Sketching

AWissenschaftliches Arbeiten

Knut Reinert

Summer 2026

Introduction

Today

- You have to analyse some genomes using **distance based phylogenentic reconstructions** (like Neighbor Joining or UPGMA, see second lecture)
- You should estimate the genomic distance using the **Jaccard index** $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$ on the sets of k-mers of the genomes (i.e. for $k = 15$). That means you have to compute the union and intersection of all k-mers of the genomes. $1 - J(A, B)$ can be used as a distance measure.
- We will show you how the Jaccard index can be estimate using sketching techniques (this lecture).
- We will assume that using large sketches cost money (i.e. compute time).
- You are do **develop a strategy** to compute the distances of the genomes so precise, that correct phylogenies are computed.

Introduction

Sketching

Let x be some data: a set, a string, an integer, etc. A **data sketch** is the output of a randomized function f (in general, the combination of a certain number of hash functions) mapping x to a sequence of bits $f(x)$ with properties 1-3 below, plus (depending on the application) also property 4:

1. The bit-size of $f(x)$ is much smaller than the bit-size of x (usually, sub-linear or even poly-logarithmic).
2. $f(x)$ can be used to compute (efficiently) some properties of x. For example, if x is a multiset then $f(x)$ could be used to compute an approximation of the number of distinct elements contained in x, or the most frequent element in x.

Introduction

Sketching

3. $f(x)$ can be updated (efficiently) if x gets updated. Importantly, it should be possible to update $f(x)$ without knowing x. For example:
    - if we add an element y to a set x, it should be possible to compute $f(x \cup \{y\})$ knowing just $f(x)$ and y (not x).
    - More general, given two sketches $f(x_1)$ and $f(x_2)$, it should be possible to compute the sketch of the composition of x_1 and x_2 (under some operator). For example, if x_1 and x_2 are sets we could be interested in obtaining the sketch of $f(x_1 \cup x_2)$, without knowing x_1 and x_2.
4. If x and y are similar according to some measure of similarity (e.g. Euclidean distance), then $f(x)$ and $f(y)$ are likely to be similar (according to some measure of similarity, not necessarily the same as before).

MinHashing

MinHash is a sketching algorithm used to estimate the similarity of sets. It was invented by Andrei Broder in 1997 and initially used in the AltaVista search engine to detect duplicate web pages and eliminate them from search results. Here we report just a definition and analysis of MinHash. MinHash is a technique for estimating the Jaccard similarity (or index) $J(A, B)$ of two sets A and B:

Definition (Jaccard similarity)

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Without loss of generality, we may assume that we work with sets of integers from the universe $[1, n]$.

MinHashing

Definition (Minhash function)

Let h be a hash function. The MinHash hash function of a set A is defined as $h'(A) = \min\{h(x) : x \in A\}$, i.e. it is the minimum of h over all elements of A.

Definition (Minhash estimator)

Let $J'_h(A, B)$ be the indicator RV. defined as follows:
$J'_h(A, B) = 1,$ if $h'(A) = h'(B), J'_h(A, B) = 0$ otherwise.
Note that $J'_h(A, B)$ is a Bernoullian RV. We prove can the following remarkable property:

Lemma

If $h : [1, n] \to [1, n]$ is a uniform permutation, then $E[J'_h(A, B)] = J(A, B)$

MinHashing

Proof

Let $|A \cup B| = N$. For $i \in A \cup B$, consider the event $\mathit{smallest}(i) = (\forall j \in A \cup B - \{i\})(h(i) < h(j))$, stating that i is the element of $A \cup B$ mapped to the smallest hash $h(i)$ (among all elements of $A \cup B$).

Since h is a permutation, exactly one element from $A \cup B$ will be mapped to the smallest hash (i.e. $\mathit{smallest}(i)$ is true for exactly one $i \in A \cup B$), so $\{\mathit{smallest}(i)\}_{i \in A \cup B}$ is a partition of cardinality $N = |A \cup B|$ of the event space. Moreover, the fact that h is completely uniform implies that $P(\mathit{smallest}(i)) = P(\mathit{smallest}(j)) = 1/N$ for all $i, j \in A \cup B$.

Now, if $\mathit{smallest}(i)$ is true and $i \in A \cap B$, then $J'_h(A, B) = 1$, if i is not in the intersection, then the estimate is 0. It is then easy to write down the proof (exercise).

MinHashing

Reducing the variance

The RV $J'_h(A, B)$ is not a good estimator since it is a Bernoullian RV and thus h has a large variance: in the worst case ($J(A, B) = 0.5$), we have $Var[J'_h(A, B)] = 0.25$ and thus h the expected error (standard deviation) of $J'_h(A, B)$ is $\sqrt{Var[J'_h(A, B)]} = 0.5$.

This means that on expectation we are off by 50% from the true value of $J(A, B)$. We know how to solve this issue: just take the average of k independent such estimators, for sufficiently large k.

MinHashing

Reducing the variance

Let $h_i : [1, n] \to [1, n]$, with $i = 1, \ldots, k$, be k independent uniform permutations. We define the MinHash sketch of a set A to be the k-tuple: $h_{min}(A) = (h'_1(A), h'_2(A), \ldots, h'_k(A))$. Then, we estimate $J(A, B)$ using the following estimator:

Improved MinHash estimator

$$J^{+}(A, B) = \frac{1}{k} \sum_{i=1}^{k} J'_{h_i}(A, B)$$

Note that the improved MinHash estimator for a union of two sets can be computed in $O(k)$ time given the MinHash sketches of two sets.

Min Hashing

Concentration bound

We can immediately apply the double-sided additive Chernoff-Hoeffding bound and obtain that $P(|J^+(A,B) - J(A,B)| \geq \epsilon) \leq 2e^{-\epsilon^2 k/2}$ for any desired absolute error $0 < \epsilon \leq 1$. Fix now any desired failure probability $0 < \delta \leq 1$. By solving $2e^{-\epsilon^2 k/2} = \delta$ we obtain $k = 2\ln(2/\delta)/\epsilon^2$. We can finally state:

Theorem

Fix any desired absolute error $0 < \epsilon \leq 1$ and failure probability $0 < \delta \leq 1$. By using $k = 2\ln(2/\delta) \in O(\log(1/\delta))$ hash functions, the estimator $J^+(A,B)$ exceeds absolute error ϵ with probability of at most δ

$$P(|J^+(A,B) - J(A,B)| \geq \epsilon) \leq \delta$$

MinHashing

To summarize, we can squeeze down any subset of $[1, n]$ to a MinHash sketch of $O(\frac{\log(1/\delta)}{\epsilon^2})$ bits so that, later, in $O(\frac{\log(1/\delta)}{\epsilon^2})$ time we can estimate the Jaccard similarity between any pair of sets (represented with MinHash sketches) with arbitrarily small absolute error ϵ and arbitrarily small failure probability δ.

Note that it is easy to combine the MinHash sketches of two sets A and B so to obtain the MinHash sketch of $A \cup B$ (similarly, to compute the MinHash sketch of $A \cup \{x\}$ given the MinHash sketch of A):

$$h_{min}(A \cup B) = (\min\{h'_1(A), h'_1(B)\}, \ldots, \min\{h'_k(A), h'_k(B)\})$$

Note further that instead of using k independent hash functions (hard to find) we instead take simply the k minimum hash values.

Containment index

Consider the case of estimating the Jaccard index between two sets A and B of very different size. The traditional minhash randomly samples from the union $A \cup B$ and uses the number of sampled points that fall in $A \cap B$ to estimate the Jaccard index. With more sampled elements falling in $A \cap B$, the more accurate the Jaccard estimate will be.

Part A) of the next Figure demonstrates the case of sampling 100 random points from $A \cup B$ leading to 3 points lying in $A \cap B$. In the containment min hash approach, we randomly sample elements only from the smaller set (in this case, A) test if this element is in B (and hence in $A \cap B$).

This is used to estimate the containment index $C(A,B) = \frac{|A \cap B|}{|A|}$, which is then used to estimate the Jaccard index $J(A,B)$ itself. This can be done by computing

$$J^{est} = \frac{|A|C^{est}}{|A| + |B| - |A|C^{est}}$$

Containment index

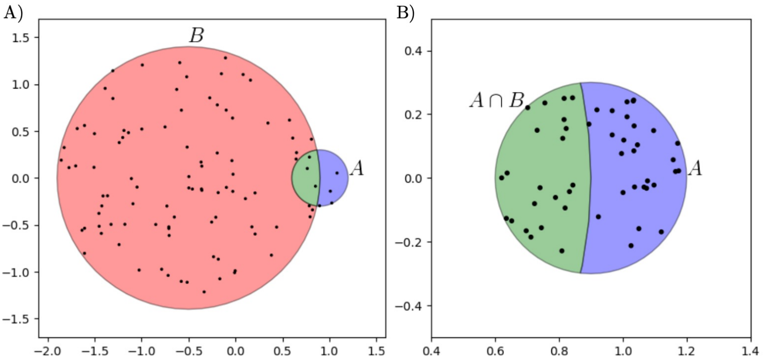

FIGURE 1. Conceptual comparison of classical min hash to the proposed containment approach when estimating the Jaccard index of very different sized sets. A) Sampling 100 points from $A \cup B$ (as is done in the classical min hash approach) leads to finding only 3 elements in $A \cap B$. B) Sampling just 50 points of A and testing if a point $x \in A \cap B$, finds 22 elements in $A \cap B$. This latter approach will be seen to lead to a better estimate of the Jaccard index.

Containment index

So how to we get a good estimate of the containment index? One answer was recently given by David Koslickis group. It is called FracMinHashing.

Definition

Given two arbitrary sets A and B which are subsets of the same domain Ω. the containment index $C(A,B)$ is defined as $C(A,B) := \frac{|A \cap B|}{|A|}$. Let h be a perfect hash function $h : \Omega \to [0,H]$ for some $H \in \mathbb{R}$. For a scale factor s where $0 \leq s \leq 1$, a FracMinHash sketch of a set A is defined as follows:

$$\mathbf{FRAC}_s(A) = \{h(a) \mid a \in A \text{ and } h(a) \leq Hs\}.$$

Containment index

Definition

The scale factor s is an easily tunable parameter that can modify the size of the sketch. Using this FracMinHash sketch, we define the FracMinHash estimate of the containment index $\hat{C}_{\text{frac}}(A, B)$ as follows:

$$\hat{C}_{\mathrm{frac}}(A, B) := \frac{|\mathbf{FRAC}_s(A) \cap \mathbf{FRAC}_s(B)|}{|\mathbf{FRAC}_s(A)|}.$$

It can be shown that

$$C_{\text{frac}}(A, B) := \frac{|\mathbf{FRAC}_s(A) \cap \mathbf{FRAC}_s(B)|}{|\mathbf{FRAC}_s(A)|\left(1 - (1 - s)^{|A|}\right)}$$

is an unbiased estimate of the containment index $C(A, B)$ called **the fractional containment index**.

Containment index

Two choices

You can either use the MinHashing approach to directly compute the Jaccard index or you can compute the Containment index first via FracMinHashing and derive from that the Jaccard index.

Cardinalities of sets

We omit here that one usually has to work with estimated cardinalities (e.g. using HLL sketches) and will provide the set cardinalities.