Distributed Multi-modal Similarity Retrieval

David Novak

Seminar of DISA Lab, October 14, 2014
Outline of the Talk

1 Motivation
   - Similarity Search
   - Effectiveness and Efficiency
   - Multi-modal Search

2 Existing Solutions
   - Similarity Indexing
   - Distributed Key-value Stores

3 Big Data Similarity Retrieval
   - Generic Architecture
   - Specific System

4 Conclusions
Motivation

- The similarity is key to human cognition, learning, memory... [cognitive psychology]
Motivation

- The similarity is key to human cognition, learning, memory...
  [cognitive psychology]
- everything we can see, hear, measure, observe is in digital form
Motivation

- The similarity is key to human cognition, learning, memory...
  [cognitive psychology]
- Everything we can see, hear, measure, observe is in digital form
- Therefore, computers should be able to search data base on similarity
The similarity is key to human cognition, learning, memory… [cognitive psychology]

everything we can see, hear, measure, observe is in digital form

Therefore, computers should be able to search data base on similarity

The similarity search problem has two aspects

effectiveness: how to measure similarity of two “objects”
  - domain specific (photos, X-rays, MRT results, voice, music, EEG,…)

David Novak
Multi-modal Similarity Retrieval
Motivation

- The similarity is key to human cognition, learning, memory…
  [cognitive psychology]
- everything we can see, hear, measure, observe is in digital form
- Therefore, computers should be able to search data base on similarity

The similarity search problem has two aspects

- **effectiveness**: how to measure similarity of two “objects”
  - domain specific (photos, X-rays, MRT results, voice, music, EEG,…)

- **efficiency**: how to realize similarity search fast
  - using a given similarity measure
  - on very large data collections
Efficiency: Motivation Example

Type of data:
- general images (photos)
Efficiency: Motivation Example

Type of data:

- general images (photos)
- every image has been processed by a deep neural network
  - to obtain a “semantic characterization” of the image (descriptor)
Efficiency: Motivation Example

Type of data:

- general images (photos)
- every image has been processed by a deep neural network
  - to obtain a “semantic characterization” of the image (descriptor)
  - compared by Euclidean distance, it measures visual similarity of images
Efficiency: Motivation Example

Type of data:

- general images (photos)
- every image has been processed by a deep neural network
  - to obtain a “semantic characterization” of the image (descriptor)
  - compared by Euclidean distance, it measures visual similarity of images
Efficiency: Motivation Example

Type of data:

- general images (photos)
- every image has been processed by a deep neural network
  - to obtain a “semantic characterization” of the image (descriptor)
  - compared by Euclidean distance, it measures visual similarity of images
Efficiency: Motivation Example

Type of data:
- general images (photos)
- every image has been processed by a deep neural network
  - to obtain a “semantic characterization” of the image (descriptor)
  - compared by Euclidean distance, it measures visual similarity of images

Efficiency problem:
- what if we had 100 million of images with such descriptors
- each descriptor is a 4096-dimensional float vector
Efficiency: Motivation Example

Type of data:
- General images (photos)
- Every image has been processed by a deep neural network to obtain a “semantic characterization” of the image (descriptor)
- Compared by Euclidean distance, it measures visual similarity of images

Efficiency problem:
- What if we had 100 million of images with such descriptors
- Each descriptor is a 4096-dimensional float vector
- ⇒ over 1.5 TB of data to be organized for similarity search
  - Answer similarity queries online

David Novak
Multi-modal Similarity Retrieval
DISA Seminar 4 / 17
Real Application: Multi-field Data

- real-world application data objects would have many “fields”:
  - attribute fields (numbers, strings, dates, etc.)
  - (several) descriptors for similarity search
  - keywords/annotations for full-text search, etc.
Real Application: Multi-field Data

- real-world application data objects would have many “fields”:
  - attribute fields (numbers, strings, dates, etc.)
  - (several) descriptors for similarity search
  - keywords/annotations for full-text search, etc.

- example:

```json
{ "ID": "image_1",
  "author": "David Novak",
  "date": "20140327",
  "categories": [ "outdoor", "family" ],
  "DNN_visual_descriptor": [5.431, 0.0042, 0.0, 0.97,... ],
  "dominant_color": "0x9E, 0xC2, 0x13",
  "keywords": "summer, beach, ocean, sun, sand" }
```
Objectives

Goal: generic, horizontally scalable system architecture that would allow
- standard attribute-based access
- keyword (full-text) search
- similarity search in “arbitrary” similarity space
Objectives

Goal: generic, horizontally scalable system architecture that would allow

- standard attribute-based access
- keyword (full-text) search
- similarity search in “arbitrary” similarity space
- multi-modal search – combination of several search perspectives, e.g.
  - direct combination of similarity modalities
  - similarity query with filtering by attribute(s)
  - re-ranking of search result by different criteria
Objectives

Goal: generic, horizontally scalable system architecture that would allow

- standard attribute-based access
- keyword (full-text) search
- similarity search in “arbitrary” similarity space
- multi-modal search – combination of several search perspectives, e.g.
  - direct combination of similarity modalities
  - similarity query with filtering by attribute(s)
  - re-ranking of search result by different criteria

... and do it all on a very large scale

- voluminous data collections
- high query throughput
Distance-based Similarity Search

- generic similarity search
  - applicable to many domains
- data modeled as metric space \((\mathcal{D}, \delta)\), where \(\mathcal{D}\) is a domain of objects and \(\delta\) is a total distance function \(\delta : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}_0^+\) satisfying postulates of identity, symmetry, and triangle inequality
Distance-based Similarity Search

- generic similarity search
  - applicable to many domains
- data modeled as metric space \((\mathcal{D}, \delta)\), where \(\mathcal{D}\) is a domain of objects and \(\delta\) is a total distance function \(\delta : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}_0^+\) satisfying postulates of identity, symmetry, and triangle inequality
- query by example: \(K\text{-NN}(q)\) returns \(K\) objects \(x\) from the dataset \(\mathcal{X} \subseteq \mathcal{D}\) with the smallest \(\delta(q, x)\)
Similarity Indexing Techniques

Metric-based similarity indexing: two decades of research

- memory structures for precise $K$-NN search
Similarity Indexing Techniques

Metric-based similarity indexing: two decades of research

- memory structures for precise $K$-NN search
- efficient disk-oriented techniques
  - precise and approximate (not all objects from $K$-NN answer returned)
Similarity Indexing Techniques

Metric-based similarity indexing: two decades of research
- memory structures for precise $K$-NN search
- efficient disk-oriented techniques
  - precise and approximate (not all objects from $K$-NN answer returned)
  - objects are partitioned and organized on disk by the similarity metric
**Similarity Indexing Techniques**

Metric-based similarity indexing: two decades of research

- memory structures for precise $K$-NN search
- efficient disk-oriented techniques
  - precise and approximate (not all objects from $K$-NN answer returned)
  - objects are partitioned and organized on disk by the similarity metric

- given query $q$, the “most-promising” partitions form the candidate set
Similarity Indexing Techniques

**Metric-based similarity indexing: two decades of research**

- **memory** structures for precise $K$-NN search
- **efficient** disk-oriented techniques
  - precise and **approximate** (not all objects from $K$-NN answer returned)
  - objects are **partitioned** and organized on disk by the similarity metric

- given query $q$, the "most-promising" partitions form the **candidate set**
- the candidate set $S_C$ is **refined** by calculating $\delta(q, x), \forall x \in S_C$
Recently, there were proposed a few indexes of different type
Recently, there were proposed a few indexes of different type

- memory index that organizes only metadata

Recently, there were proposed a few indexes of different type

- **memory index** that organizes only metadata


![Diagram](image-url)
Distributed Similarity Indexes

Distributed Data Structures for metric-based similarity search

- data partitioned to nodes according to the metric
- at query time, query-relevant partitions (nodes) accessed
Distributed Similarity Indexes

Distributed Data Structures for metric-based similarity search

- data partitioned to nodes according to the metric
- at query time, query-relevant partitions (nodes) accessed
  - GHT*, VPT*, MCAN, M-Chord
Distributed Similarity Indexes

Distributed Data Structures for metric-based similarity search

- data partitioned to nodes according to the metric
- at query time, query-relevant partitions (nodes) accessed
  - GHT*, VPT*, MCAN, M-Chord

(a) (b)
Current Distributed Stores

Currently, many efficient distributed key-value or document stores emerged

- distributed hash tables
- objects organized by IDs (ID-object map)
  - quick access to “documents” by IDs
- secondary indexes on attributes
Current Distributed Stores

Currently, many efficient distributed key-value or document stores emerged:

- distributed hash tables
- objects organized by IDs (ID-object map)
  - quick access to “documents” by IDs
- secondary indexes on attributes
Generic Architecture

key-value store (ID-object) on the whole dataset $X$
Generic Architecture

key-value store (ID-object) on the whole dataset $X$
Generic Architecture

key-value store (ID-object) on the whole dataset $X$

$k$-NN($q$.field)

similarity index $I^\text{field}_{Xi}$

candidate set $C_{Xi} \subseteq X_i$

merge partial answers

refinement on part of $C_{Xi}$

refinement on part of $C_{Xi}$

merge partial answers

refinement on part of $C_{Xi}$
Generic Architecture

- **k-NN(q.field)**
- **similarity index** $I_{X_i}^{field}$
- **inverted file index** $I_{X_i}^{field2}$
- **attribute index** $I_{X_i}^{field3}$
- **key-value store (ID-object) on the whole dataset $X$**

Steps:
1. Candidate set $C_{Xi} \subseteq X_i$
2. Refinement on part of $C_{Xi}$
3. Merge partial answers

Diagram:
- Workers connecting to indexes and databases.
System Features

Types of queries

- **ID-object** query (often useful to initiate $k$-NN($q$) query)
- **attribute**-based queries (secondary indexes)
- **key-word** (full-text) queries (Lucene-like index)
- **similarity** queries (via similarity indexes)
System Features

Types of queries

- **ID-object** query (often useful to initiate $k$-NN($q$) query)
- **attribute**-based queries (secondary indexes)
- **key-word** (full-text) queries (Lucene-like index)
- **similarity** queries (via similarity indexes)
- **combined** similarity queries (late fusion)
- $K$-NN query with attribute **filtering**
- distributed **re-ranking** query answer
System Features

Types of queries

- **ID-object** query (often useful to initiate $k$-NN($q$) query)
- attribute-based queries (secondary indexes)
- key-word (full-text) queries (Lucene-like index)
- similarity queries (via similarity indexes)
- combined similarity queries (*late fusion*)
- $K$-NN query with attribute filtering
- distributed re-ranking query answer

- efficient management of multiple data collections
  \[ \mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \cdots \cup \mathcal{X}_s \]
- core key-value store is well horizontally scalable
Specific System: Large-scale Image Management

100M objects from the **CoPhIR** dataset (benchmark):

```json
{
    "ID": "002561195",
    "title": "My wife & daughter on Gold Coast beach",
    "tags": "summer, beach, ocean, sun, sand, Australia",
    "mpeg7_scalable_color": "25 36 0 127 69...",
    "mpeg7_color_layout": "25 41 53 20; 32; -16...",
    "mpeg7_color_structure": "25 41 53 20; 32;...",
    "mpeg7_edge_histogram": "5 1 2 3 7 7 3 6...",
    "mpeg7_homogeneous_texture": "232 201 198 180 201...",
    "GPS_coordinates": "45.50382, -73.59921",
    "flickr_user": "david_novak"
}
```
System Schema

- **Infinispan (Ispn) ID-object store**

- PPP-Codes index $I_{mpeg7}$
- PPP-Codes index $I_{GPS}$
- Lucene $I_{tags,title}$
- B+-tree $I_{flickr_user}$

**worker**

- Ispn node
- index

**worker**

- Ispn node
- index

**worker**

- Ispn node
- index

**worker**

- Ispn node
- index
Specific System: Demo

20M objects of this type:

```json
{  "ID": "002561195",
   "title": "My wife & daughter on Gold Coast beach",
   "keywords": "summer, beach, ocean, sun, sand, Australia",
   "DNN_visual_descriptor": [5.431, 0.0042, 0.0, 0.97,... ] }
```
Conclusions

We have proposed and alfa-tested system architecture that

- provides large-scale similarity search
- ...on a broad family of data + similarity measures
- is distributed and horizontally scalable
Conclusions

We have proposed and alfa-tested system architecture that
- provides large-scale similarity search
- ...on a broad family of data + similarity measures
- is distributed and horizontally scalable
- can manage multi-field data:
  - attribute, keywords, several similarity modalities
  - many variants of multi-modal search queries
Conclusions

We have proposed and alfa-tested system architecture that

- provides large-scale similarity search
- ...on a broad family of data + similarity measures
- is distributed and horizontally scalable
- can manage multi-field data:
  - attribute, keywords, several similarity modalities
  - many variants of multi-modal search queries

Challenges:

- full implementation and thorough testing
- the similarity index can be bottleneck $\implies$ distribute it