Metric Embedding into the Hamming Space with the n-Simplex Projection

Lucia VADICAMO
Fabrizio FALCHI

Institute of Information Science and Technologies, CNR, Pisa, Italy

Vladimir MIC
Pavel ZEZULA

Faculty of Informatics
Masaryk University
Brno, Czech Republic

2nd October 2019

Motivation & Preliminaries

- An efficient similarity search is nowadays necessary to process big volumes of complex data
- The similarity model: metric space (D, d)
- Transformations of the space (D, d) to Hamming space $(\{0,1\}^\lambda,\ h)$ are suitable to facilitate searching in big volumes of data
- Notation:
  - bit-strings are *sketches*
  - techniques transforming metric spaces to Hamming spaces are *sketching techniques*

Transformations to Hamming Space

- Many sketching techniques were proposed
- No generally best sketching technique exists

Transformations to Hamming Space

- Many sketching techniques were proposed
- No generally best sketching technique exists
  - their quality is data dependent

Transformations to Hamming Space

- Many sketching techniques were proposed
- No generally best sketching technique exists
  - their quality is data dependent
  - they are of a different applicability
    - limit the metric space (D, d) to be e.g. the Euclidean space, vector space, arbitrary metric space, . . .

Transformations to Hamming Space

- Many sketching techniques were proposed
- No generally best sketching technique exists
  - their quality is data dependent
  - they are of a different applicability
    - limit the metric space (D, d) to be e.g. the Euclidean space, vector space, arbitrary metric space, ...
  - they require various costs of
    - transformation learning *(before the search)*
    - transformation of objects $o \in D$ to sketches
    - pre-processing of the searched dataset *(before the search)*
    - transformation of the query object $q \in D$ to the query sketch $sk(q)$ *(during the search)*

Transformations to Hamming Space

- Many sketching techniques were proposed
- No generally best sketching technique exists
  - their quality is data dependent
  - they are of a different applicability
    - limit the metric space (D, d) to be e.g. the Euclidean space, vector space, arbitrary metric space, ...
  - they require various costs of
    - transformation learning *(before the search)*
    - transformation of objects $o \in D$ to sketches
    - pre-processing of the searched dataset *(before the search)*
    - transformation of the query object $q \in D$ to the query sketch $sk(q)$ *(during the search)*

Motivation to Propose Transformation Technique

- We propose a novel sketching technique, and we want to achieve a good trade-off between:

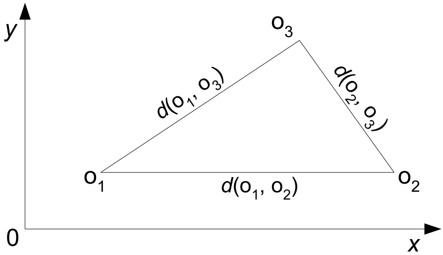
1. quality of the space approximation
2. applicability of technique
3. cost of the transformation learning
4. cost of the object transformation

Overview of the Proposed Sketching Technique

- Proposed *NSP_50* sketching technique:
  - exploits the *n-Simplex projection* to transform the given metric space to the Euclidean vector space
  - binarizes the Euclidean space to Hamming space

n-Simplex Property

- the n-Simplex projection is applicable to spaces with *n-point property*:
- the n-point property:
  - *„any n points $o_1, .., o_n \in D$ can be isometrically embedded into the $(n-1)$-dimensional Euclidean vector space"*
  - example: each metric space meets the 3-point property *(– due to the triangle inequality)*

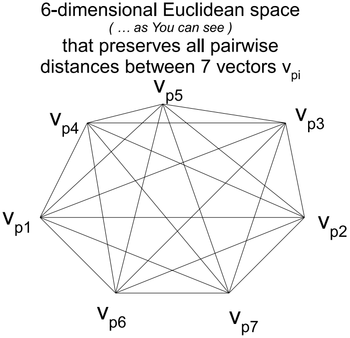


n-Simplex Projection

- n-Simplex projection exploits the n-point property:
  - n pivots $p_i \in D$ can be isometrically embedded into $(n-1)$-dimensional Euclidean space

n-Simplex Projection

- n-Simplex projection exploits the n-point property:
  - n pivots $p_i \in D$ can be isometrically embedded into $(n-1)$-dimensional Euclidean space
  - Example for $n = 7$: each p_i transformed to the 6-dimens. vector v_{pi}:



$$V_{p1} = (v_{11}; v_{12}; v_{13}; v_{14}; v_{15}; v_{16})$$
$$V_{p2} = (v_{21}; v_{22}; v_{23}; v_{24}; v_{25}; v_{26})$$
$$V_{p3} = (v_{31}; v_{32}; v_{33}; v_{34}; v_{35}; v_{36})$$
$$V_{p4} = (v_{41}; v_{42}; v_{43}; v_{44}; v_{45}; v_{46})$$
$$V_{p5} = (v_{51}; v_{52}; v_{53}; v_{54}; v_{55}; v_{56})$$
$$V_{p6} = (v_{61}; v_{62}; v_{63}; v_{64}; v_{65}; v_{66})$$
$$V_{p7} = (v_{71}; v_{72}; v_{73}; v_{74}; v_{75}; v_{76})$$

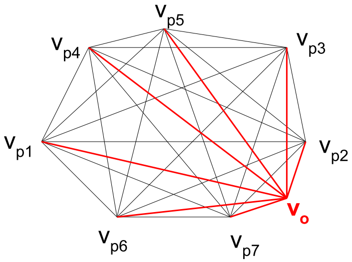
Figure: 7 pivots isometrically embedded into 6-dimensional Euclidean vector space

n-Simplex Projection

- (n+1)-point property guarantees it is possible to isometrically embed next object $o \in D$ while adding a dimension to the Euclidean space:
  - o is transformed to vector v_o
  - a new coordinate is added to all v_{pi} vectors
  - both is done in a way that all pairwise distances are still preserved

n-Simplex Projection

- (n+1)-point property guarantees it is possible to isometrically embed next object $o \in D$ while adding a dimension to the Euclidean space:
    - o is transformed to vector v_o
    - a new coordinate is added to all v_{pi} vectors
    - both is done in a way that all pairwise distances are still preserved



$V_{p1} = (v_{11}; v_{12}; v_{13}; v_{14}; v_{15}; v_{16}; 0)$
$V_{p2} = (v_{21}; v_{22}; v_{23}; v_{24}; v_{25}; v_{26}; 0)$
$V_{p3} = (v_{31}; v_{32}; v_{33}; v_{34}; v_{35}; v_{36}; 0)$
$V_{p4} = (v_{41}; v_{42}; v_{43}; v_{44}; v_{45}; v_{46}; 0)$
$V_{p5} = (v_{51}; v_{52}; v_{53}; v_{54}; v_{55}; v_{56}; 0)$
$V_{p6} = (v_{61}; v_{62}; v_{63}; v_{64}; v_{65}; v_{66}; 0)$
$V_{p7} = (v_{71}; v_{72}; v_{73}; v_{74}; v_{75}; v_{76}; 0)$
$\mathbf{V_o = (v_1; v_2; v_3; v_4; v_5; v_6; v_7)}$

Please notice values added to vectors v_{pi} must be the same to preserve distances between these vectors

Contribution: *NSP_50* Sketching Technique

- We propose the *NSP_50* sketching technique that transforms metric spaces with the n-point property to Hamming space:
  - It selects n pivots
  - transforms all data-objects to n-dimensional Euclidean space by the n-Simplex projection
  - [1] evaluates the median value for each coordinate of vectors v_o, and binarize them:
    - sets 0 iff the value in the vector is smaller then the median
  - number of pivots n thus also defines the length of produced sketches

[1] before this step, we randomly rotate the Euclidean space to distribute the information over coordinates

Compared Sketching Techniques

- We compare the *NSP_50* technique experimentally and theoretically with other sketching techniques:
  - The *GHP_50* uses the *generalyzed hyperplane partitioning* (GHP) to split dataset into approx. halves. Each instance of the GHP determines the value of one bit in all sketches. The *GHP_50* produces sketches with low correlated bits.
  - The *BP_50* uses the *ball partitioning* (BP) to split data into halves to set values in a bit of all sketches. Also aims to produce sketches with low correlated bits.
  - The *PCA_50* use the principal component analysis to shorten vectors in the Euclidean space. Then it binarizes the vectors in a same way as *NSP_50*.

Properties of Sketching Techniques

- Proper analysis is in the paper
- The main features of sketching techniques:

*NSP*_50

- wide applicability
- good quality of space approximation
- cheap transformation learning
- λ distance computations and λ^2 flops to transform object

*GHP*_50

- very wide applicability
- still a good space approximation
- expensive transformation learning
- 2λ distance computations to transform object

*BP*_50

- very wide applicability
- very poor approximation quality when applied to complex spaces
- expectable transformation learning cost
- λ distance computations to transform object

*PCA*_50

- narrow applicability to Euclidean spaces *(could be partially extended)*
- very good space approximation
- cheap transformation learning
- $\lambda \cdot$ "*space dim*" flops to transform object

Experiments – Test Data

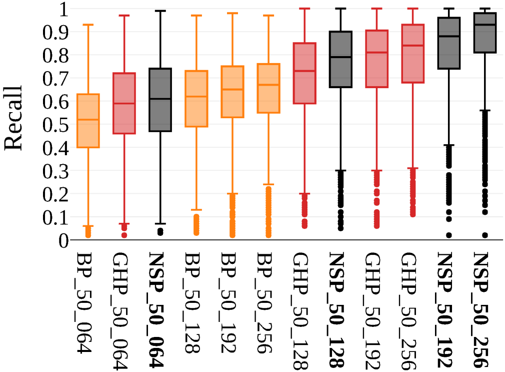
- We search for 100 nearest neighbours in 1 million datasets of image visual descriptors
  - *DeCAF* descriptors: Euclidean space of 4,096 dim. vectors extracted from the *Profiset image collection* using the Deep Convolutional Neural Network
  - SIFT descriptors from the *ANN dataset* that form the Euclidean space with 128 dimensions
  - *Adaptive-binning feature histograms* compared by the Signature Quadratic Form Distance (SQFD), extracted from the *Profiset image collection*. Each signature consists of, on average, 60 cluster centroids in a 7-dimensional space.

Experiments – Setup

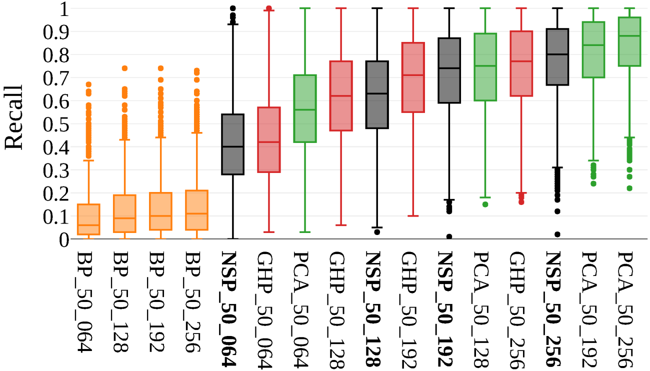
- We randomly select 1,000 query objects q for each dataset, and we compare the precise query answer with the approximate one:
- Approximate evaluation based on sketches:
  - For each q, we pre-select the *candidate set* $candSet(q)$ of 2,000 descriptors from the dataset (0.2 %) with the most similar sketches $sk(o)$ to the sketch of the query object $sk(q)$
  - We evaluate the distances $d(q, o), o \in candSet(q)$ to return 100 most similar objects from the candidate set
- Precise query answer:
  - 100 objects o from the dataset with minimum distances $d(q, o)$ to the query object q
- Comparison:
  - the recall expresses the relative size of the intersection of the approximate and the precise query answer

Experiments – Results SQFD dataset

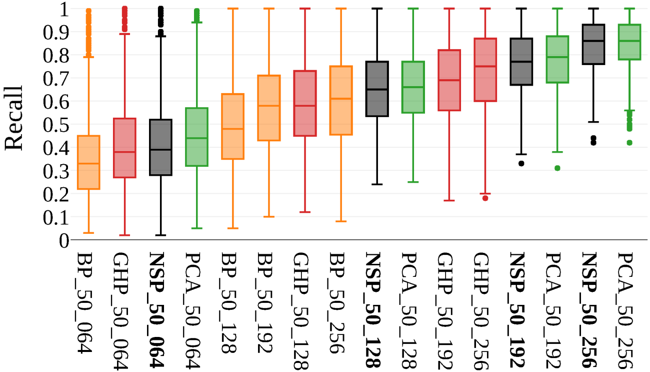
- Box-plots express the distribution of the recall values per query objects
- Suffix in the name of the sketching technique expresses the length of sketches (in bits)



Experiments – Results DeCAF dataset



Experiments – Results SIFT dataset



Conclusions

- We have proposed a novel sketching technique NSP_50
- based on the n-Simplex projection
- provides a good trade-off between applicability, quality and costs of transformation

Proper comparison

Comparison table:
http://www.nmis.isti.cnr.it/falchi/SISAP19SM.pdf