i$ and $c$

- For each row $r$
  - Compute $h_1(r), h_2(r), \ldots, h_n(r)$
  - For each column $c$
    - If $r$ has 0 in row, do nothing
    - If $r$ has 1 in row, then for each $i = 1, 2, \ldots, n$, set $\text{SIG}(i, c)$ to the smaller of the current value of $\text{SIG}(i, c)$ and $h_i(r)$

Example

<table>
<thead>
<tr>
<th>Row (element)</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>Hash 1</th>
<th>Hash M</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Example

<table>
<thead>
<tr>
<th>Row (element)</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>Hash 1</th>
<th>Hash M</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

$(\text{Row} \# + 1) \mod 5$

$(3 \times \text{Row} \# + 1) \mod 5$
Example continued

- \( h_1(x) = (x + 1) \mod 5 \)
- \( h_2(x) = (3x + 1) \mod 5 \)
- In the previous slide, \( h_1(0) \) and \( h_2(0) \) are both 1
- The row numbered 0 has 1's in S1 and S4

Example continued

- Row number 1
  - Only in S3 is 1
  - Hash value
    - \( h_1(1) = 2 \) and \( h_2(1) = 4 \)

Example continued

- Row number 2
  - S2 and S4
  - Hash value
    - \( h_1(2) = 3 \) and \( h_2(2) = 2 \)

Example continued

- Row number 3
  - S1, S3 and S4 have 1
  - Hash value
    - \( h_1(3) = 4 \) and \( h_2(3) = 0 \)

Calculating a single MinHash value with MapReduce

- Suppose you have some documents and have stored n-grams of these documents in a large table
- Each column
  - n-grams for a single document
- Each row \( r \)
  - \( r \)-th n-gram for all the documents
- Compute a minhash value for each of your documents using a single hash function

Approach 1. Partitioning by rows

- The table must be partitioned across the processors by rows
  - Each processor receives a subset of the n-grams for every document
- Can each processor compute the minhash for each document?
  - No. It should pass the hash values and group them based on the document ID (Assume that document ID is stored separately)
Design your map function

- What should be the input <key, value>?
- What should be the output <key, value>?

```java
map (k: docID, v: (row_number,freq) {
    if (freq >0)
        emit_intermediate(ik = k,
                        iv =hash(row_number))
}
```

Design your reduce function

- MapReduce system will group the hashes by documentID
- Reduce function will find the minimum hash value within the group

```java
reduce (ik: docID, ivs: hashval[]) {
    var minhash := INFINITY
    for each iv in ivs {
        if iv < minhash {
            minhash := iv
        }
    }
    emit_final(fk = ik, fv = minhash)
}
```

Approach 2. Partitioning by column

- All the k-grams for a document go to the same processor

```java
map (k: docID, v: kgram_row_with_value[]) {
    var minhash := INFINITY
    for each kgram with non 0 value in v {
        var h := hash(kgram_row)
        if h < minhash {
            minhash := h
        }
    }
    emit_intermediate(ik = k, iv = minhash)
}
```

Finding the most similar documents

- Number of pairs of documents to compare is too high
- Example
  - 1M documents, signatures of length 250 (4 Byte each)
  - $1M \times 1,000 \text{Bytes} = 1$ GB
  - Number of comparisons = $1M^2/2$ (Half of trillion pairs)
  - 1ms per calculation of similarity
  - 6 days to complete computing
- Do we need to calculate the similarity for all of the pairs?

Locality-sensitive hashing (LSH)
- Near-neighbor search
- LSH
  - Hash items several times, so that similar items are more likely to be hashed to the same bucket than dissimilar items
  - Assign the candidate pair to the same bucket
- Reduce false positives
  - Dissimilar pairs in the same bucket
- Reduce false negatives
  - Similar pairs in different buckets

Dividing a signature matrix into 4 bands and 3 rows per band

D1(1,2,1) and D13 (1,2,1) will go to the same bucket (1,3,1) and (0,3,0) will NOT go to the same bucket unless they show the same values in the other band.

Analysis of the Banding Technique
- Suppose that we use $b$ bands of $r$ rows each
- Suppose that a particular pair of documents have Jaccard Similarity $s$
  - $P$ (the minhash signatures for these documents agree in any one particular row of the signature matrix) = $s$

Threshold
- The value of similarity $s$ to determine two documents are similar
- An approximation of the threshold is $(1/b)^{1/r}$
Example

- For example
  - If there are 16 bands and each band contains 4 rows
    \[ S = 0.5 \]
  - If there are 8 bands and each band contains 8 rows
    \[ S = 0.77 \]
  - If there are 4 bands and each band contains 16 rows
    \[ S = 0.91 \]

Example

- Suppose that there are 20 bands (b=20), and each band includes 5 rows (r=5).
- \[ P(\text{the signatures agree in all the rows of at least one band}) = 1 - (1 - s r)^b \]

<table>
<thead>
<tr>
<th>( s )</th>
<th align="right">( 1 - (1 - s r)^b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>.2</td>
<td align="right">.006</td>
</tr>
<tr>
<td>.3</td>
<td align="right">.047</td>
</tr>
<tr>
<td>.4</td>
<td align="right">.186</td>
</tr>
<tr>
<td>.5</td>
<td align="right">.470</td>
</tr>
<tr>
<td>.6</td>
<td align="right">.802</td>
</tr>
<tr>
<td>.7</td>
<td align="right">.975</td>
</tr>
<tr>
<td>.8</td>
<td align="right">.9996</td>
</tr>
</tbody>
</table>