

Vector search #1 – Introduction to vector search

Planning of the classes

Meet the teachers



Edo Liberty, Pinecone

(Director of AWS AI Labs, Sr Director
Yahoo Research)

- Numerical Linear Algebra
- Streaming Algorithms
- Machine Learning Theory
- Randomized Algorithms



Matthijs Douze, Meta

(10 years at INRIA)

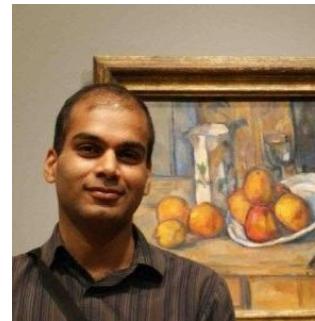
- Computer vision
- Similarity search
- Unsupervised learning
- Real-time 3D reconstruction



Nataly Brokhim, Princeton

(Researcher at Google AI)

- Machine Learning Theory
- Regret Minimization
- Boosting



Harsha Simhadri, Microsoft

(Senior Principal Researcher, MSR)

- web-scale approximate nearest-neighbor search
- new ML operators and architectures

Classes by weeks

- 9/8 - Class 1 - Introduction to Vector Search [Matthijs + Edo + Nataly]
 - 9/15 - Class 2 - Text embeddings [Matthijs]
 - 9/22 - Class 3 - Image embeddings [Matthijs]
 - 9/29 - Class 4 - Low Dimensional Vector Search [Edo]
 - 10/6 - Class 5 - Dimensionality Reduction [Edo]
 - 10/13 - **No Class - Midterm Examination Week**
 - 10/20 - **No Class - Fall Recess**
 - 10/27 - Class 6 - Approximate Nearest Neighbor Search [Edo]
 - 11/3 - Class 7 - Clustering [Edo]
 - 11/10 - Class 8 - Quantization for lossy vector compression [Matthijs]
 - 11/17 - Class 9 - Graph based indexes [Guest lecturer + Edo]
 - 11/24 - No Class - **Thanksgiving Recess**
 - 12/1 - Class 10 - Student project and paper presentations [Edo + Nataly]
-
- The diagram illustrates the distribution of class weeks into three distinct categories, each represented by a vertical brace (bracey) grouping specific weeks:
- Applied ML/AI Centric:** Weeks 1 through 5, which include Class 1 (Introduction to Vector Search), Class 2 (Text embeddings), Class 3 (Image embeddings), Class 4 (Low Dimensional Vector Search), and Class 5 (Dimensionality Reduction). These are grouped by a brace on the right side.
 - Whiteboard Oriented:** Weeks 6 through 8, which include Class 6 (Approximate Nearest Neighbor Search), Class 7 (Clustering), and Class 8 (Quantization for lossy vector compression). These are grouped by a brace on the right side.
 - Experimental/Applied CS:** Weeks 9 through 10, which include Class 9 (Graph based indexes) and Class 10 (Student project and paper presentations). These are grouped by a brace on the right side.

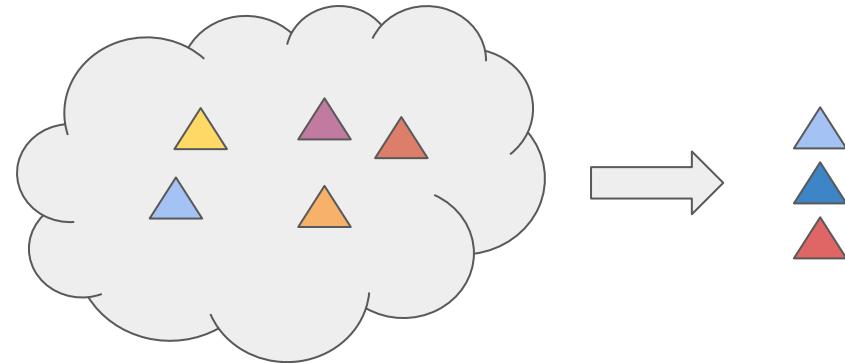
Grading and Project Information

Class grading is based on a final project a) write up and b) in-class presentation

- Project Administrator: Nataly
- Projects can be done individually, in teams of two or at most three students.
- Expect to spend a few hours over the semester on the project proposal
- All project proposals should be approved before the thanksgiving break
- Project can be in three different flavors
 - **Theory/Research:** Explore a research problem, conduct literature survey, and propose a potential idea for improvement.
 - **Data Science/AI:** Build an interesting vector search application using Pinecone, explain what value it brings, and what insights you gained.
 - **Engineering/HPC:** Adapt or add to FAISS, explain your improvements, show results.
- Expect to spent 3-5 full days on the project itself (on par with preparing for a final)
- Project write up submission is due 12/1
- In class project presentation, each student presents their work in five minutes.

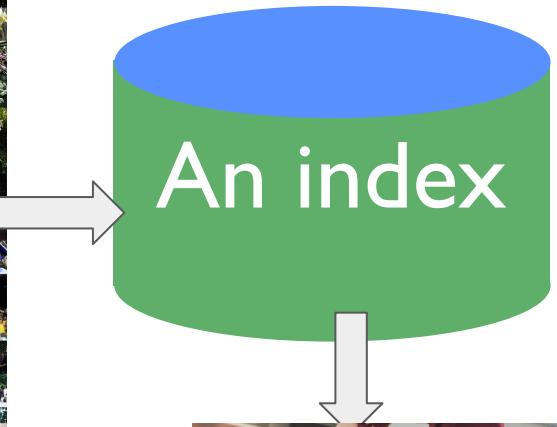
Information retrieval

Context



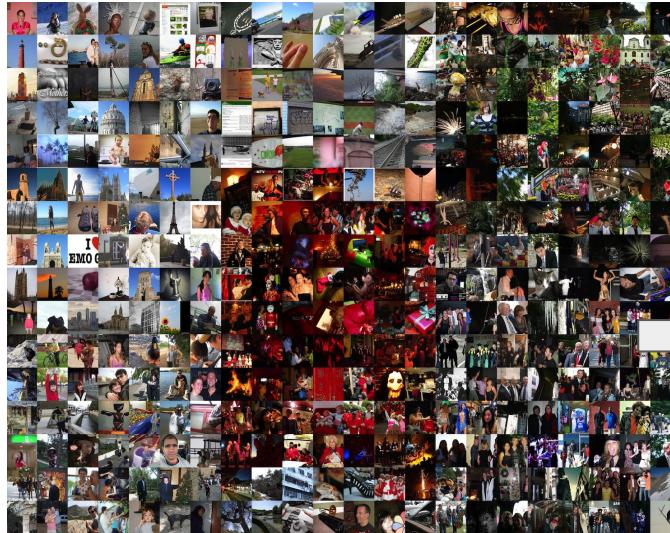
Classification

- A database
- Query concept:
“Bike”
- Example: Google/bing/Yandex/Baidu/Naver image search



Similarity search

- A database



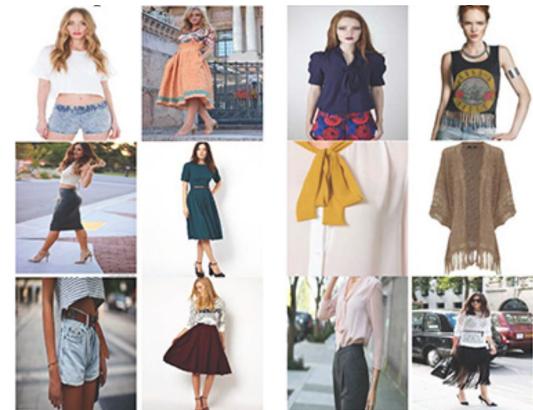
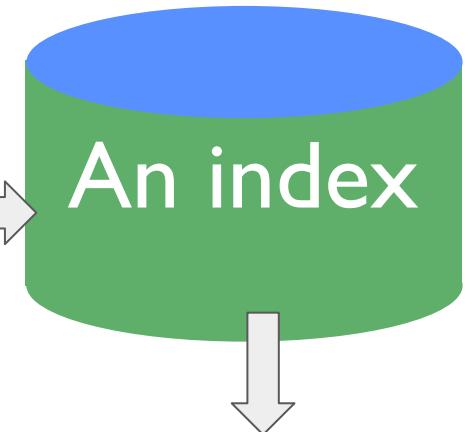
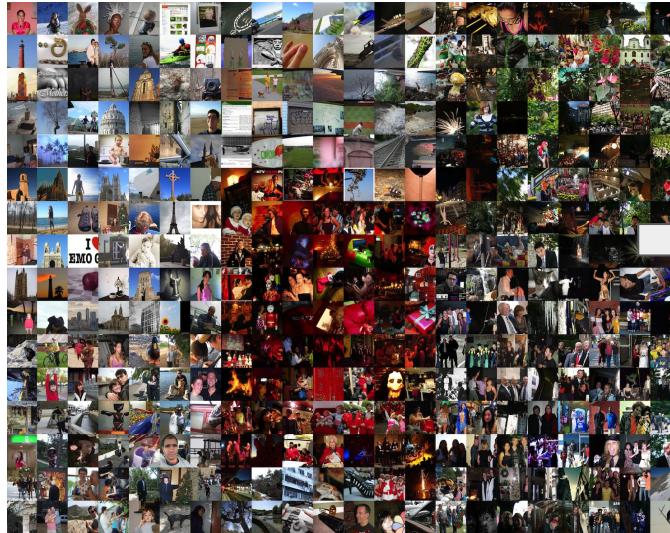
- A query



- “Reverse image search” engine (Google/Bing, Tineye, ...)

Recommendation

- A database
- Query = the user
Based on past behavior

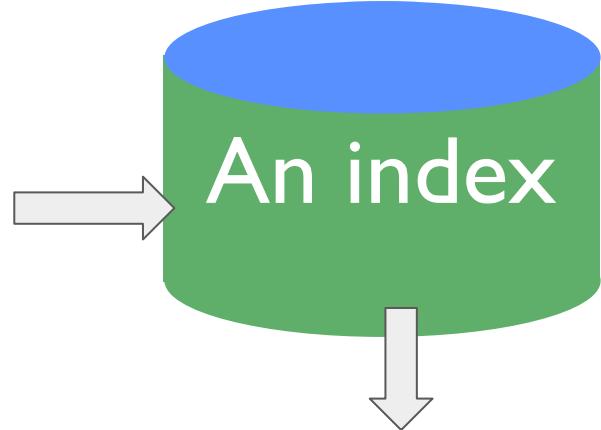


Question answering

- A database
- Query = “Who was the president of Pakistan in 2006?”



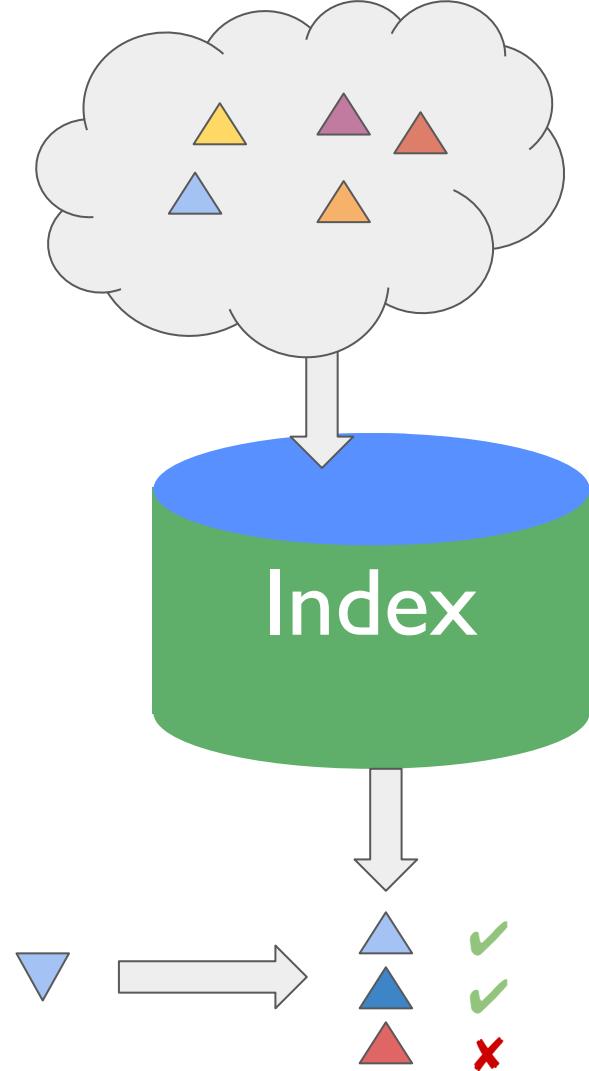
WIKIPEDIA
The Free Encyclopedia



Pervez Musharraf

Formally: information retrieval

- A database
 - Collection of items
 - Items = text / images / audio / video / products / etc.
 - Indexed, updated over a long time period
- A query
 - Item = what triggers the query
 - May be of the same type as the database items or not
- Result(s)
 - Subset of database items that are relevant to the query
 - Typically needs to be provided quickly (interactive time)
 - Typically ordered
 - Correct or incorrect result
- We need the algorithm!



Data volumes and scale: for one person

- Data sizes on one's computer
- Text
 - A few 100s of documents on a computer
 - 40 e-mails per day
- Images
 - Average smartphone contains 2000 pictures
- Connections
 - Average of 388 friends on Facebook
- → Search scale is ~100 - 10k

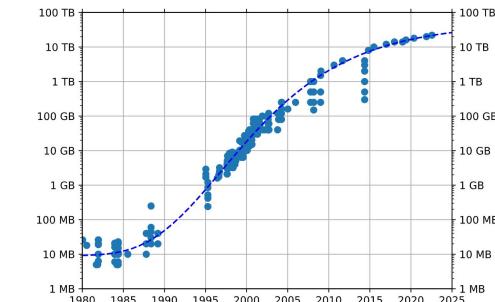
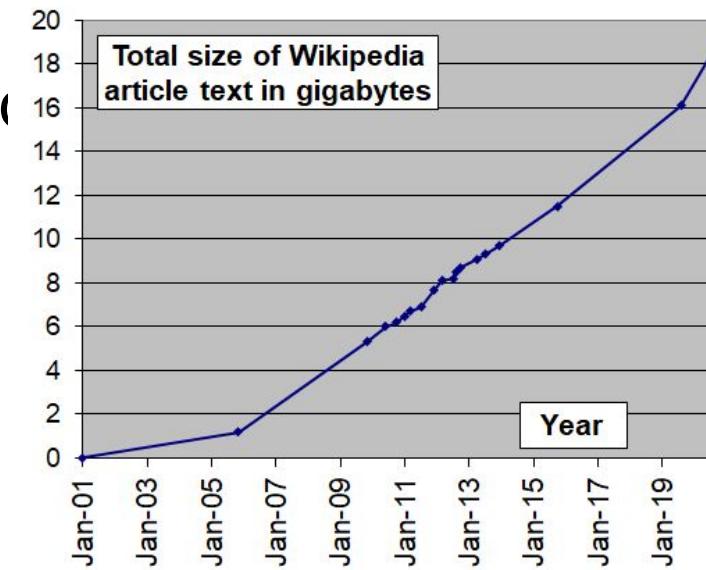


[source: <https://www.lightstalking.com/photo-statistics/>]

[source: <https://www.lightstalking.com/photo-statistics/>
https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia
https://upload.wikimedia.org/wikipedia/commons/9/90/Hard_drive_capacity_over_time.svg]

Data volumes and scale: for the world

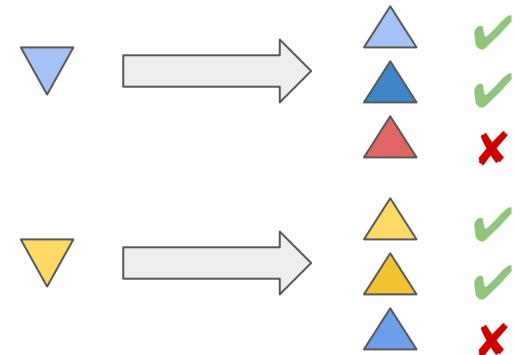
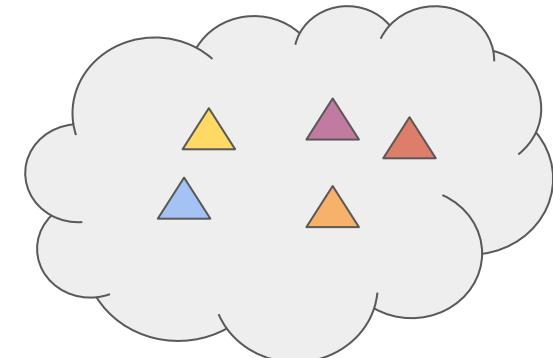
- Text
 - Size of wikipedia: 54M articles
 - Visible web: 15B pages (in 2015)
- Images
 - Average smartphone contains 2000 pictures
 - 3.8B smartphone users
 - 1B+ images uploaded to FB per day
 - 4T photos stored on Google Photos, (+4B per day)
- Video
 - 500h of video uploaded to Youtube per minute (2019)
- Specialized applications
 - CERN has 70 PB of storage for particle collision experiments
- Growth is exponential
 - More users \times times more content per user
 - Grows faster than hard drive capacity
 - Need for more data centers
- → search scale is 1B to 10T items



Evaluation: datasets and metrics

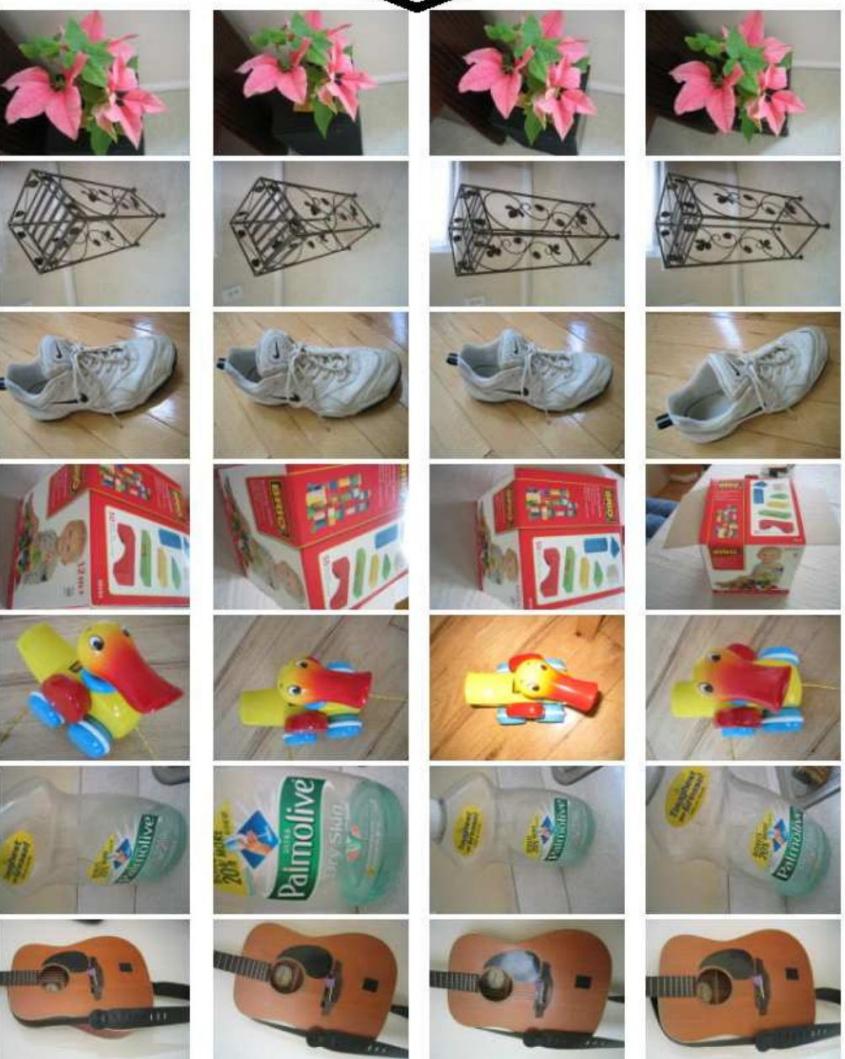
A dataset

- Fixed database
- Fixed queries
- Known results
 - What the human considers a correct result
 - “Ground truth”



Object recognition on images

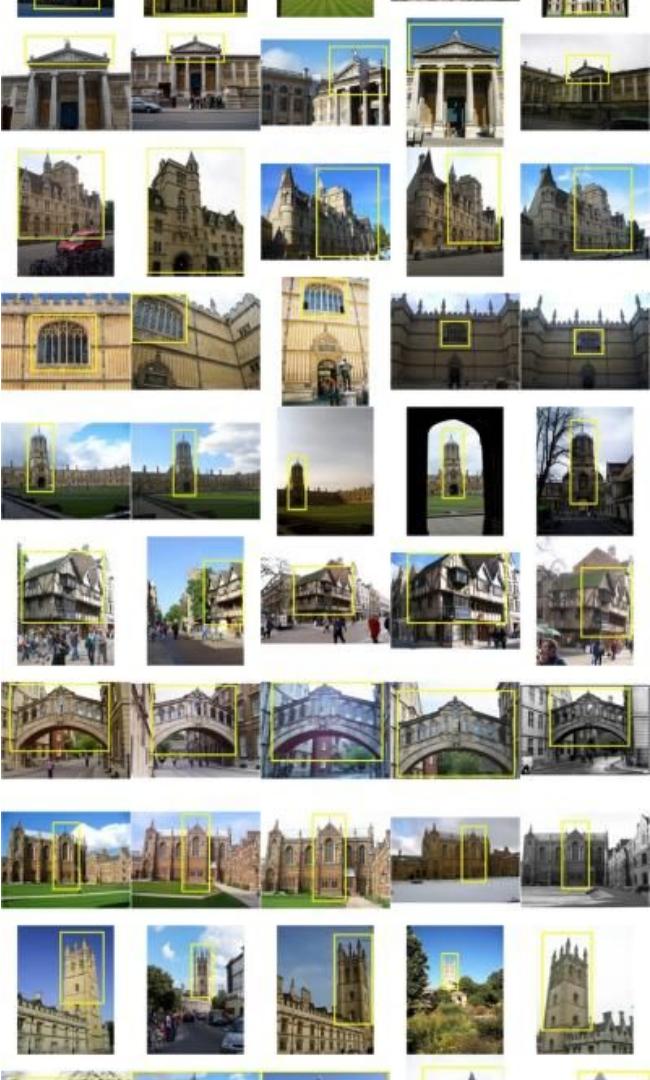
- UKBench
- 10200 images
 - Groups of 4
- Setup:
 - Each image is used as a query in turn
 - GT = other images of the same group



[Nister, Stewenius, Scalable recognition with a vocabulary tree, CVPR'16.]

Building recognition

- Oxford building dataset
- 5000 database images
- 55 query images + bounding box
 - From 12 buildings in Oxford
- Ground-truth: same building



Q&A from document corpus

- TriviaQA
- Database: 662k documents
- Queries:
 - 95k Q&A pairs
- Ground truth:
 - Exact same answer
 - Correct evidence (on avg. 5 docs / query contain the answer)
- Retrieval + extract answer

Question: The Dodecanese Campaign of WWII that was an attempt by the Allied forces to capture islands in the Aegean Sea was the inspiration for which acclaimed 1961 commando film?

Answer: The Guns of Navarone

Excerpt: The Dodecanese Campaign of World War II was an attempt by Allied forces to capture the Italian-held Dodecanese islands in the Aegean Sea following the surrender of Italy in September 1943, and use them as bases against the German-controlled Balkans. The failed campaign, and in particular the Battle of Leros, inspired the 1957 novel **The Guns of Navarone** and the successful 1961 movie of the same name.

Question: American Callan Pinckney's eponymously named system became a best-selling (1980s-2000s) book/video franchise in what genre?

Answer: Fitness

Excerpt: Callan Pinckney was an American fitness professional. She achieved unprecedented success with her Callanetics exercises. Her 9 books all became international best-sellers and the video series that followed went on to sell over 6 million copies. Pinckney's first video release "Callanetics: 10 Years Younger In 10 Hours" outsold every other **fitness** video in the US.

Recommendation

- Netflix prize dataset
 - Can also be considered as a classification dataset...
- Database (aka items): 17.7k movies
- Queries (aka users): 480k users
- Training data
 - Number of stars given by some (user, item) pair
 - Very sparse (we know what the user thinks only of a tiny subset of movies)
- Ground-truth:
 - Held-out ratings

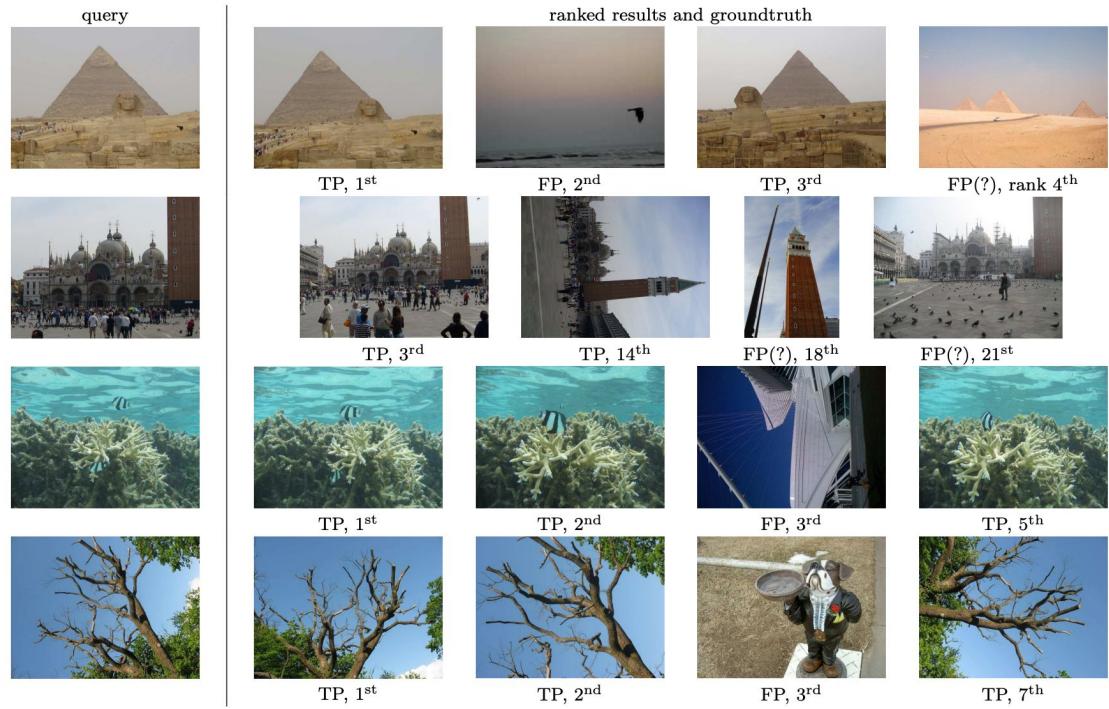
Training data:

quadruplet of the form

```
<user, movie, date of grade, grade>
```

Results for a image dataset

- Holidays dataset
- True / false positive



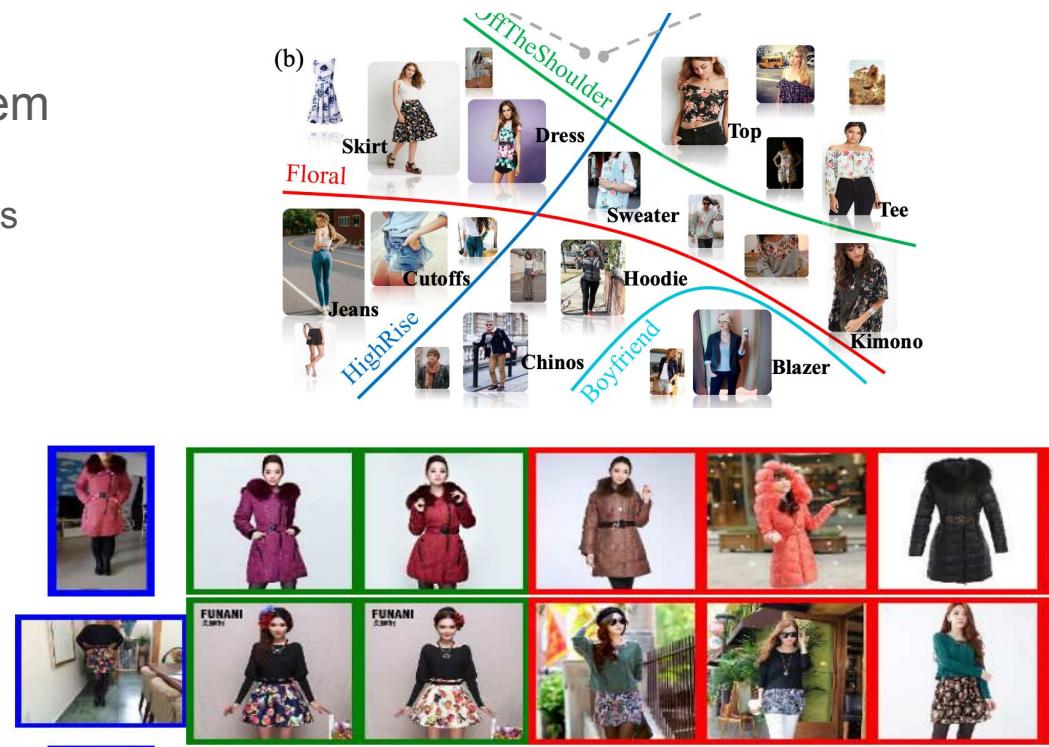
[Hamming embedding and weak geometric consistency for large scale image search, Jegou, Douze, Schmid, ECCV'08]

Example: fashion dataset

- DeepFashion
- Long-standing and hard problem
 - Expert knowledge
 - Very important for commerce sites
- Correct / incorrect results

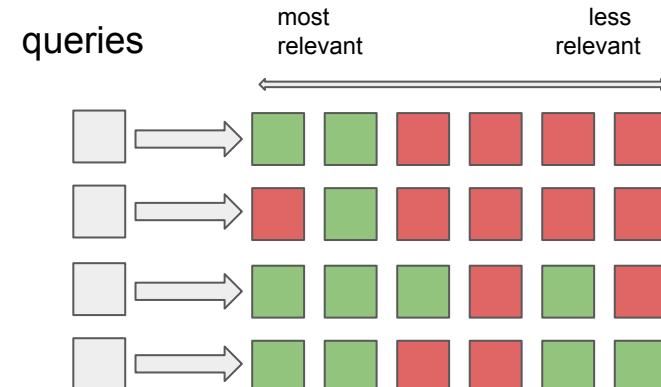
[DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations, Ziwei Liu, CVPR'16]

[Cross-domain fashion image retrieval
Bojana Gajic, Ramon Baldrich, CVPR'18 WS]



Information retrieval metrics

- Based on a ranked result list
 - For all queries
 - Ranks for the whole database or just a subset
 - Results are assessed as correct or incorrect
 - “True / False” positive (TP / FP)

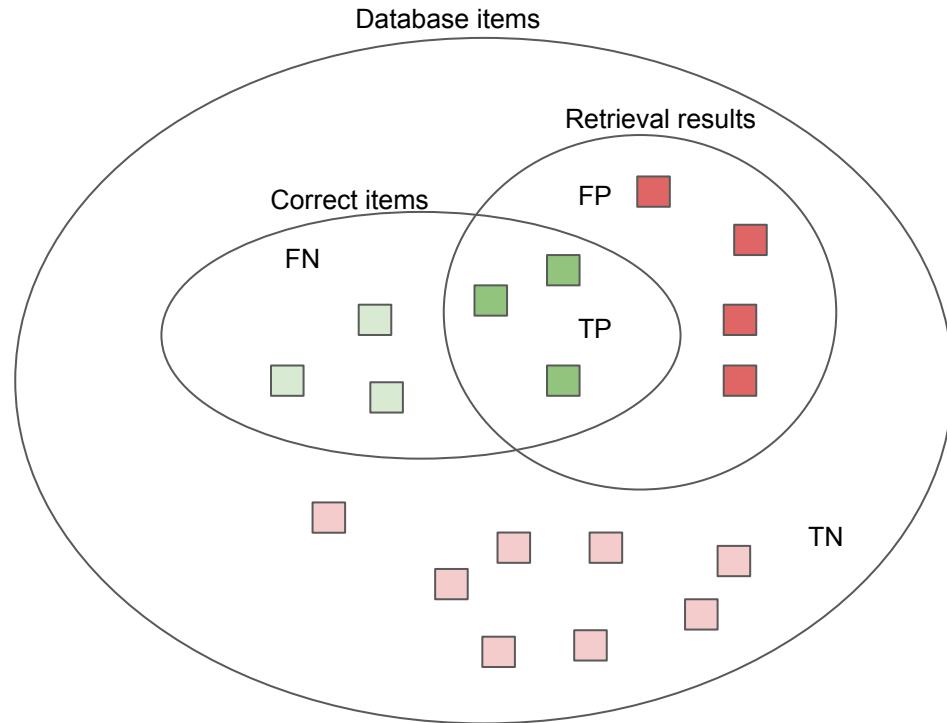


Metrics: precision / recall

- Set of results
 - Crop result list at certain rank

$$\text{Precision} = \frac{\#TP}{\#\text{results}} = \frac{\#TP}{\#TP + \#FP}$$

$$\text{Recall} = \frac{\#TP}{\#\text{correct items}}$$



Recall @ rank

- Fix the rank (=size of result list) and measure recall
 - Precision is a monotonous function of recall

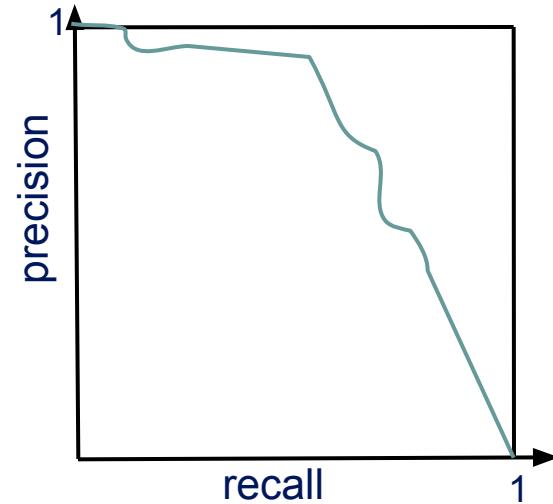
$$\text{Precision} = \frac{\# \text{TP}}{\# \text{results}} = \text{recall} \times \frac{\#\text{correct items}}{\text{rank}}$$



- Metric for a set of queries
 - Average over per-query results
- Special case: accuracy
 - single positive (think classification problem)
 - recall @ 1 = accuracy
- Used in ukbench
 - Rank = 4
 - Normal: number of correct results per query...

Precision-recall plot

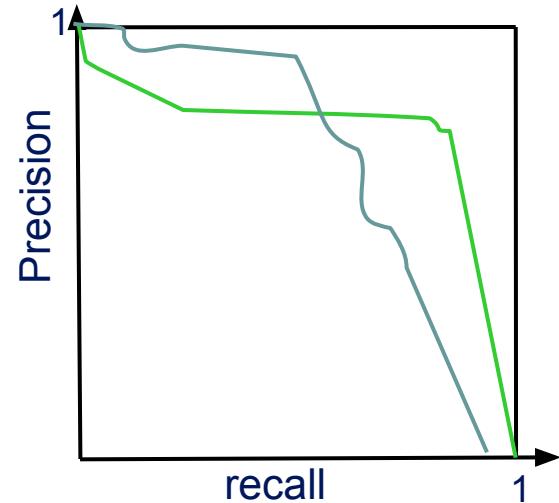
- Vary the rank in result list
 - Crop result list size
- Measure the precision and recall for all ranks
- For random ranking of database
 - Expected precision is constant = correct / (database size)
- For reasonable retrieval systems
 - Precision decreases
 - Recall increases
 - Start at (0, 1)



Metric for precision-recall plots

- Summary of performance
- High-precision:
 - web search results...
 - Few people look beyond 5 results
- High-recall:
 - Exhaustive identification of harmful content
 - Expensive / manual verification

Compare the green and blue methods

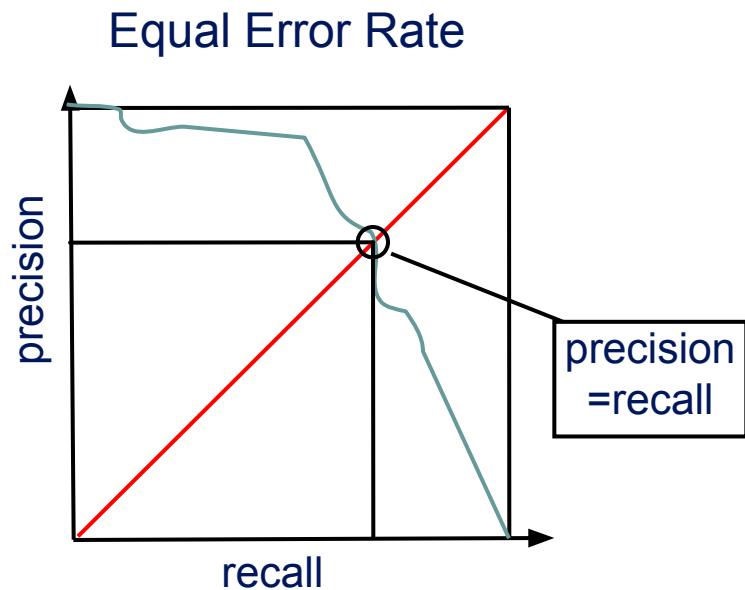


Equal error rate

- Classical metric for biometric systems
 - Face / fingerprint / iris recognition
- False Acceptance Rate = False Rejection Rate

$$\text{FAR} = \frac{\#\text{FP}}{\#\text{results}} = 1 - \frac{\#\text{TP}}{\#\text{results}} = 1 - \text{precision}$$

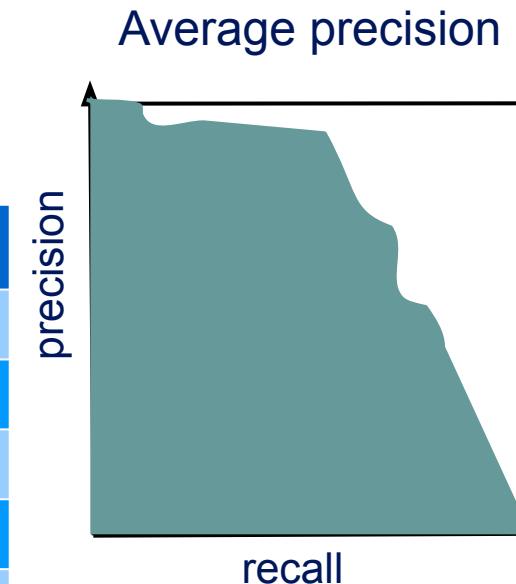
$$\text{FRR} = \frac{\#\text{FN}}{\#\text{correct}} = \frac{\#\text{correct} - \#\text{TP}}{\#\text{correct}} = 1 - \text{recall}$$



Average precision

- Computed as area under the P-R curve
 - summing up trapezoid areas
 - Other variant: rectangles
- Used for image retrieval

rank	#TP	recall	precision
NA		0	1
1	0		
2	1		
3	2		
4	2		
5	3		
6	3		
7	4		



Exercise: compute average precision

- Given the following query and sorted result list, compute the AP with trapezoids



1



2



3



4



5



6



7



8



9

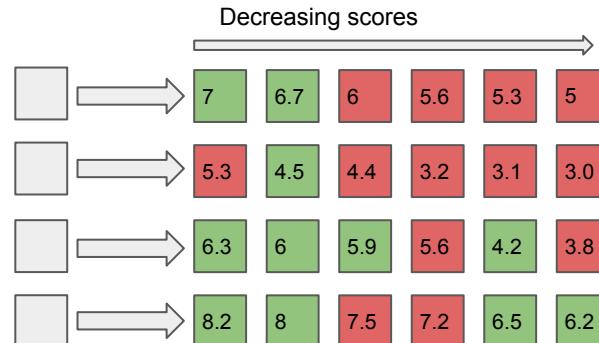


10



Per-query AP

- Per query AP:
 - Relevant when each query is consumed in isolation
 - User queries
 - Objectives: Minimize latency, maximize few first results, unbounded result list
- Usually based on scores
 - Ordering = sorting scores
- For several queries
 - Average over queries
 - all queries are equally important
 - Mean AP (mAP)

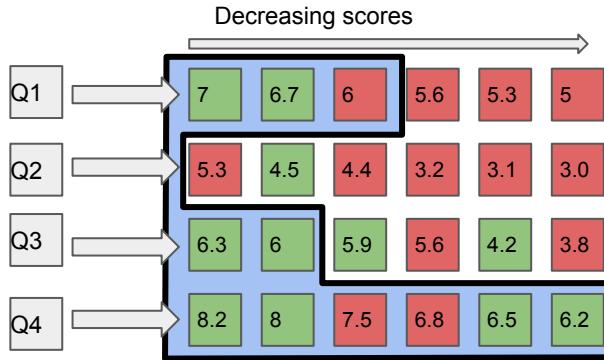


Global AP

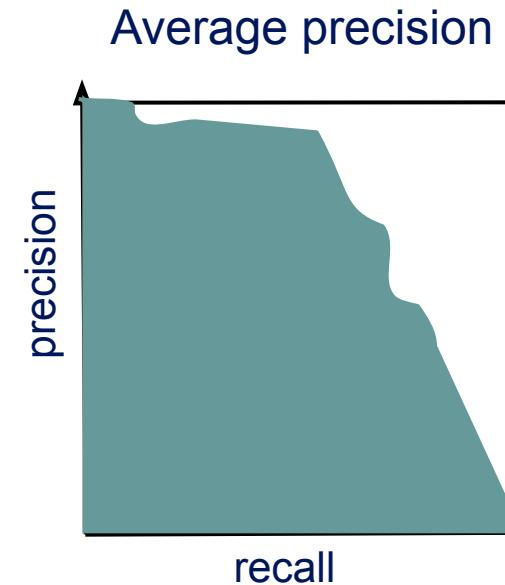
- Relevant with a batch of queries
- Queries are not all as important
 - some queries don't have any correct result (precision undefined)
 - some have many
- Use a global threshold on scores
 - Pairwise comparison
- How many correct (query item, db item) ?



Global AP



rank	score	query	TP/FP
1	8.2	Q4	TP
2	8	Q4	TP
3			
4			
5			
6			
7			



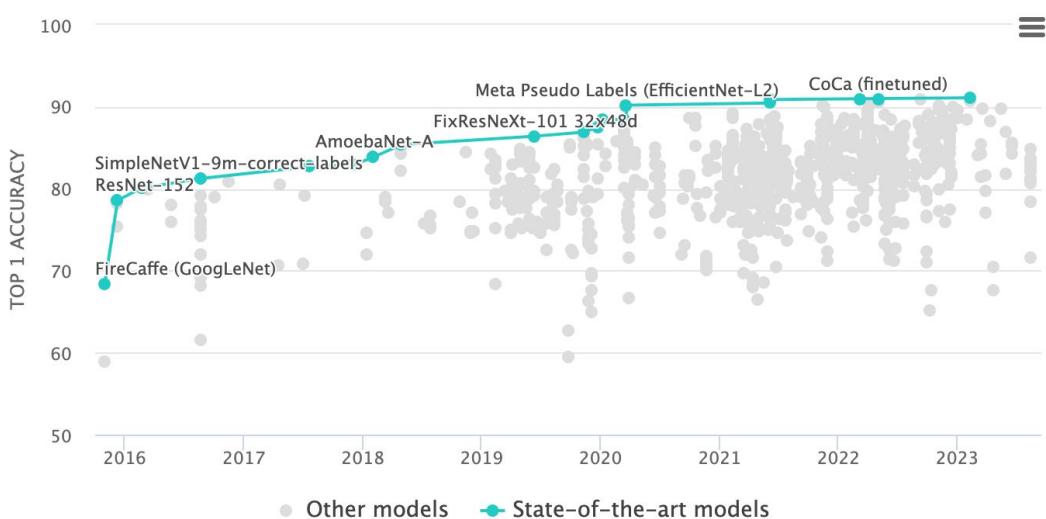
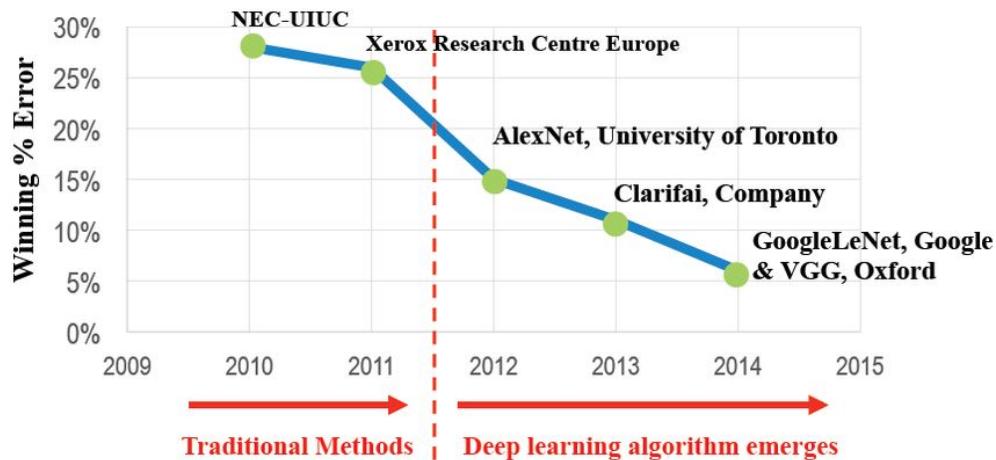
Evaluation: benchmarks

Benchmark

- A dataset
- Ground-truth
- A metric
- Baseline
 - Reference result with a simple algorithm
- Standardized evaluation
 - For papers
 - For companies

Career of a benchmark

- Typical example
 - Imagenet classification
- Changed metric
 - Top-5 error → top-1 accuracy
- Pre- and post-deep learn^{~~~}
- Migration to Paperwithcode
- Saturation
- Questions about the Benchmark
 - Annotation errors
 - Overfitting



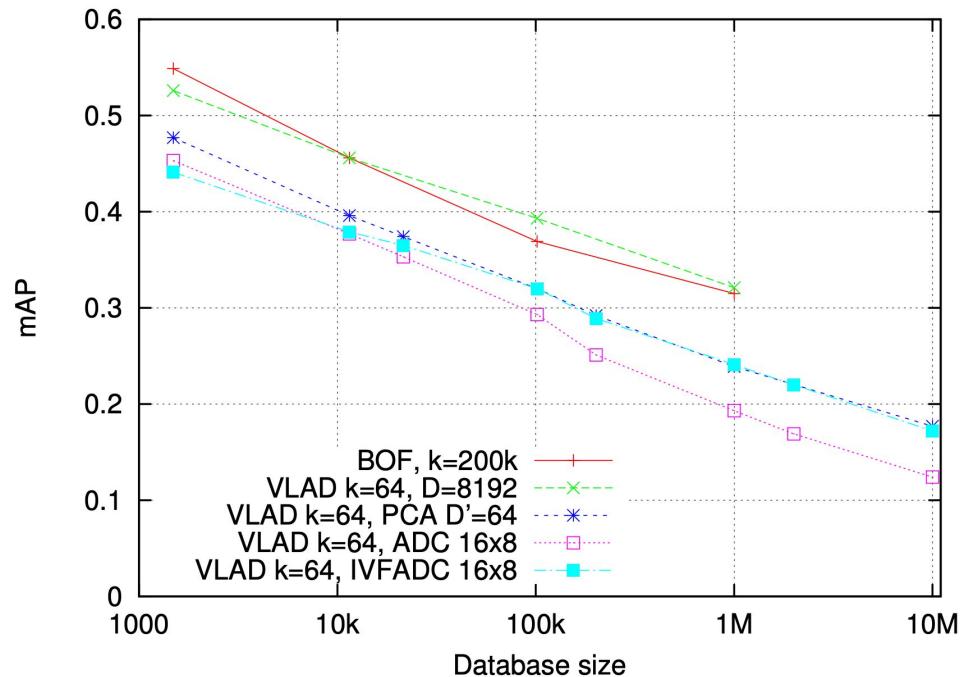
Benchmarks for image retrieval

- Oxford building dataset
- ROxford (Hard)



It is hard to search in a larger dataset

- More false positives in results
 - Insert in from to TPs
- But not **too** hard
 - Random result list would be linearly decreasing
- The Holidays dataset



Information retrieval with regular databases

Re-use existing infrastructure

- What do we have?
 - Filesystems,
 - Database systems
 - Text / keyword-based search engines
- Map items into a compatible representation
 - How far can we get with that?

Classical databases

- Set of data that is put in relation
- A DBMS (DataBase Management System) manages the data entries
 - Oracle, DB2, Sybase, PostgreSQL, Mysql...
 - Key-value store
- Useful for well structured data
 - Tabular records
 - Easy to manipulate, search etc.
- Typically queries via SQL queries

Classical databases

- Structured data
 - Tables with records (rows)
- Query language: SQL

```
SELECT id FROM table WHERE date = "2022-04-05"
```

```
SELECT id FROM table WHERE keywords LIKE "%bike%"
```

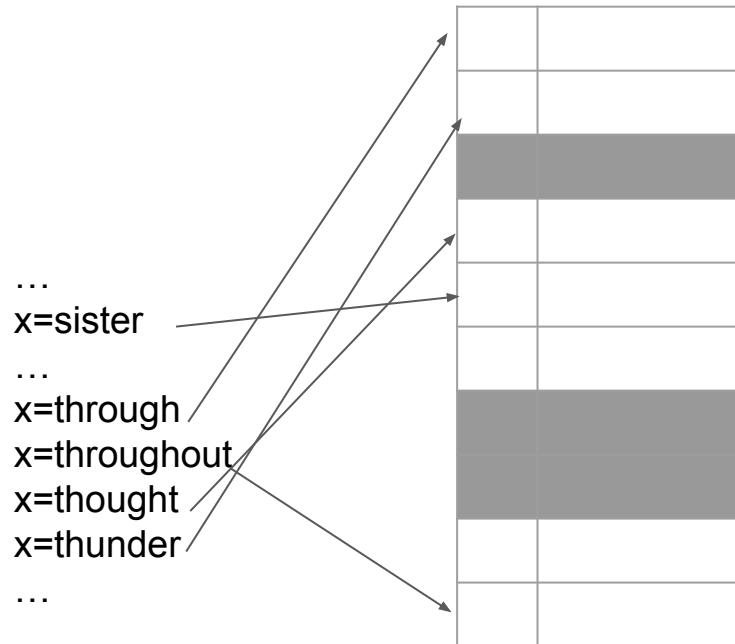
- Efficient search

Classical databases – point access

- Based on hash tables
- Hash function: maps input array (characters...) to an integer
 - Good hash function is as discontinuous as possible
 - Something like (with large prime numbers!)

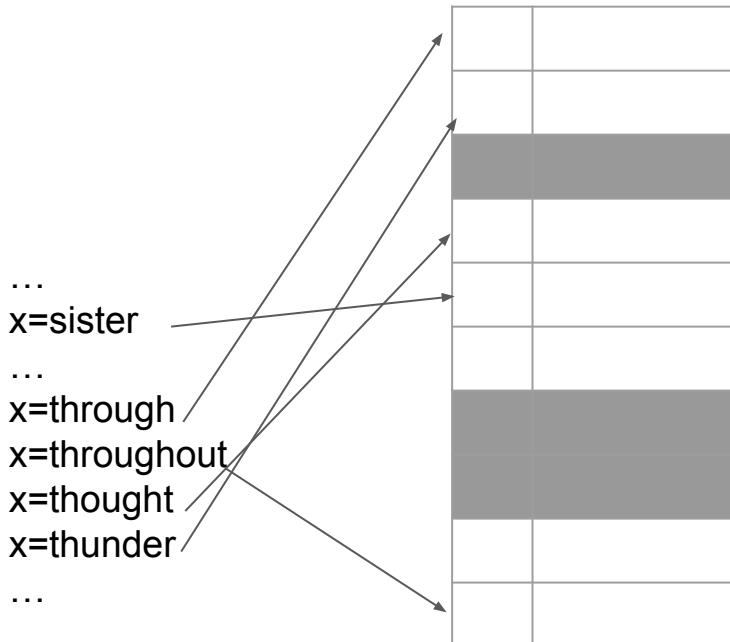
$$k(x) = ((\sum_i r_i x_i) \bmod P) \bmod m$$

- Entry stored at $k(x)$
- A fraction of the entries is empty
 - ~30%
 - over allocation of memory
 - Collisions



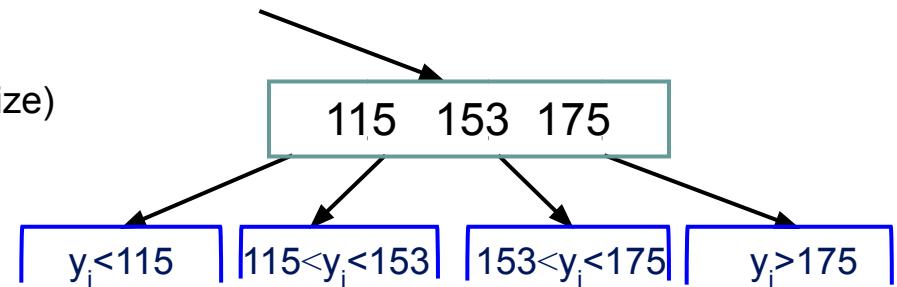
Classical databases – point access

- A fraction of the entries is empty
 - ~30%
 - over allocation of memory
 - Managing collisions...
- Search
 - Query x
 - Start search at $q(x)$
 - Walk through collisions
- For keys known exactly
 - By design of the hash function
 - Small change in the key → large jump in the table



Classical databases – range access

- Based on tree search
- Example: B+ trees
- Internal nodes with separating values
 - Fixed capacity B (determined by disk block size)
 - Not necessarily full



B+ tree: leaves

- Contain database records
- Linking to next entry



B+ tree: leaves

- Invariant:
 - Always 50 to 100% full
 - When too many elements in a node: split
 - When two neighboring nodes are < 50% full on average: merge
- Search single value:
 - Route to relevant leaf
 - Complexity logarithmic in database size
- Serarching an interval [a, b]
 - Route to relevant leaf
 - Follow links until > b

Exercise: which nodes are visited for this search

- TODO example tree

Classical databases – the inverted index

- For text-based indexing
 - Document = sequence of words
 - Direct index = maps document id to list of words in it
- Inverted index
 - Map word → list of documents that contain the word

Classical databases – the inverted index

I ride a bike in Paris.

Paris has more and more bike lanes.

Bike rides in Amsterdam

- For text-based indexing
 - Document = sequence of words
 - Direct index = maps document id to list of words in it
- Inverted index
 - Map word → list of documents that contain the word
 - “Inverted list”
 - Word stemming (remove plural)
 - Stop words (common words)

a	1
Amsterdam	3
and	2
bike	1, 2, 3
has	2
in	1, 3
lane	2
more	2
Paris	1, 2
ride	1, 3

The inverted index: search

- Query = “bike Paris”



Amsterdam	3
bike	1, 2, 3
has	2
lane	2
more	2
Paris	1, 2
ride	1,3



Doc 1,
Doc 2

- Query cost
 - Depends on size of inverted lists
 - Intersection of sorted lists
 - Worst-case linear in total size of lists

Metadata

Metadata = auxiliary data

- Not the main content
 - Data associated with it
- Easier to manipulate
 - More semantically high level
 - Made for indexing

Images: EXIF metadata

- Added by camera
 - Or post-processing software
- JPEG or HEIC

General	Exif	GPS	TIFF
Altitude	184,47 m (605,2 ft)		
Altitude Reference	above sea level		
Destination Bearing	21,327		
Destination Bearing Reference	True direction		
Horizontal Positioning Error	17,226		
Image Direction	21,327		
Image Direction Reference	True north		
Latitude	44° 43' 38,772" N		
Longitude	5° 1' 17,07" E		
Speed	0		
Speed Reference	Kilometers per hour		

General	Exif	GPS	TIFF
Aperture Value	2		
Brightness Value	1,082		
Color Space	Uncalibrated		
Components Configuration	1, 2, 3, 0		
CompositelImage	2		
Date Time Digitized	9 Aug 2023 at 21:15:18		
Date Time Original	9 Aug 2023 at 21:15:18		
Digital Zoom Ratio	1,217		
Exif Version	2.3.2		
Exposure Bias Value	0		
Exposure Mode	Auto exposure		
Exposure Program	Normal program		
Exposure Time	1/50		
Flash	Off, did not fire		
FlashPix Version	1.0		
FNumber	2		
Focal Length	6		
Focal Length In 35mm Film	63		
Photographic Sensitivity	400		
Lens Make	Apple		
Lens Model	iPhone 11 Pro back trip...		
Lens Specification	1,54, 6, 1,8, 2,4		
Metering Mode	Pattern		
Time Zone for Modification	+02:00		
Time Zone for Digitized	+02:00		
Time Zone for Original Date	+02:00		
Pixel X Dimension	4032		
Pixel Y Dimension	3024		
Scene Capture Type	Standard		
Scene Type	A directly photographed scene		
Sensing Method	One-chip color area sensor		
Shutter Speed Value	1/50		
Subject Area	2022, 1515, 2329, 1395		
Sub-second Time Digitized	527		
Sub-second Time Original	527		
White Balance	Auto white balance		



This is the village I live in!

AAC metadata

- Contains info like author, copyright, etc
 - https://wiki.multimedia.cx/index.php/FFmpeg_Metadata

"title"	Name
"author"	Artist
"album_artist"	Album Artist
"album"	Album
"grouping"	Grouping
"composer"	Composer
"year"	Year
"track"	Track Number
"comment"	Comments
"genre"	Genre

0 to 19	
Number	Genre
00	Blues
01	Classic rock
02	Country
03	Dance
04	Disco
05	Funk
06	Grunge
07	Hip-hop
08	Jazz
09	Metal
10	New age
11	Oldies
12	Other
13	Pop
14	Rhythm and blues

Text-based retrieval

- Text is easy to search
 - Inverted file
- Beyond metadata
 - Let's add a description of the content and use text
 - keyword based retrieval
- What image search on the web is based on
 - “ALT text”

Content annotations

Making content easy to search is not obvious (Shutterstock)



10 Tips on Creating Great Keywords and Titles for Your Images

By [Jordyn Giesbrecht](#) on November 2, 2018



Jordyn Giesbrecht

Jordyn tells stories of people, spaces, and places through her

Accurate keywords and titles make all the difference when you offer your images online. Here's how to amp up your descriptions and get your images in front of the right audience.

Share this post



Annotation ambiguity

- “Maria and Bella during spring break”
- “My fitness program for this summer”
- “Big Sur beaches”
- “Woman and dog running”

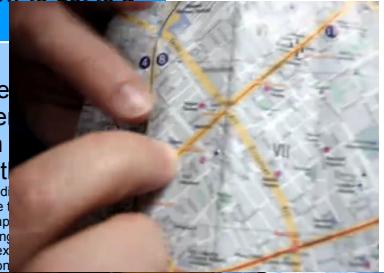


Example: annotator instructions (Pascal VOC)



What to label	All objects of the defined categories, unless: you are unsure what the object is. the object is very small (at your discretion). less than 10-20% of the object is visible. If this is not possible because too many objects, mark image as bad.
Viewpoint	Record the viewpoint of the 'bulk' of the object e.g. the body rather than the head. Allow viewpoints within 10-20 degrees. If ambiguous, leave as 'Unspecified'. Unusually rotated objects e.g. upside-down people should be left as 'Unspecified'.
Bounding box	Mark the bounding box of the visible area of the object (<i>not</i> the estimated total extent of the object). Bounding box should contain all visible pixels, except where the bounding box would have to be made excessively large to include a few additional pixels (<5%) e.g. a car aerial.
Truncation	If more than 15-20% of the object lies outside the bounding box mark as Truncated. The flag indicates that the bounding box does not cover the total extent of the object.
Occlusion	If more than 5% of the object is occluded within the bounding box, mark as Occluded. The flag indicates that the object is not totally visible within the bounding box.

Example: annotator instructions (Trecvid 2012)

Event Name	Giving directions to a location	
Definition:	One or more people give directions to one or more other people, either in person or over the phone, by explaining verbally and/or with gestures how to get to a particular location.	
Explication:	<p>People may give directions in response to being asked for them, or they may give them without being asked as a part of a normal conversation if the conversation is about a location (e.g. telling a friend how to get to a new restaurant that just opened or a place they may be interested in eating at). People may ask for directions from strangers they see on the street, or call an information service or someone they know to ask for directions over the phone. Or, the person giving directions and the one getting directions may be traveling together, and one person is serving as the navigator while the other(s) follow the directions. This commonly happens when the person getting the directions is driving and the person giving directions is reading them from a map, printout, or smart phone. Note that the person giving directions is not relevant for this event, and that the person giving directions must be visible. If people are visiting a new city or country, they will often have a map or a guidebook to reference when asking for directions from a person on the street. Depending on whether the people asking for directions are/will be walking, driving, taking a bus, or cycling, the directions given may reference city blocks, highways, or subway/train/bus routes. People giving directions often gesture along with their directions, for example, pointing to the right and turning their head to the right as they explain to go right, and this could be done even if giving them on the phone. People giving directions on the phone may stay on the line with the person they are helping while they reach their destination, giving them the directions step by step in real time.</p>	
scene:	outdoors, indoors	
objects/people:	map, driver, car, bicycle, subway, train, bus, pedestrian, car passenger, portable telephone	
activities:	gestures indicating directions (e.g. pointing or extending arm straight/to right or left of speaker), person pointing out location on a map, head movement indicating direction	
audio:	narration of directions	

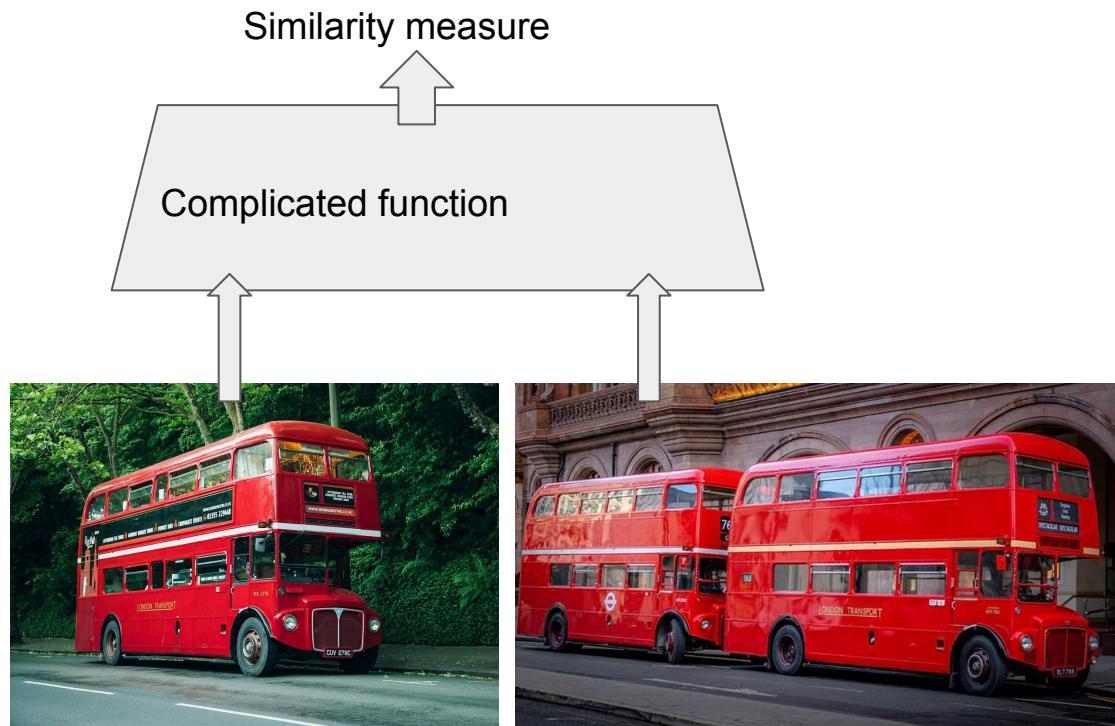
Limitations of text-based retrieval

- Requires user intervention
 - People don't bother writing keywords for their data
- Difficult to define (ambiguous)
- Does not scale
 - More content created per user
 - People don't spend more time annotating...
- ⇒ A dead end!
- Need processing from the content itself

Similarity measures

Comparing items

- Query item
- Database item
- Distance function
 - Or similarity
- Building a comparison function
 - Hand-crafted
 - Based on machine learning:
matching and non-matching
examples
 - Subject of 2 lessons



Distance measures

- A distance measure is a function $E \times E \rightarrow \mathbb{R}^+$
 - (P1) separation: $d(x,y) = 0 \Leftrightarrow x = y$
 - (P2) symmetry: $d(x,y) = d(y,x)$
 - (P3) triangular inequality: $d(x,z) \leq d(x,y) + d(y,z)$
- Relaxation: dissimilarity measure
 - (P1') $x = y \Rightarrow d(x, y) = 0$
- Similarity is the opposite of dissimilarity

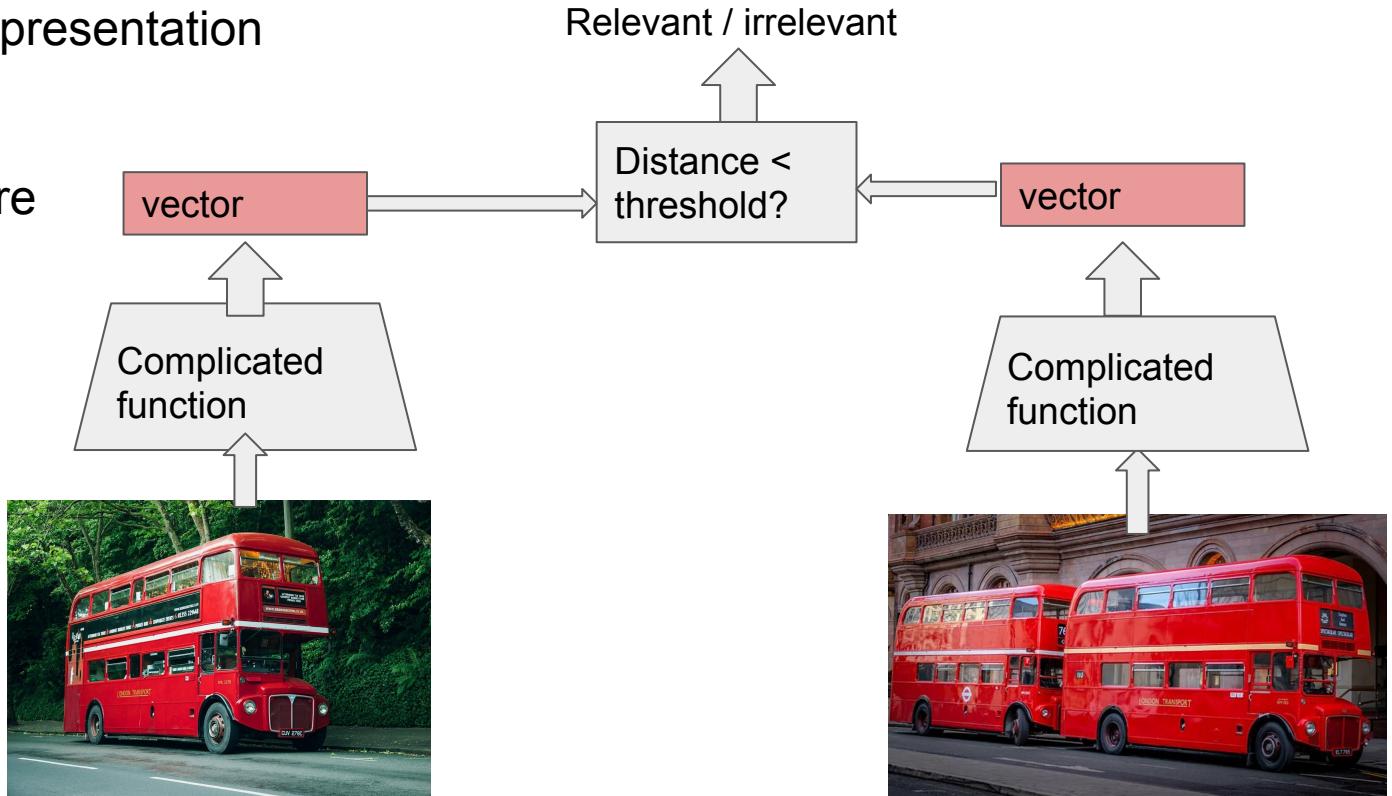
Limitations

- Architecture is complex
 - Directly takes all the image / video pixels as input
 - Expensive to evaluate the function
 - Needs to be evaluated for all the N database elements for a single query
- Compared to
 - Hash indexing O(1)
 - Range search O(log N)
 - Inverted file O(1)
- ⇒ too expensive!

Vector embeddings

Embedding vectors: motivation

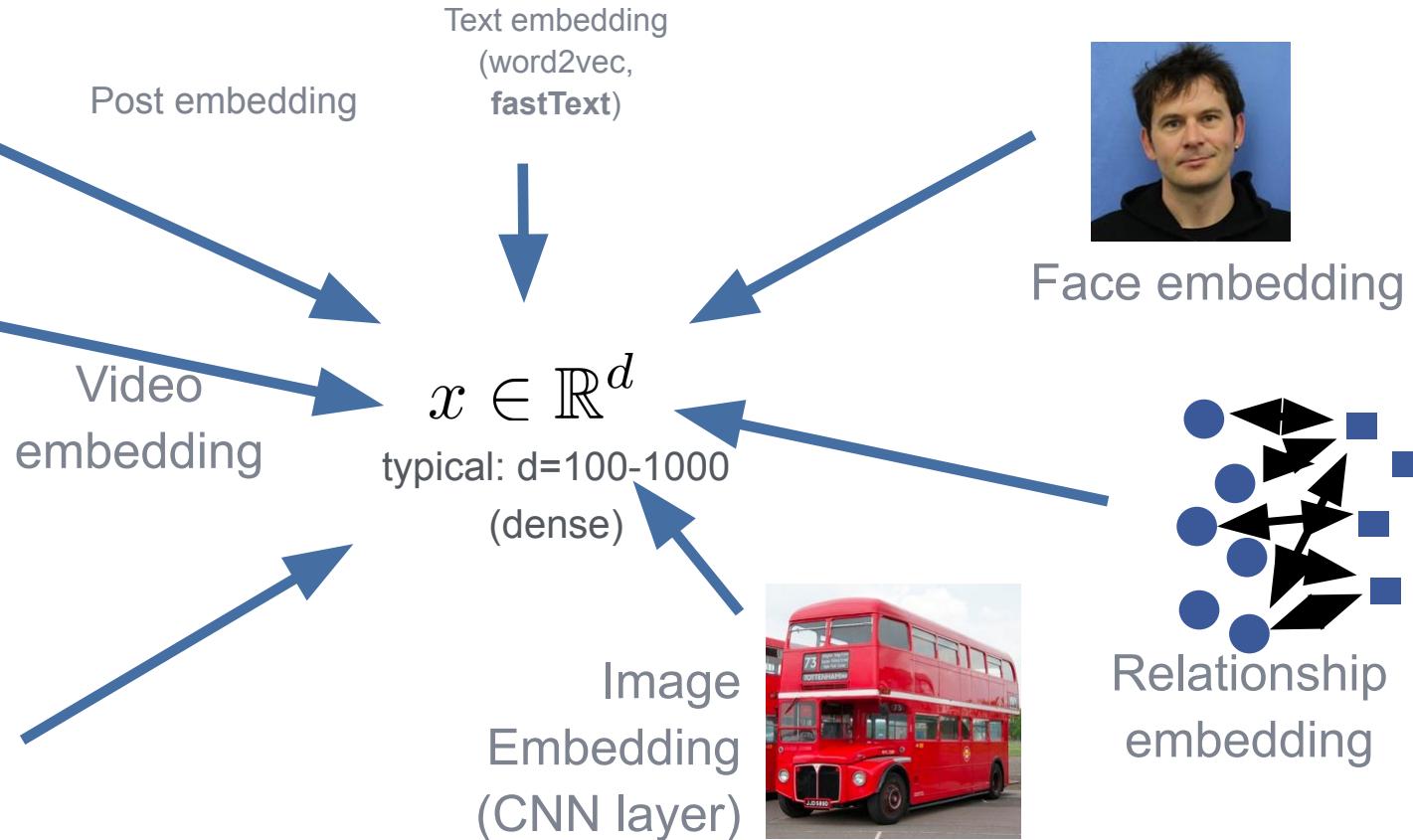
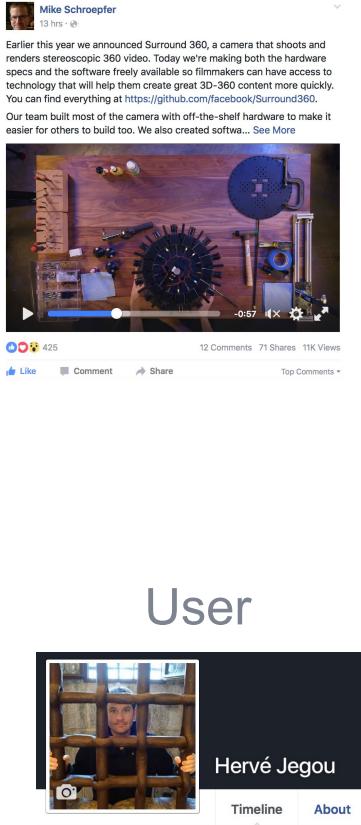
- Intermediate representation
 - simple!
 - The vector
- Easy to compare
- Constrained comparison



What is a vector? (I knew it, but forgot during the summer holidays)

- Table of numbers
 - Real (floating point) numbers
 - Sometimes integers or even single bits
- [0.4, 0.6, -0.5,, 1.5]
- Fixed size d
- Compact to store (more compact than the original data item)
- Cheap to compare – $O(d)$
- Explainable – result is a usable item
 - Unlike some classification approaches

Embedding vectors: motivation

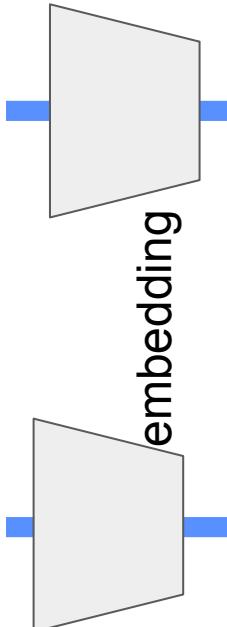


The embedding contract

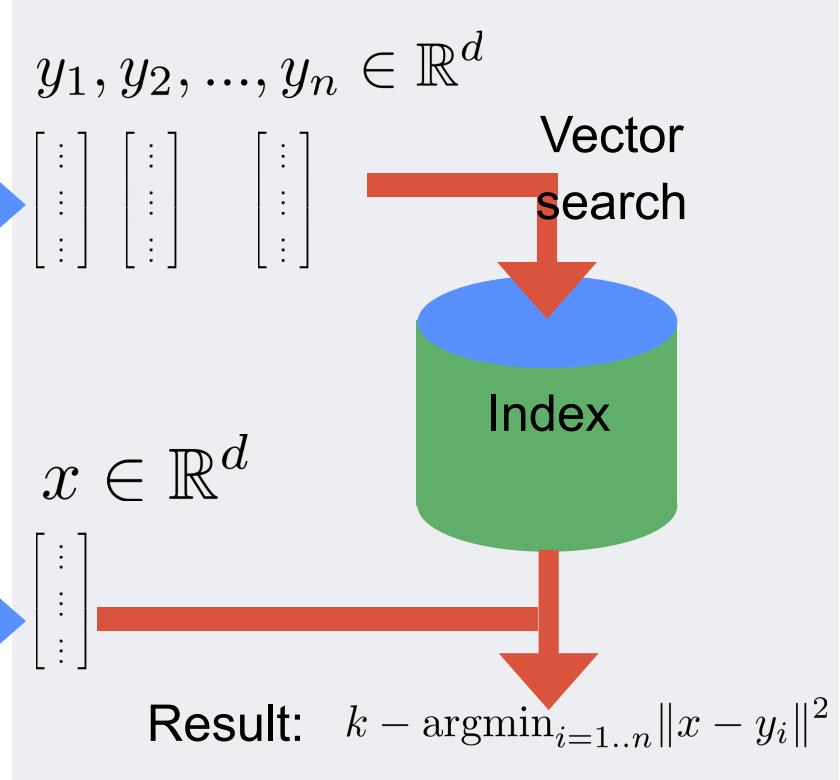
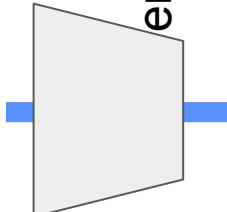
- The embedding extractor and embedding indexer
- Embedding function:
 - “I extract embedding vectors that can be compared”
 - Given distance function
- Embedding indexer (vector search lib):
 - “I will manage the scaling and search”
 - Search – efficiency and accuracy
 - Database operations – update, insertion, removal, migration...

Retrieval with embeddings

Collection:



Query:



Comparing vectors

Vector distances

- Distances that obey the 3 criteria
- For vectors x, y of size d
- The L_p norm family

$$d_p(x, y) = L_p(x - y) = \left(\sum_{i=1}^d (x_i - y_i)^p \right)^{\frac{1}{p}}$$

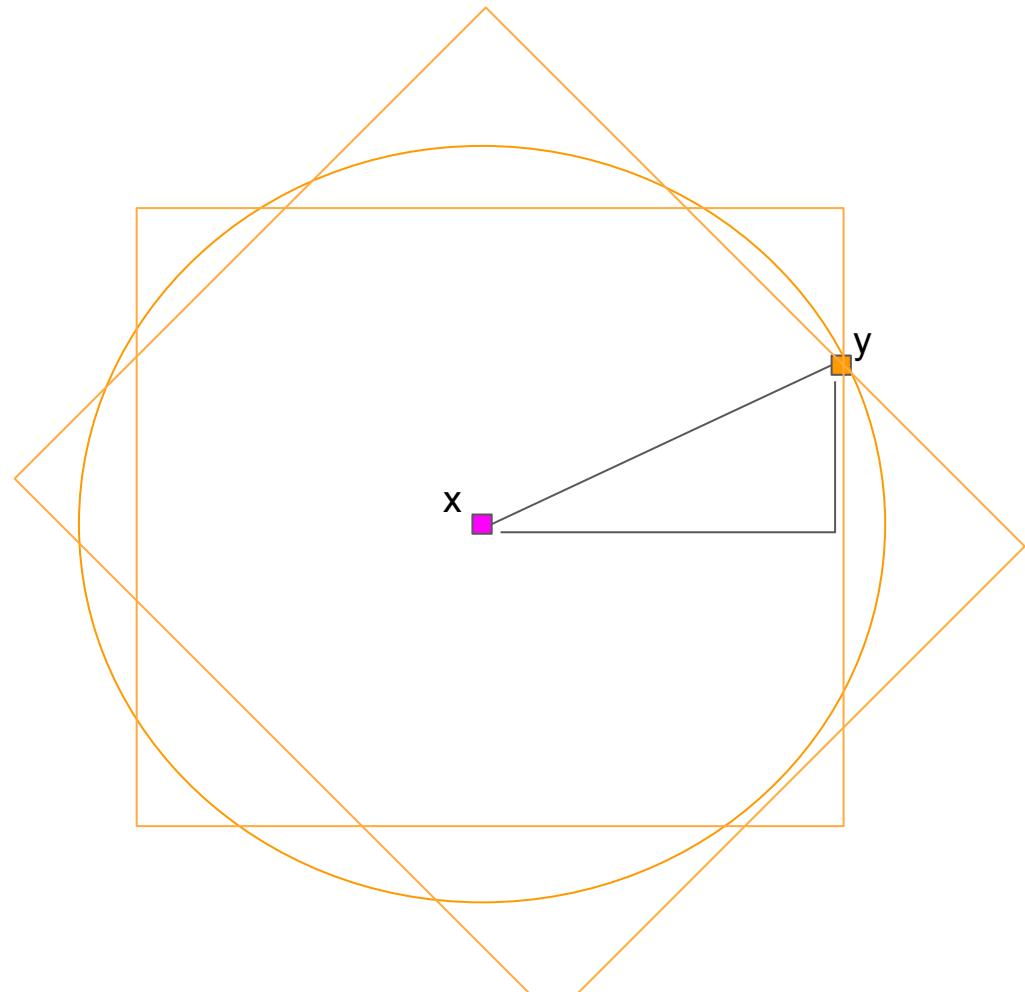
$$\implies d_2(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

$$d_1(x, y) = L_1(x - y) = \sum_{i=1}^d |x_i - y_i|$$

$$d_\infty(x, y) = L_\infty(x - y) = \max_{i=1..d} |x_i - y_i|$$

Vector distances in 2D

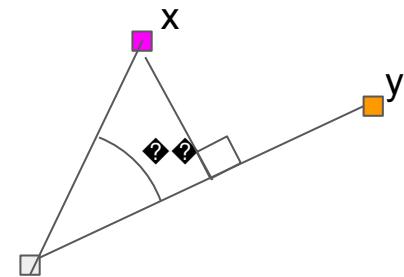
- Spheres
 - Locus of equal distances
- Which sphere is for what distance?



Maximum inner product search

- Not a distance
- Vector inner product as a similarity measure

$$\langle x, y \rangle = \|x\| \times \|y\| \times \cos(\alpha)$$



Cosine similarity

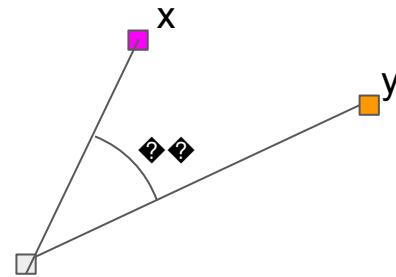
- Angle between two vectors

$$\cos(x, y) = \cos(\alpha)$$

$$= \frac{\langle x, y \rangle}{\|x\| \times \|y\|}$$

$$= \frac{\sum_{i=1}^d x_i y_i}{\sqrt{\sum_{i=1}^d x_i^2} \times \sqrt{\sum_{i=1}^d y_i^2}}$$

- Special case: norm-1 vectors
 - Equivalent to max inner product search
 - Equivalent to L2 search!



Hamming distance

- Between binary vectors (bits)
- Number of differing bits between x and y

$$x = [1, 0, 0, 1]$$

$$y = [1, 1, 0, 0]$$

- Hamming distance = $d_1 = d_2^2$
- Integer between 0 and d
- Easy to compute
 - On integers
 - 2 machine instructions

Bridges between distances

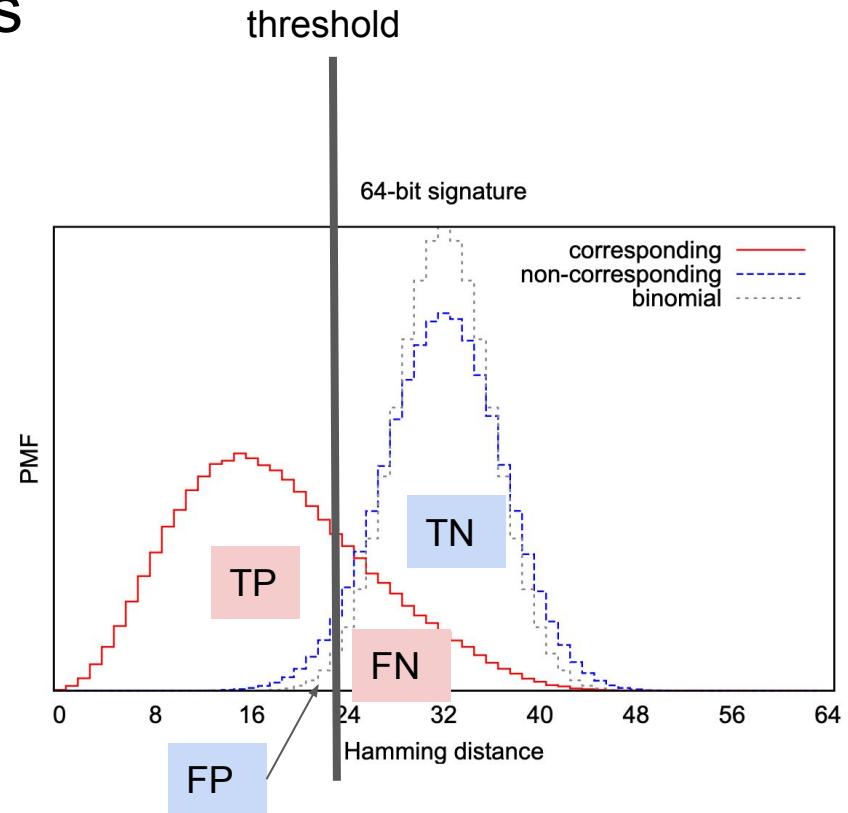
- Useful equality

$$\|x - y\|^2 = \|x\|^2 + \|y\|^2 - 2\langle x, y \rangle$$

- Equivalence for normalized vectors
 - Cosine, max inner product, L2 are equivalent
- L2 \leftrightarrow inner product
 - transforming vectors
 - In dimension d+1

Distances and retrieval results

- Reading TP and FP from distance plots
- Binomial distribution
 - For random binary embeddings
- Usually many more incorrect results than correct ones



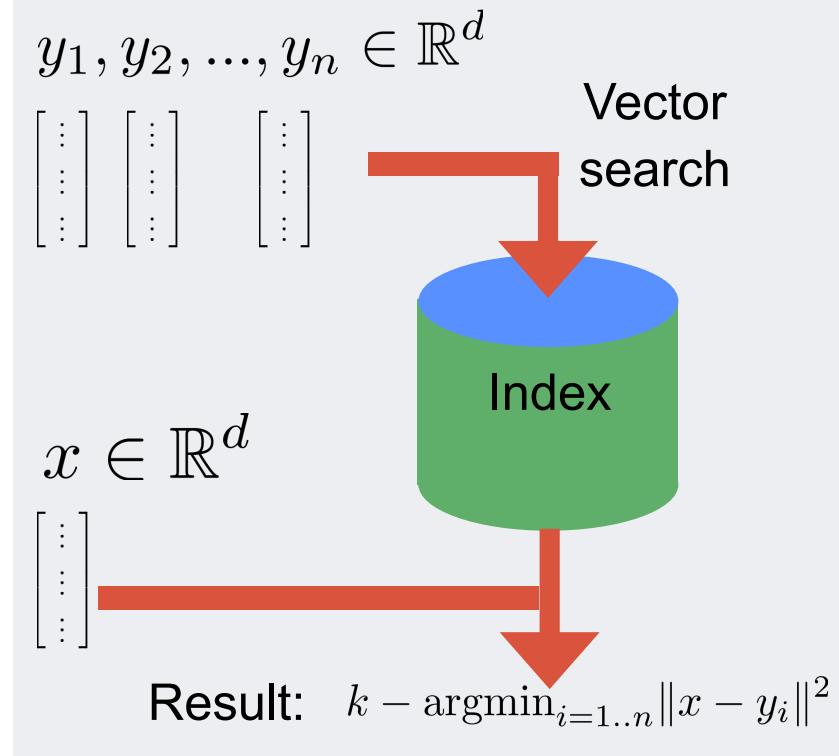
Vector search

One-to-many comparison

- Simple operation

$$k = \operatorname{argmin}_{i=1..n} \|x - y_i\|^2$$

- Can be computed with brute force algorithm
 - $O(d * n)$



Vector search types, ground truth and metrics

- K-nearest neighbor search

- Find the top-k nearest vectors
- Example: search results
- Nearest neighbor recall @ rank k
- Intersection measure

$$k = \operatorname{argmin}_{i=1..n} \|x - y_i\|^2$$

- Range search

- Find all vectors within a radius r
- Example: remove violating images
- Metric: precision & recall

$$\{i \in \{1, \dots, n\} \text{ s.t. } \|x - y_i\| < r\}$$

End-to-end vs. vector search metric

- Correct item returned at rank 1
 - correct item returned at rank 1 with exact vector search
 - Vector search returns correct result for this vector

Accuracy =

- Decompose the metric

Embedding accuracy

X

Vector search accuracy

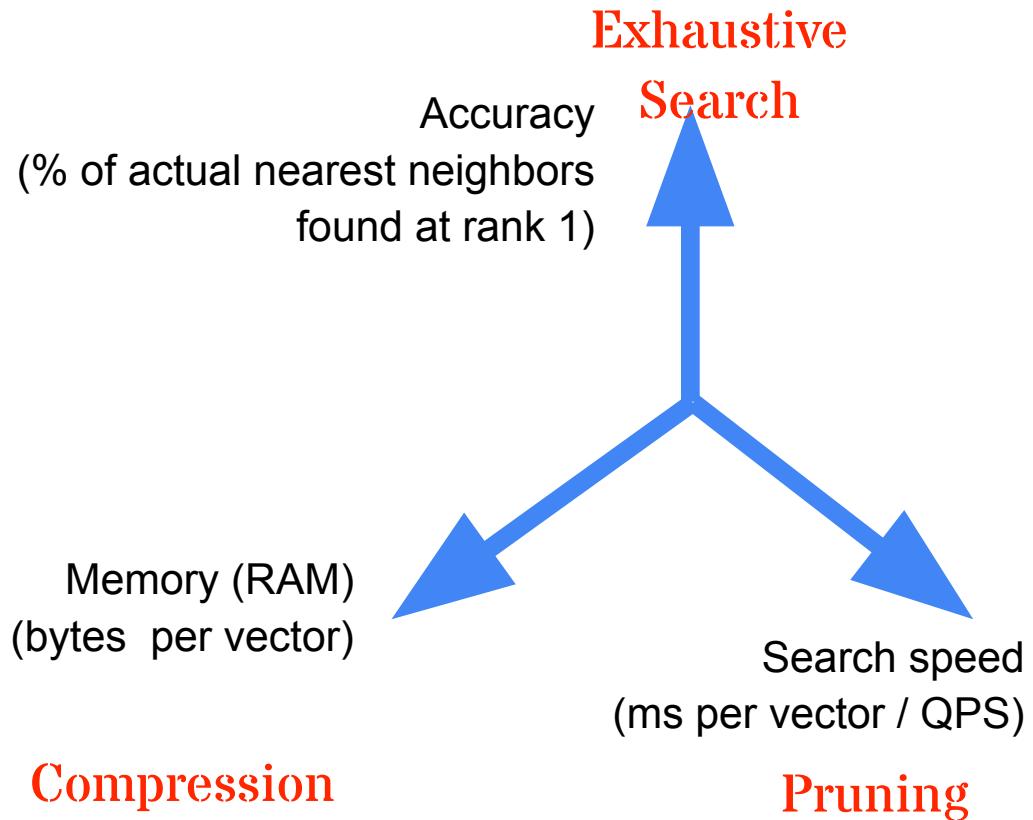
Brute-force vector search

- Used as baseline and building block for other methods
- Complexity
 - Distance computations $O(n*d)$
 - Retrieving top-k: $O(n*\log(k))$
 - Retrieving range search: $O(n)$
- Large factor between slow and fast implementations

Performance axes for vector search

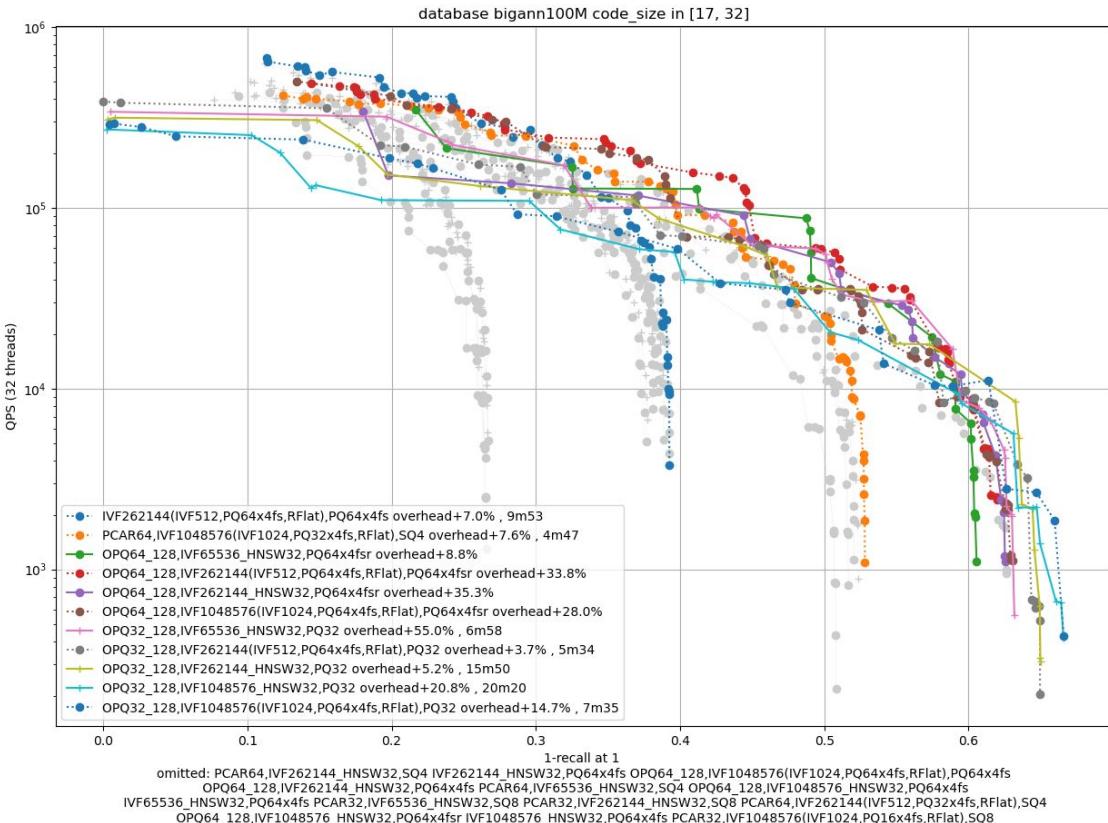
- Other axes:

- Index build time
- Memory overhead
- Training time



Tradeoffs

- Fix dataset
- Fix one axis
- Tradeoffs between the two others
- Pareto-optimal front



What's next...

- 2 classes about how to compute embeddings
- The rest about vector search!