K-Nearest Neighbors, a.k.a., Memory Based Learning, a.k.a., Instance Based Learning, a.k.a. Case Based Reasoning.

Alvaro Soto

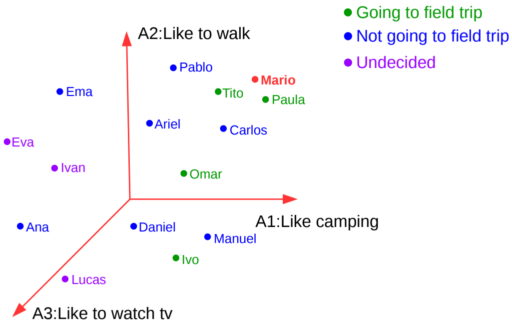
Computer Science Department (DCC), PUC

I. K-Nearest Neighbors

- The most intuitive and simple machine learning technique.
- Case based reasoning: inference about a new instance is directly based on info about known similar cases.
- A supervised machine learning approach. Supervised?
- It can be used for prediction and classification. Prediction and classification?

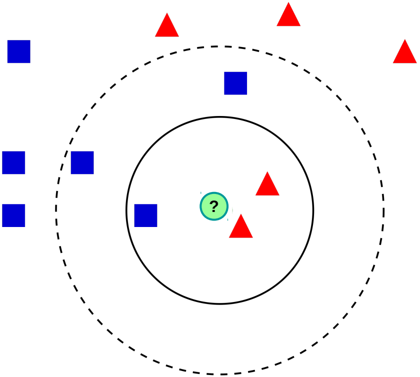
K-Nearest Neighbors

Consider K=1, Is Mario going to the field trip?



K-Nearest Neighbors: Classification

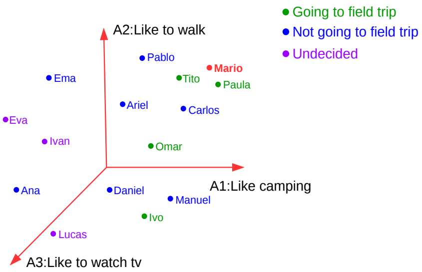
Consider K=3, What class is the instance at the center?



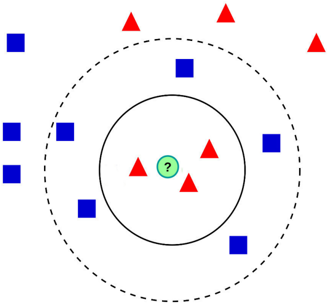
Majority Vote: choose most voted class among K neighbors.

K-Nearest Neighbors: Classification

Consider K=5, Is Mario going to the field trip?

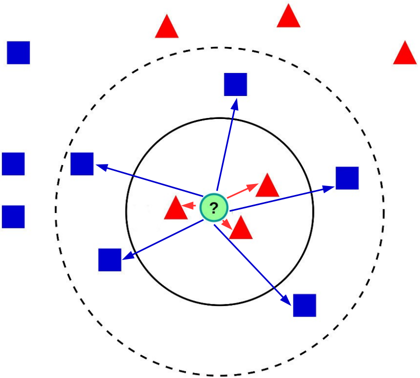


Consider K=8, What class is the instance at the center?



Any problem?, Can we improve this situation?, How?

K-Nearest Neighbors: Classification



Weighted majority vote: weight vote is proportional to the inverse of the distance to the corresponding neighbor.

II. Distance metric

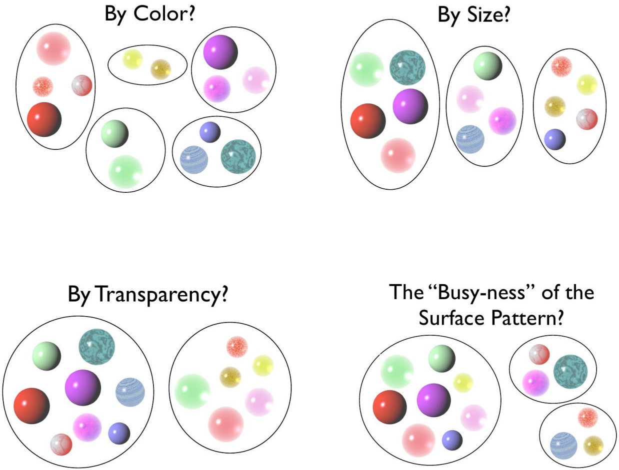
Traditional K-NNs are based on Euclidean distance.
Is this always a good choice?





Distance metric matters

Clustering Marbles



Distance metric matters

- Finding suitable metric distances is one of the main tasks of a machine learning technique.

Distance metric matters

- Finding suitable metric distances is one of the main tasks of a machine learning technique.
- As an example, we can cast supervised classification as a distance metric learning problem. The goal is to find a transformation (new space), where instances of the same class are close together and far from instances of other classes (usually each class is a vertex of a simplex).

Distance metric matters

- Finding suitable metric distances is one of the main tasks of a machine learning technique.
- As an example, we can cast supervised classification as a distance metric learning problem. The goal is to find a transformation (new space), where instances of the same class are close together and far from instances of other classes (usually each class is a vertex of a simplex).
- In general, machine learning techniques move input instances to a new space, where the underlying distance embeds semantic relations among the instances. Semantic is given by the information in the labels.

Distance metric matters

- Finding suitable metric distances is one of the main tasks of a machine learning technique.
- As an example, we can cast supervised classification as a distance metric learning problem. The goal is to find a transformation (new space), where instances of the same class are close together and far from instances of other classes (usually each class is a vertex of a simplex).
- In general, machine learning techniques move input instances to a new space, where the underlying distance embeds semantic relations among the instances. Semantic is given by the information in the labels.
- In the this course, we will learn that deep learning is an excellent tool to build semantic distance spaces.

What About K-NN on Non Euclidean spaces?

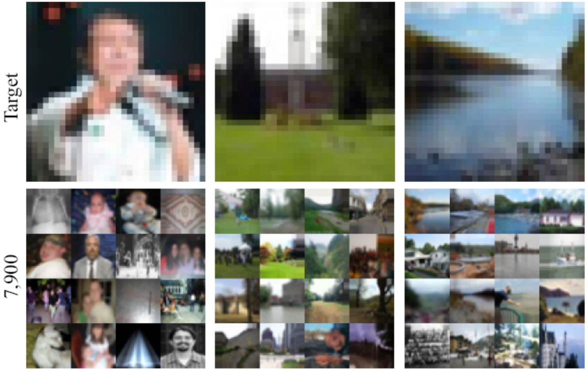
Case Study:

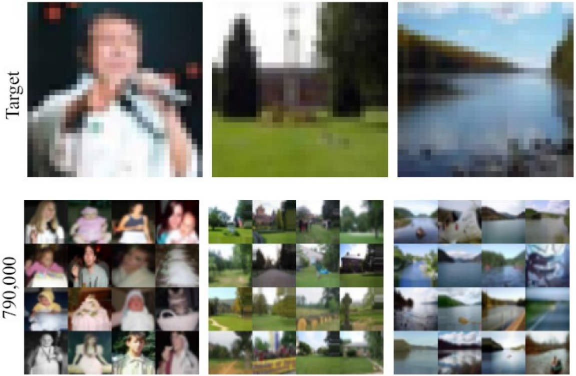
80 million tiny images: A large dataset for non-parametric object and scene recognition.

A. Torralba, R. Fergus and W. T. Freeman.
IEEE PAMI, 2006.

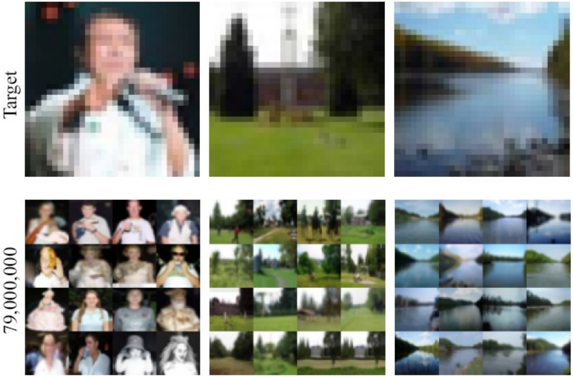
What can we do with 80M images?

K-NN Non Eucliden spaces

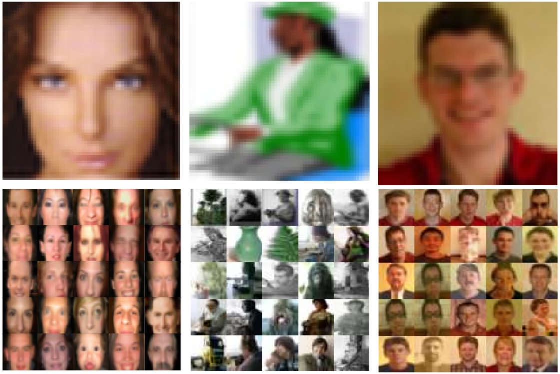



Target
790,000

K-NN Non Eucliden spaces



Big data makes magic!



Distance metric between images.

$$D^2_{\text{ssd}} = \sum_{x,y,c} (I_1(x,y,c) - I_2(x,y,c))^2$$

$$D^2_{\text{warp}} = \min_{\theta} \sum_{x,y,c} (I_1(x,y,c) - T_\theta[I_2(x,y,c)])^2$$

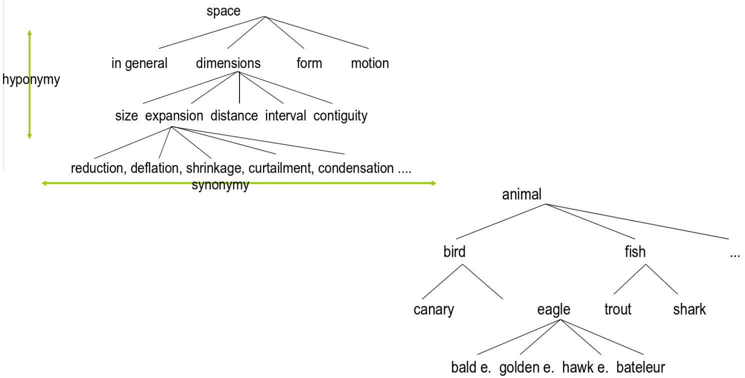
Patch horizontal mirror, translations and scaling.

$$D^2_{\text{shift}} = \min_{|D_{x,y}|\le w} \sum_{x,y,c} (I_1(x,y,c) - \hat{I}_2(x+D_x, y+D_y, c))^2$$

Individual pixel shift in 5x5 window. Tθ given By best Dwarp.

Keywords

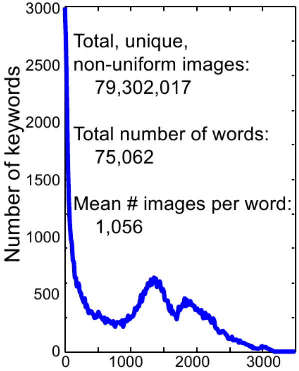
WordNet: lexical database of English words. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of semantic and lexical relations.



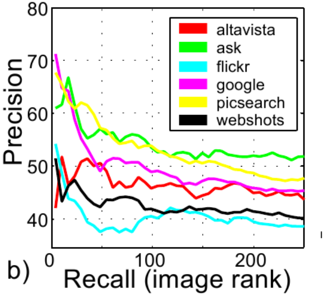
Dataset

- **Words:** They extract all non-abstract nouns from Wordnet, a total of 75.846 words.
- **Images** : For each word, they automatically download all the images provided by 7 independent image search engines: Altavista, Ask, Flickr, Cydral, Google, Picsearch and Webshots.
- Running over 8 months, they collect approx. 80M images (97.245.098 images).
- Finally, they filter out the dataset removing duplicate and uniform images (images with zero variance). Also, they remove words that do not produce enough images (around 1% of the keywords have no images). The final dataset contains 79.302.017 images from 75.062 words.
- Due to efficiency reasons, they store the images using a resolution of 32x32 pixels.

K-NN Non Eucliden spaces



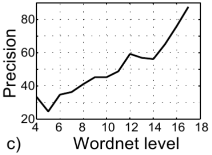
K-NN Non Eucliden spaces

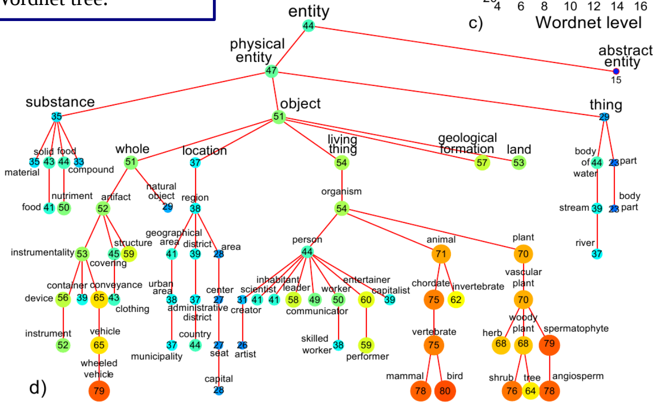


Retrieval accuracy drops after the 100th image. In average first 44 images are correct.

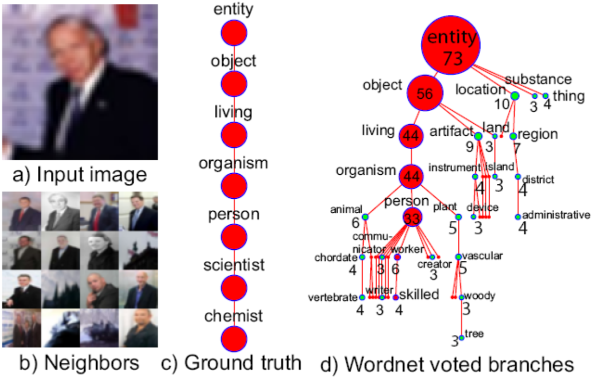
Accuracy of labeling for different nodes of a portion of the Wordnet tree.

More precision for more specific words

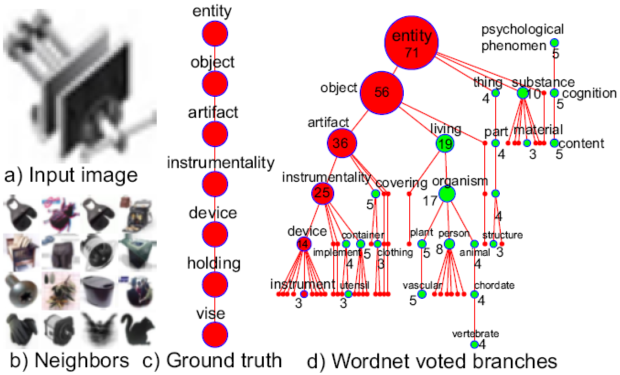




Object recognition results



Object recognition results

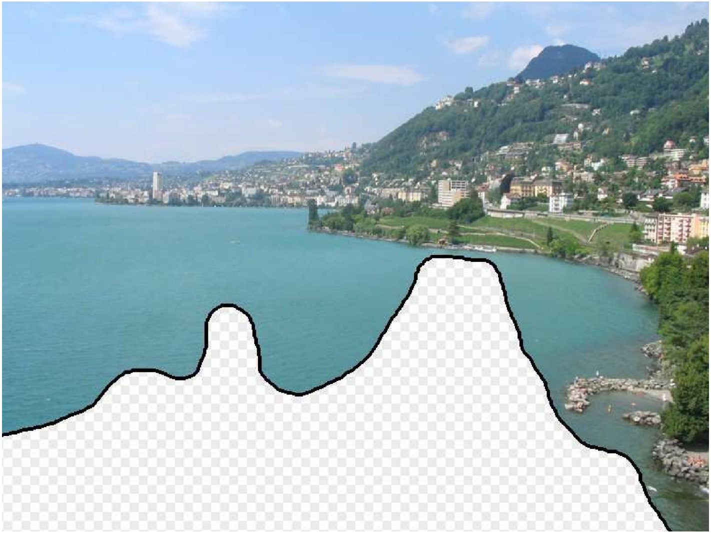


Case Study:

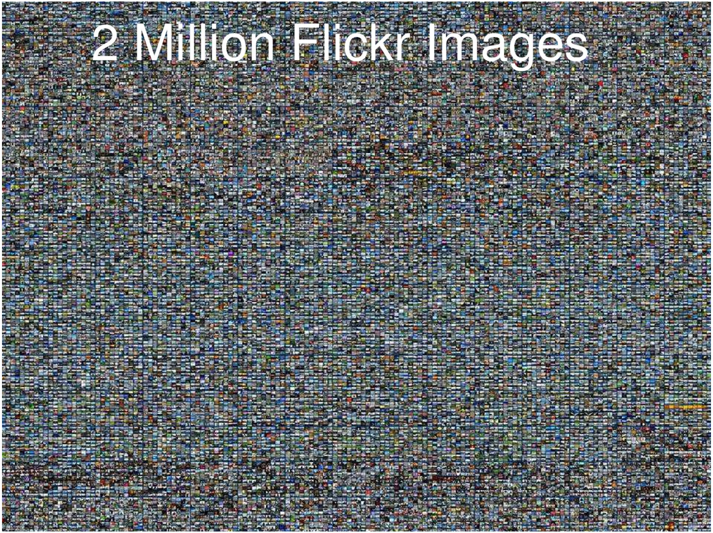
Scene Completion using Millions of Photographs.
J. Hays and A. Efros
SIGGRAPH 2007.

K-NN Non Eucliden spaces

K-NN Non Eucliden spaces

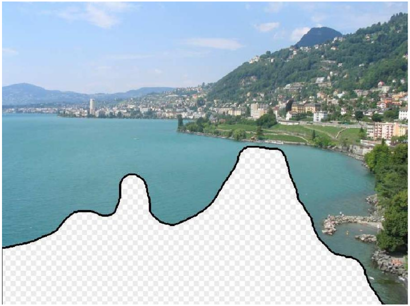


K-NN Non Eucliden spaces



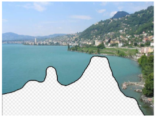
K-NN Non Eucliden spaces

Input Image



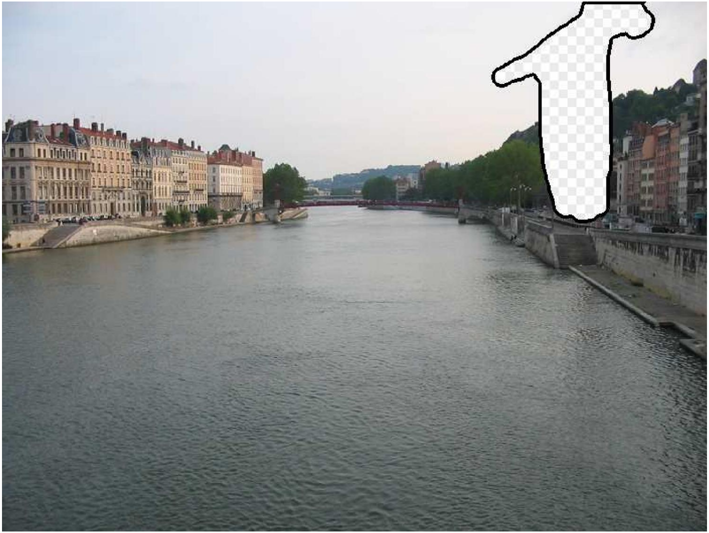
Most similar image
in Flickr dataset

K-NN Non Eucliden spaces




K-NN Non Eucliden spaces

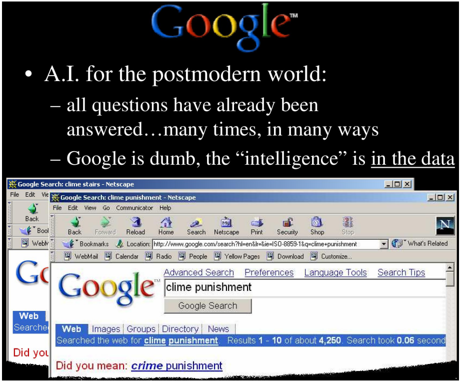
K-NN Non Eucliden spaces



K-NN Non Eucliden spaces

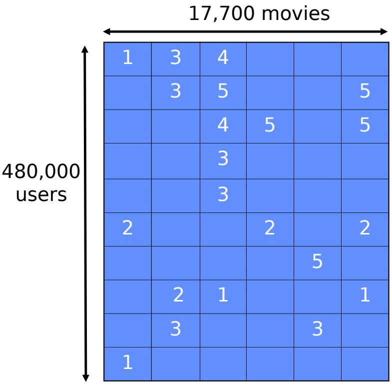
K-NN Non Eucliden spaces

A suitable distance metric ?



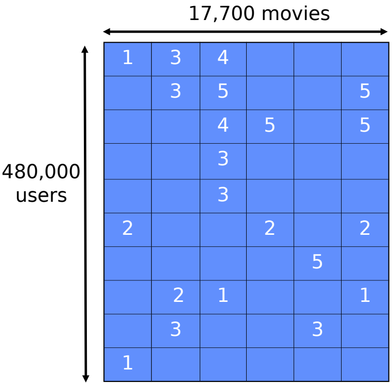
Netflix contest dataset

- 100 million ratings.
- Atributes = [user, movie-id, time-stamp, rating value].
- Generated by users between Oct. 1998 and Dec. 2005.
- Users randomly chosen among set with at least 20 ratings (small perturbations to help with anonymity).



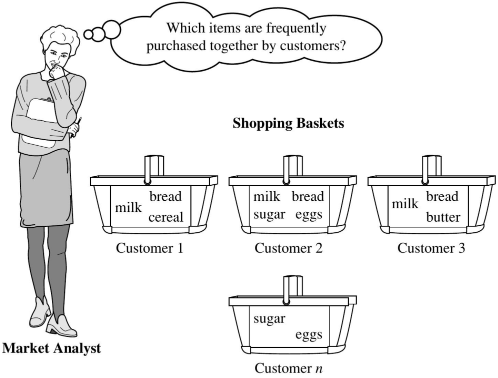
Netflix contest dataset

Problem: dataset is 99% sparse



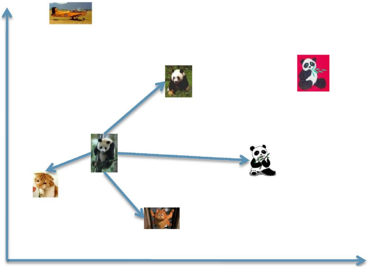
Is this a problem for a K-NN strategy?

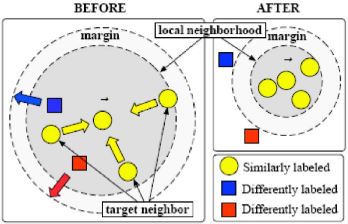
Association rules



In many cases, Euclidean distance is not a good way to estimate semantic distances between feature vectors. However, this can be valid in subspaces of the original feature space.

Learning a distance metric: Optimizing cost function





Learning a distance metric: Optimizing cost function

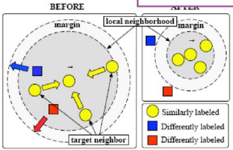
$$\varepsilon(\mathbf{L}) = \sum_{ij} \eta_{ij} \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 + c \sum_{ijl} \eta_{ij}(1-y_{il}) \left[1 + \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 - \|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2\right]_+$$

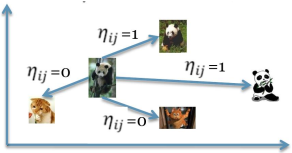
For inputs and target neighbors
It is equal to 1

$y_{il} \in \{0,1\}$ indicates if $\vec{x}_i$ and $\vec{x}_l$ has same label. So For input and neighbors having different labels, it is equal to 1

Distance between inputs and target neighbors

Distance between input and neighbors with different labels





$$\mathcal{D}(\vec{x}_i, \vec{x}_j) = (\vec{x}_i - \vec{x}_j)^\top \mathbf{M} (\vec{x}_i - \vec{x}_j)$$

$$\mathbf{M} = \mathbf{L}^\top \mathbf{L}.$$

Relevant ideas from the previous slides

1. KNN achieves amazing results when using large datasets.

Relevant ideas from the previous slides

1. KNN achieves amazing results when using large datasets.
2. A critical issue to apply a KNN strategy is to use a suitable distance metric.

Relevant ideas from the previous slides

1. KNN achieves amazing results when using large datasets.
2. A critical issue to apply a KNN strategy is to use a suitable distance metric.
3. Supervided machine learning is closely related to learning suitable metric distances.

Relevant ideas from the previous slides

1. KNN achieves amazing results when using large datasets.
2. A critical issue to apply a KNN strategy is to use a suitable distance metric.
3. Supervided machine learning is closely related to learning suitable metric distances.
4. There several approaches to learn metric distances.

Relevant ideas from the previous slides

1. KNN achieves amazing results when using large datasets.
2. A critical issue to apply a KNN strategy is to use a suitable distance metric.
3. Supervided machine learning is closely related to learning suitable metric distances.
4. There several approaches to learn metric distances.
    - Finding subspaces

Relevant ideas from the previous slides

1. KNN achieves amazing results when using large datasets.
2. A critical issue to apply a KNN strategy is to use a suitable distance metric.
3. Supervided machine learning is closely related to learning suitable metric distances.
4. There several approaches to learn metric distances.
    - Finding subspaces
    - Optimizing a suitable loss function

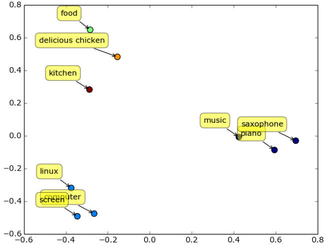
Relevant ideas from the previous slides

1. KNN achieves amazing results when using large datasets.
2. A critical issue to apply a KNN strategy is to use a suitable distance metric.
3. Supervided machine learning is closely related to learning suitable metric distances.
4. There several approaches to learn metric distances.
   - Finding subspaces
   - Optimizing a suitable loss function
   - And so on ...

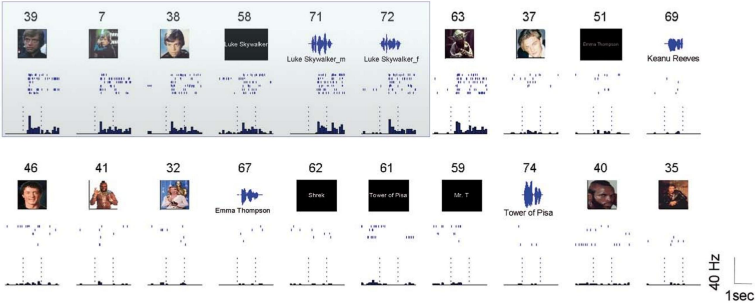
Distributed Representations

Distributed Representations

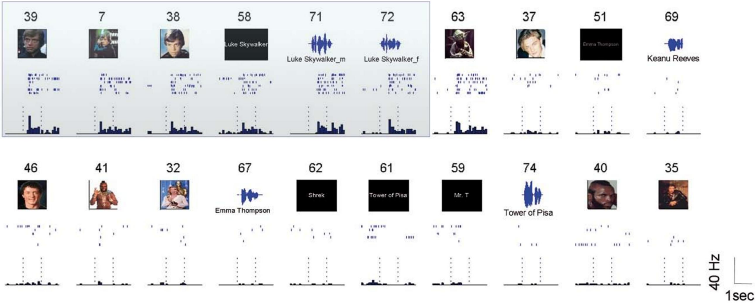
Learning a semantic space (semantic embedding, semantic distance).



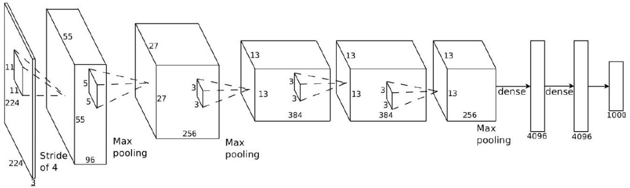
III. Distributed Representations

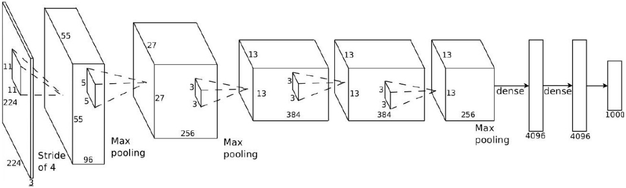


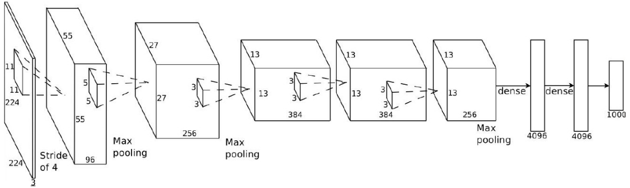
III. Distributed Representations



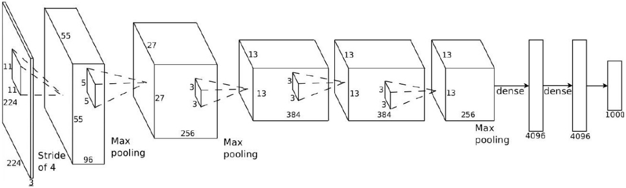
Embedding space can combine information from different sensing modalities: vision, audio, text, etc.

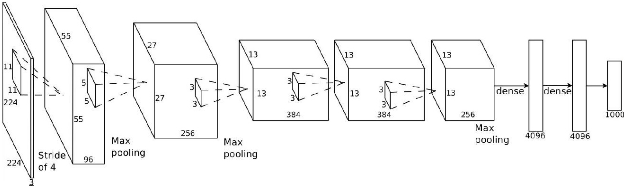

11
11
224
224
3
Stride
of 4
55
55
96
5
5
Max
pooling
27
27
256
3
3
Max
pooling
13
13
384
3
3
13
13
384
3
3
13
13
256
Max
pooling
dense
dense
4096
4096
1000

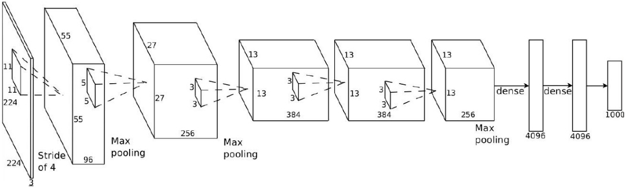

224
224
3
11
11
Stride of 4
55
55
96
5
5
Max pooling
27
27
256
3
3
Max pooling
13
13
384
3
3
13
13
384
3
3
13
13
256
Max pooling
dense
dense
4096
4096
1000

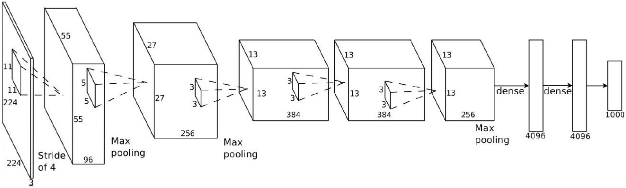


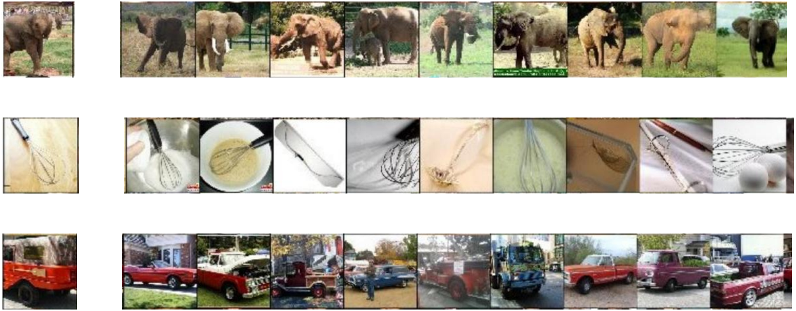
224
224
3
11
11
Stride of 4
55
55
96
5
5
Max pooling
27
27
256
3
3
Max pooling
13
13
384
3
3
13
13
384
3
3
13
13
256
Max pooling
dense
dense
4096
4096
1000

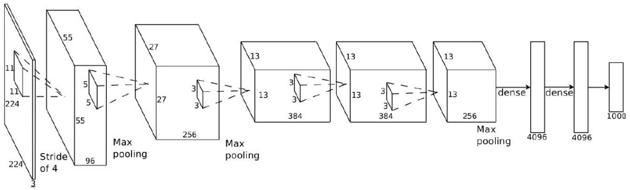

224
224
3
11
11
Stride of 4
55
55
96
5
5
Max pooling
27
27
256
3
3
Max pooling
13
13
384
13
13
384
13
13
256
Max pooling
dense
dense
4096
4096
1000


11
11
224
224
3
Stride of 4
55
55
96
5
5
Max pooling
27
27
256
3
3
Max pooling
13
13
384
3
3
13
13
384
3
3
13
13
256
Max pooling
dense
dense
4096
4096
1000


224
224
3
11
11
Stride of 4
55
55
96
5
5
Max pooling
27
27
256
3
3
Max pooling
13
13
384
3
3
13
13
384
3
3
13
13
256
Max pooling
dense
dense
4096
4096
1000


224
224
3
11
11
Stride of 4
55
55
96
5
5
Max pooling
27
27
256
3
3
Max pooling
13
13
384
3
3
13
13
384
3
3
13
13
256
Max pooling
dense
4096
dense
4096
1000





Deep learning models are excellent tools to move data to meaningful spaces, where simple Euclidean distance reveals semantic relations present in the training data (labels).

IV. Finding the NNs

- Obtaining a suitable distance metric is just part of the story.
- What problem will we face when we apply that distance using a KNN strategy?
- Hint: consider a big data scenario.

IV. Finding the NNs

- Obtaining a suitable distance metric is just part of the story.
- What problem will we face when we apply that distance using a KNN strategy?
- Hint: consider a big data scenario.

Locality Sensitive Hashing (LSH) to the rescue

Hashing

- Design a hash funtion $f(\cdot)$ that maps each possible input query $\mathbf{q}$ to a different integer value $Y_k \in [0, K]$, (where K a big value, greater than $|\mathbf{q}|$).

Hashing

- Design a hash funtion $f(\cdot)$ that maps each possible input query **q** to a different integer value $Y_k \in [0, K]$, (where K a big value, greater than $|\mathbf{q}|$).
- Define an array with K positions.

Hashing

- Design a hash funtion $f(\cdot)$ that maps each possible input query $\mathbf{q}$ to a different integer value $Y_k \in [0, K]$, (where K a big value, greater than $|\mathbf{q}|$).
- Define an array with K positions.
- Store info relevant to query $\mathbf{q}$ in array position $f(\mathbf{q})$.

Hashing

- Design a hash funtion $f(\cdot)$ that maps each possible input query $\mathbf{q}$ to a different integer value $Y_k \in [0, K]$, (where K a big value, greater than $|\mathbf{q}|$).
- Define an array with K positions.
- Store info relevant to query $\mathbf{q}$ in array position $f(\mathbf{q})$.

Hashing

- Design a hash funtion $f(\cdot)$ that maps each possible input query $\mathbf{q}$ to a different integer value $Y_k \in [0, K]$, (where K a big value, greater than $|\mathbf{q}|$).
- Define an array with K positions.
- Store info relevant to query $\mathbf{q}$ in array position $f(\mathbf{q})$.

Complexity to acces info relevant to query $\mathbf{q}$?

Hashing

- Design a hash funtion $f(\cdot)$ that maps each possible input query $\mathbf{q}$ to a different integer value $Y_k \in [0, K]$, (where K a big value, greater than $|\mathbf{q}|$).
- Define an array with K positions.
- Store info relevant to query $\mathbf{q}$ in array position $f(\mathbf{q})$.

Complexity to acces info relevant to query $\mathbf{q}$?

Requirements:

- $f(\cdot)$ must avoid collisions, why?.
- $f(\cdot)$ must be quick to compute, why?.

Locality Sensitive Hashing (LSH)

- Same than before, but LSH does not avoid collisions between related items (ex. semantically similar objects).

Locality Sensitive Hashing (LSH)

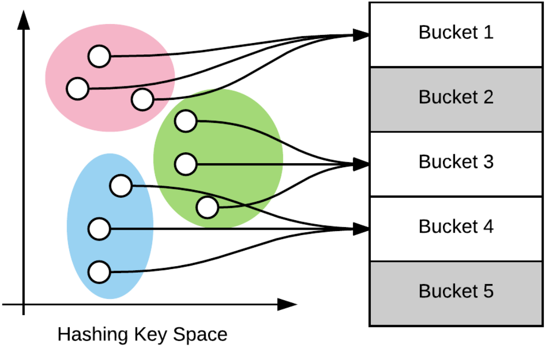
- Same than before, but LSH does not avoid collisions between related items (ex. semantically similar objects).
- LSH approximates nearest-neighbor queries in a way that scales well even for huge datasets.

Locality Sensitive Hashing (LSH)

- Same than before, but LSH does not avoid collisions between related items (ex. semantically similar objects).
- LSH approximates nearest-neighbor queries in a way that scales well even for huge datasets.

Locality Sensitive Hashing (LSH)

- Same than before, but LSH does not avoid collisions between related items (ex. semantically similar objects).
- LSH approximates nearest-neighbor queries in a way that scales well even for huge datasets.



Conclusions

KNNs: A powerful approach.

Main ingredients:

1. Tons of data, including metadata such as labels.

Conclusions

KNNs: A powerful approach.

Main ingredients:

1. Tons of data, including metadata such as labels.
2. Suitable distance metric $\rightarrow$ distributed representations

Conclusions

KNNs: A powerful approach.

Main ingredients:

1. Tons of data, including metadata such as labels.
2. Suitable distance metric $\rightarrow$ distributed representations
3. Fast NN retrieval $\rightarrow$ suitable hashing method.

Conclusions

KNNs: A powerful approach.

Main ingredients:

1. Tons of data, including metadata such as labels.
2. Suitable distance metric $\rightarrow$ distributed representations
3. Fast NN retrieval $\rightarrow$ suitable hashing method.

Conclusions

KNNs: A powerful approach.

Main ingredients:

1. Tons of data, including metadata such as labels.
2. Suitable distance metric $\rightarrow$ distributed representations
3. Fast NN retrieval $\rightarrow$ suitable hashing method.

DL World

- We will provide ingredient 1, DL will take care of the rest!

Conclusions

KNNs: A powerful approach.

Main ingredients:

1. Tons of data, including metadata such as labels.
2. Suitable distance metric $\rightarrow$ distributed representations
3. Fast NN retrieval $\rightarrow$ suitable hashing method.

DL World

- We will provide ingredient 1, DL will take care of the rest!
- DL: "Learning an Efficent and Robust Associative Memory System"