Designing Sketches for Similarity Filtering

Vladimir Mic David Novak Pavel Zezula

Faculty of Informatics
Masaryk University
Brno, Czech Republic

December 12, 2016

Motivation

- Handling objects according to their pairwise similarity closely corresponds to the human perception of reality:
  - example: *little children use similarity relations as a predominant basis for classification* [1]

[1] D. G. Kemler, *"Classification in young and retarded children: The primacy of overall similarity relations,"* 1982

Motivation

- Handling objects according to their pairwise similarity closely corresponds to the human perception of reality:
  - example: *little children use similarity relations as a predominant basis for classification* [1]
- Similarity of objects
  - example of objects: images, plots, time series, fingerprints, motions, sounds, music ...
  - similarity: visual (in general), similarity of shapes, colours, subsequences, subfigures

[1] D. G. Kemler, *"Classification in young and retarded children: The primacy of overall similarity relations,"* 1982

Motivation

- Handling objects according to their pairwise similarity closely corresponds to the human perception of reality:
  - example: *little children use similarity relations as a predominant basis for classification* [1]
- Similarity of objects
  - example of objects: images, plots, time series, fingerprints, motions, sounds, music . . .
  - similarity: visual (in general), similarity of shapes, colours, subsequences, subfigures
- Similarity search, query by example
  - Find most similar objects to given query object q in (big) dataset X
  - Goal: do it quickly, possibly in a real time
  - Common approach: provide an approximate answer

[1] D. G. Kemler, *"Classification in young and retarded children: The primacy of overall similarity relations,"* 1982

Similarity Model

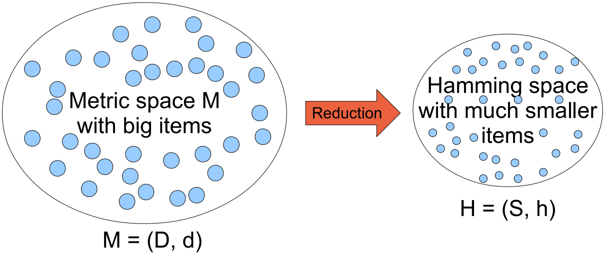
- Similarity search is usually performed on characteristic features extracted from objects
- Domain of these features is D
- Similarity of two objects is described by similarity function
  - we use an opposite approach: a *distance function d* which measures dissimilarity of objects
  - The bigger the value $d(x, y)$ is, the less similar objects x, y are

Similarity Model

- Similarity search is usually performed on characteristic features extracted from objects
- Domain of these features is D
- Similarity of two objects is described by similarity function
  - we use an opposite approach: a *distance function d* which measures dissimilarity of objects
  - The bigger the value $d(x, y)$ is, the less similar objects x, y are
- Similarity model: the Metric space (D, d)
  $\forall x, y, z \in D$:
  - $d(x, y) \geq 0$ (non-negativity)
  - $d(x, y) = d(y, x)$ (symmetry)
  - $d(x, y) = 0 \iff x = y$ (identity)
  - $d(x, y) + d(y, z) \geq d(x, z)$ (triangle inequality)

Our Approach – Instance of Dimensionality Reduction

Dimensionality reduction of the Metric space to Hamming space:



- M: general Metric space
- S: domain of bit-strings of length λ
- h: Hamming distance = the number of different bits in two bit strings

Our Approach – Goal

- Bit-string $sk(o)$ created for object $o \in D$ is called sketch of object o

$Sk(o)$:

1	0	1	1	0	0	0	0

- Goal: create short sketches well reflecting spatial relationships between objects in the Metric space M

Our Approach – Goal

- Bit-string $sk(o)$ created for object $o \in D$ is called sketch of object o

$Sk(o)$:

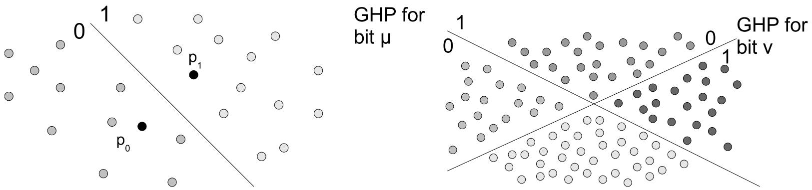
1	0	1	1	0	0	0	0

- Goal: create short sketches well reflecting spatial relationships between objects in the Metric space M
  - well approximate object ordering with respect to an arbitrary query q: $d(q, o_1) < d(q, o_2) \implies h(Sk(q), Sk(o_1)) < h(Sk(q), Sk(o_2))$
- Possible usage: *Filter and Refine* similarity search
  - *Filter*: having a query q, filter dataset X using Hamming distances $h(sk(q), sk(o)), o \in X$,
  - *Refine*: evaluate distance $d(q, o)$ for objects o whose sketches $sk(o)$ have small Hamming distances $h(sk(q), sk(o))$

Sketching Technique

Sketching technique suitable for the Metric space:

- dataset X is divided by Generalized hyperplane partitioning (GHP) into two parts
- first bit of all sketches $sk(o), o \in X$ is set according to this division to 1 or 0
- another instance of GHP is selected to set another bit etc.



(a) GHP to set one bit (b) Two instances of GHP to set bits μ and ν

- Key question: how to select pivots p_0 and p_1 for GHPs?

GHP – Pivot Selection

The following properties improve the accuracy of the approximation of the ordering with respect to an arbitrary query. Having a dataset X:

- each bit of sketches should be set to 1 in one half of sketches (balanced bits)

$Sk(o_1)$:	1	0	1	1	0	0	1	1
$Sk(o_2)$:	1	0	0	0	1	1	0	1
$Sk(o_3)$:	0	1	0	1	1	0	0	0
$Sk(o_4)$:	0	1	1	0	0	1	1	0

Example: four sketches with balanced bits

- sketches should have low pairwise correlated bits
  - absolute value of Pearson correlation coefficient

Experiments – procedure

Experiments:

- Datasets X: 1M visual descriptors of images,
  - *DeCAF* dataset: 4,096 dimensional vectors, Euclidean distance (L_2)
  - *Cophir* dataset: 280 dimensional vectors, weighted sum of L_1 and L_2 distances

Experiments – procedure

Experiments:

- Datasets X: 1M visual descriptors of images,
  - *DeCAF* dataset: 4,096 dimensional vectors, Euclidean distance (L_2)
  - *Cophir* dataset: 280 dimensional vectors, weighted sum of L_1 and L_2 distances
- Precise *k nearest neighbours* (kNN) query evaluation:
  - For a given query q evaluate all distances $d(q, o), o \in X$ and return k nearest objects

Experiments – procedure

Experiments:

- Datasets X: 1M visual descriptors of images,
  - *DeCAF* dataset: 4,096 dimensional vectors, Euclidean distance (L_2)
  - *Cophir* dataset: 280 dimensional vectors, weighted sum of L_1 and L_2 distances
- Precise *k nearest neighbours* (kNN) query evaluation:
  - For a given query q evaluate all distances $d(q, o), o \in X$ and return k nearest objects
- Approximate kNN query evaluation:
  - For query q select k' sketches with small Hamming distances $h(sk(q), sk(o)), o \in X$ from query sketch $sk(q)$
  - Objects $o \in X$ corresponding to these sketches form a *CandidateSet(q)*
  - Evaluate similarity $d(q, o), o \in$ *CandidateSet(q)* to determine k most similar objects

Experiments – procedure

Experiments:

- Datasets X: 1M visual descriptors of images,
  - *DeCAF* dataset: 4,096 dimensional vectors, Euclidean distance (L_2)
  - *Cophir* dataset: 280 dimensional vectors, weighted sum of L_1 and L_2 distances
- Precise *k nearest neighbours* (kNN) query evaluation:
  - For a given query q evaluate all distances $d(q, o), o \in X$ and return k nearest objects
- Approximate kNN query evaluation:
  - For query q select k' sketches with small Hamming distances $h(sk(q), sk(o)), o \in X$ from query sketch $sk(q)$
  - Objects $o \in X$ corresponding to these sketches form a *CandidateSet(q)*
  - Evaluate similarity $d(q, o), o \in$*CandidateSet(q)* to determine k most similar objects
- Comparison: size of intersection of approximate answer with the precise one divided by k. (Denoted k-recall@k')

GHP – Proper Pivot Selection Results

Results, $|X| = 1\text{M}$, $k = 10$, sketch length $\lambda = 64$ bits. Three curves:

1. randomly selected pivots
2. pivots producing balanced bits
   - selected from superset of pivots, evaluated on a sample set of 100K objects, bits balanced with tolerance 5 %
3. balanced bits with low pairwise correlations

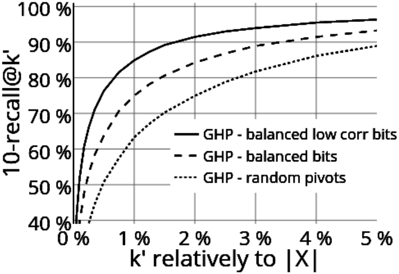


Figure: Dependency of 10-recall@k' on k'

Sketching Technique – Sketch Length Determination

Another question: what is a suitable length of sketches for particular data?
Example: fixing desired level of recall:

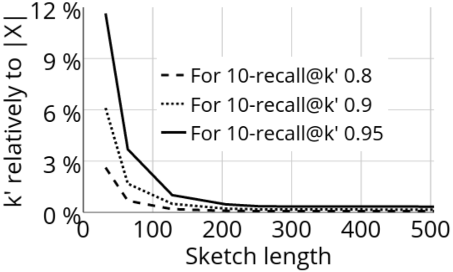


Figure: Dependency of k' on Sketch length fixing recall value 10-recall@k'

In this case we assume a suitable sketch length to be 200 – 240 (depends on preferences)

Formal Basics – Intrinsic Dimensionality

- *Intrinsic dimensionality* (*iDim*) – *the minimum number of parameters needed to account for the observed properties of the data*
- *iDim* describes the *data complexity*

Formal Basics – Intrinsic Dimensionality

- *Intrinsic dimensionality* (*iDim*) – *the minimum number of parameters needed to account for the observed properties of the data*
- *iDim* describes the *data complexity*

Other authors say:

- *Dimensionality reduction is an ill-posed problem that can only be solved by assuming certain properties of the data (such as its intrinsic dimensionality)*
- *Ideally, the reduced representation should have a dimensionality that corresponds to the intrinsic dimensionality of the data*

Formal Basics – Intrinsic Dimensionality

- *Intrinsic dimensionality* (*iDim*) – *the minimum number of parameters needed to account for the observed properties of the data*
- *iDim* describes the *data complexity*

Other authors say:

- *Dimensionality reduction is an ill-posed problem that can only be solved by assuming certain properties of the data (such as its intrinsic dimensionality)*
- *Ideally, the reduced representation should have a dimensionality that corresponds to the intrinsic dimensionality of the data*

Our approach:

1. Measure *iDim* of data and use it to estimate suitable sketch length
   - We use Chávez's formula $iDim = \frac{\mu^2}{2 \cdot \sigma^2}$, based on mean μ and variance σ^2 of distance distribution
2. Assume that *iDim* of created sketches will not be very different from *iDim* of data

Our Findings

In this paper we derive relationship between:

- *iDim* of sketches,
- length of sketches λ,
- average pairwise bit correlation c

For sketches with balanced bits we transform Chávez's formula:

$$iDim \approx \frac{\lambda}{2 \cdot (1 + (\lambda - 1) \cdot c^2)} \tag{1}$$

Observations:

- *iDim* of sketches decreases with the second power of correlation c

Our Findings

In this paper we derive relationship between:

- *iDim* of sketches,
- length of sketches λ,
- average pairwise bit correlation c

For sketches with balanced bits we transform Chávez's formula:

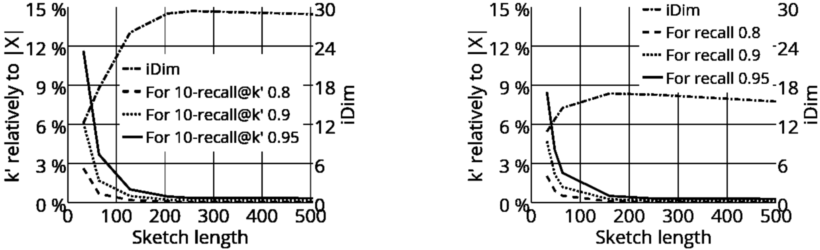
$$iDim \approx \frac{\lambda}{2 \cdot (1 + (\lambda - 1) \cdot c^2)} \tag{1}$$

Observations:

- *iDim* of sketches decreases with the second power of correlation c
- Search for λ bits with lowest pairwise correlations in a set of λ' bits has complexity $O(\lambda'^{\lambda} \cdot \lambda^2)$ – a heuristic must be used
- If we want low correlated bits, the correlation c grows with sketch length λ

Suitable Sketch Length Estimation

Let us focus on *iDim* of sketches and its relationship to observed recall:



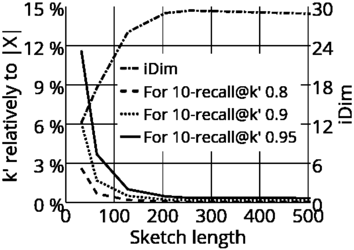
(a) DeCAF dataset – *iDim* 26.9

(b) Cophir dataset – *iDim* 12.7

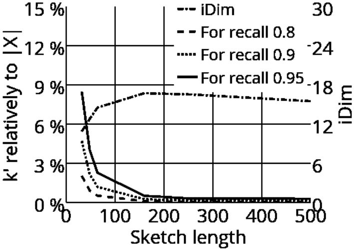
Figure: Dependency of k' needed to achieve given 10-recall@k' on sketch length λ

Suitable Sketch Length Estimation

Let us focus on *iDim* of sketches and its relationship to observed recall:



(a) DeCAF dataset – *iDim* 26.9



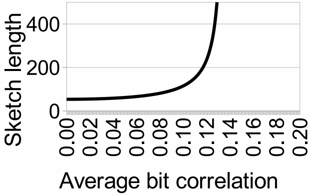
(b) Cophir dataset – *iDim* 12.7

1. *iDim* achieves its maximum for a certain length λ and then decreases – due to too big increase of average bit correlation c in sketches
2. Maximal *iDim* of sketches well corresponds to *iDim* of the original space – it is slightly higher
3. Length of sketches with maximal *iDim* well corresponds to suitable length of sketches for similarity search

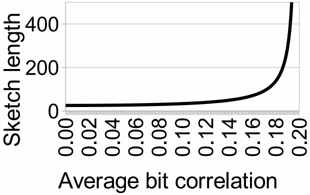
Suitable Sketch Length Estimation

Suitable sketch length estimation:

1. Measure *iDim* of data
2. Assume that produced sketches will have the same *iDim*[2]
3. Substitute this *iDim* to Equation 1 to get dependency of sketch length λ on bit correlation c:



(a) *iDim* 26.9 (DeCAF dataset)



(b) *iDim* 12.7 (Cophir dataset)

Observation: only some combination of bit correlation c and sketch length λ on the given curve are reachable. A difficulty of finding λ bits with given correlation c is related to slope of tangent of this function.

[2] This step is discussed in a paper

Analysis

Results: sketch length λ, k' for given recall, correlation c, slope of tangent

λ	k' for recall 0.9	k' for recall 0.95	*iDim* sketches	c	slope of tangent
DeCAF descriptors (*iDim*: 26.9)					
128	0.50 %	1.01 %	26.1	0.107	3,428
205	0.22 %	0.47 %	29.0	0.111	9,960
256	0.17 %	0.35 %	29.4	0.115	16,095
4,096	0.05 %	0.12 %	21.9	0.150	4,630,746
CoPhIR dataset (*iDim*: 12.7)					
64	1.19 %	2.27 %	14.5	0.138	1,284
160	0.25 %	0.51 %	16.7	0.154	9,614
256	0.16 %	0.30 %	16.5	0.163	25,557
2,048	0.09 %	0.18 %	9.0	0.235	1,721,419

Conclusion: the length of sketches with high *iDim*, which is suitable for the similarity search can be estimated according to slope of tangent: we recommend value 10,000 – 15,000

Analysis

Results: sketch length λ, k' for given recall, correlation c, slope of tangent

λ	k' for recall 0.9	k' for recall 0.95	*iDim* sketches	c	slope of tangent	
DeCAF descriptors (*iDim*: 26.9)						
128	0.50 %	1.01 %	26.1	0.107		3,428
205	0.22 %	0.47 %	29.0	0.111	(9,443)	9,960
256	0.17 %	0.35 %	29.4	0.115	(15,183)	16,095
4,096	0.05 %	0.12 %	21.9	0.150		4,630,746
CoPhIR dataset (*iDim*: 12.7)						
64	1.19 %	2.27 %	14.5	0.138		1,284
160	0.25 %	0.51 %	16.7	0.154	(8,037)	9,614
256	0.16 %	0.30 %	16.5	0.163	(21,812)	25,557
2,048	0.09 %	0.18 %	9.0	0.235		1,721,419

Conclusion: the length of sketches with high *iDim*, which is suitable for the similarity search can be estimated according to slope of tangent: we recommend value 10,000 – 15,000

Summary

Sketches – short bit-strings suitable for the similarity filtering

Our paper contains:

- proposal of sketching technique which produce sketches with defined properties
  - balanced bits
  - low correlated bits
- formal procedure to estimate suitable sketch length
  - according to intrinsic dimensionality of data