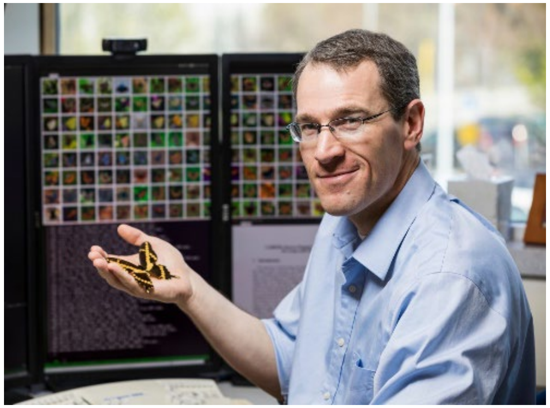
Computer Vision: Understanding, Interpreting and Learning from Visual Data

DATA SCIENCE SUMMER INSTITUTE

Lawrence Livermore National Laboratory



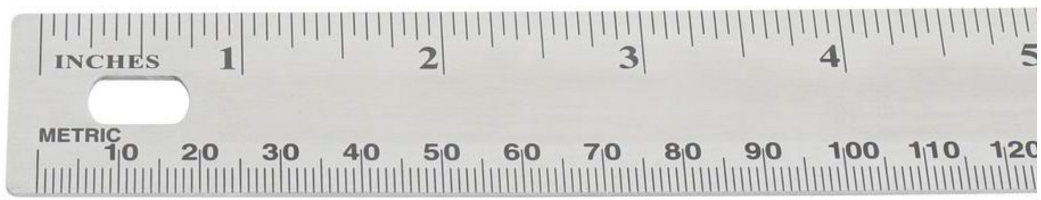
Ryan Farrell

BYU
BRIGHAM YOUNG UNIVERSITY

July 25, 2019
July 30, 2019
July 31, 2019
August 1, 2019

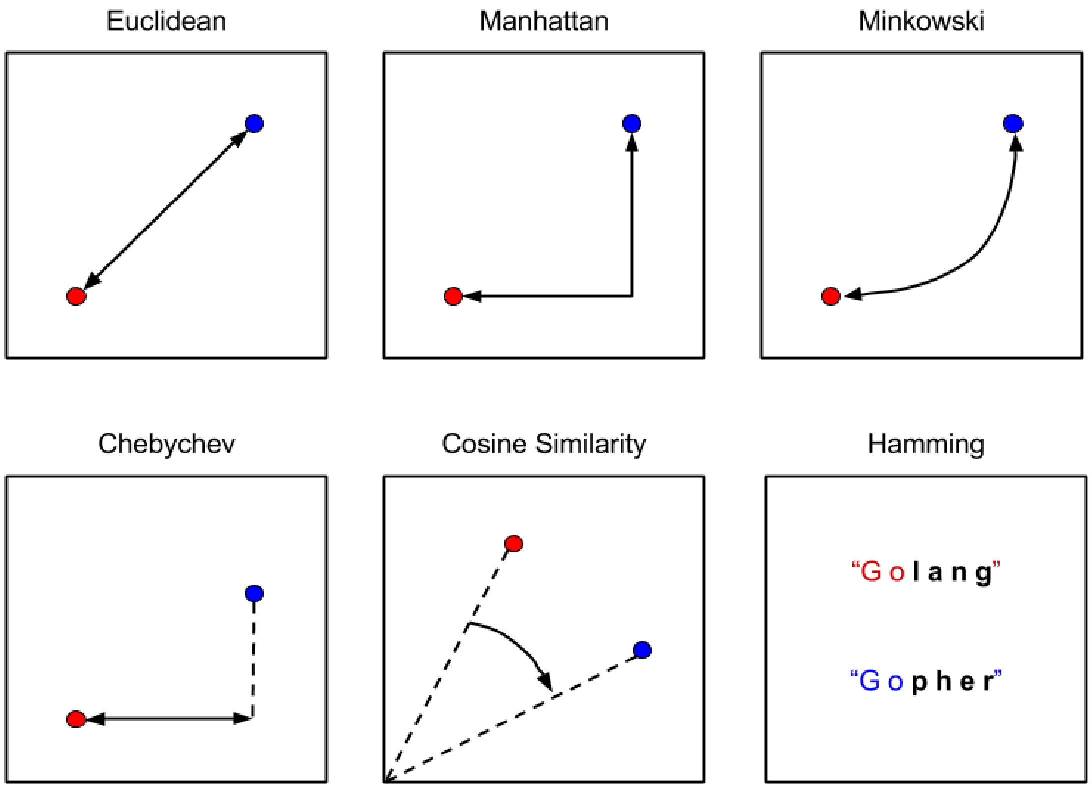
Day 2 – Clustering / Metric Learning

Similarity / Distance

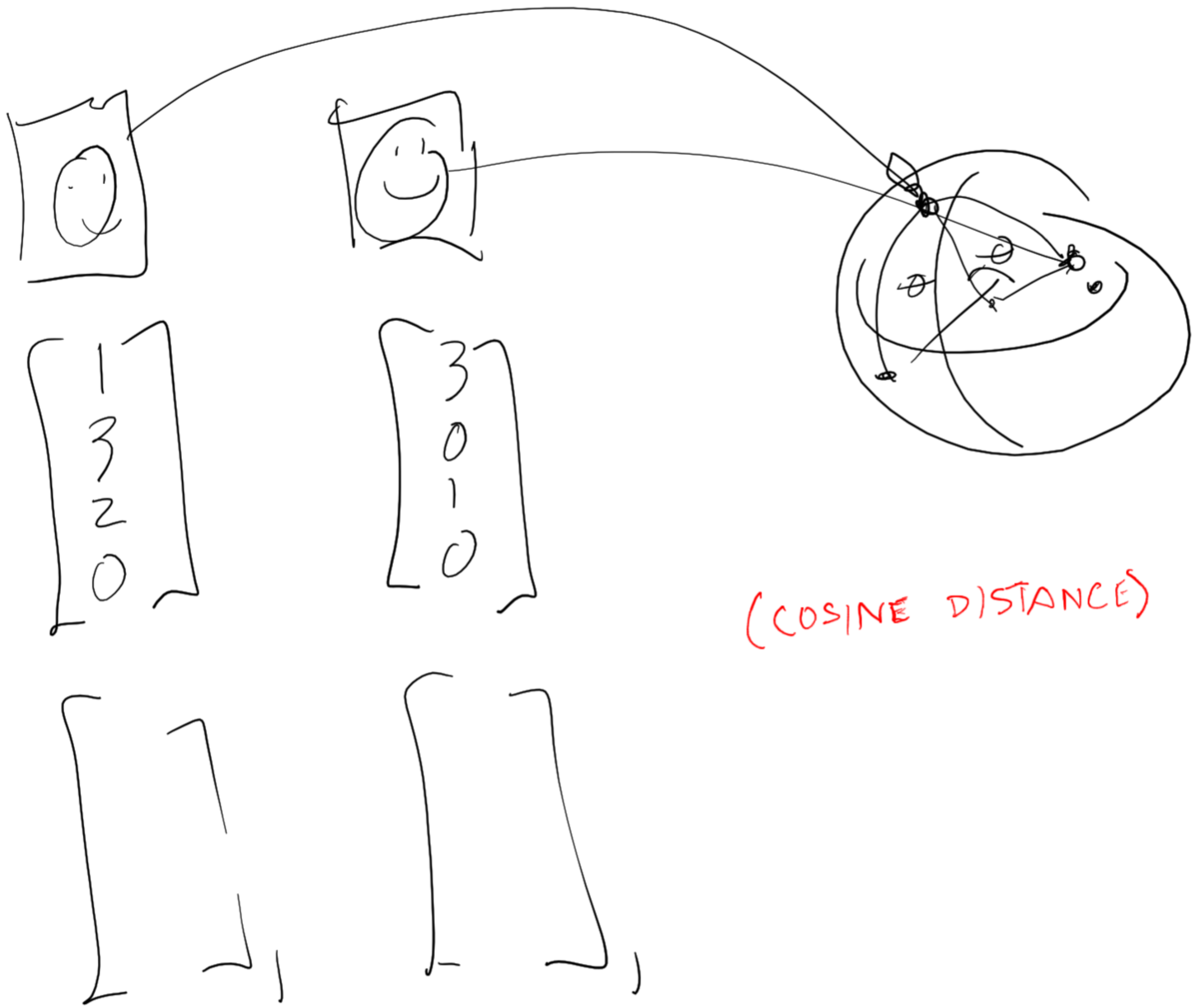


Distance Metrics

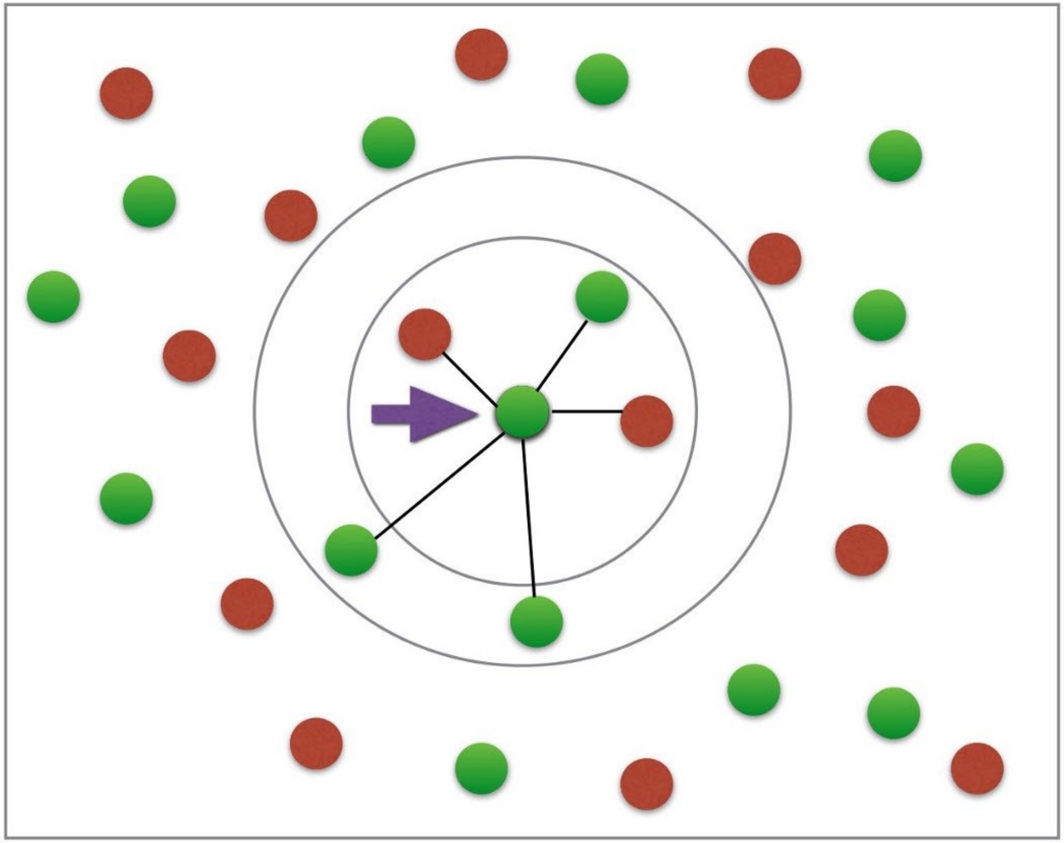
$$D(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{l=1}^{d} |x_{il} - x_{jl}|^{1/p} \right)^{p}$$



From Packt

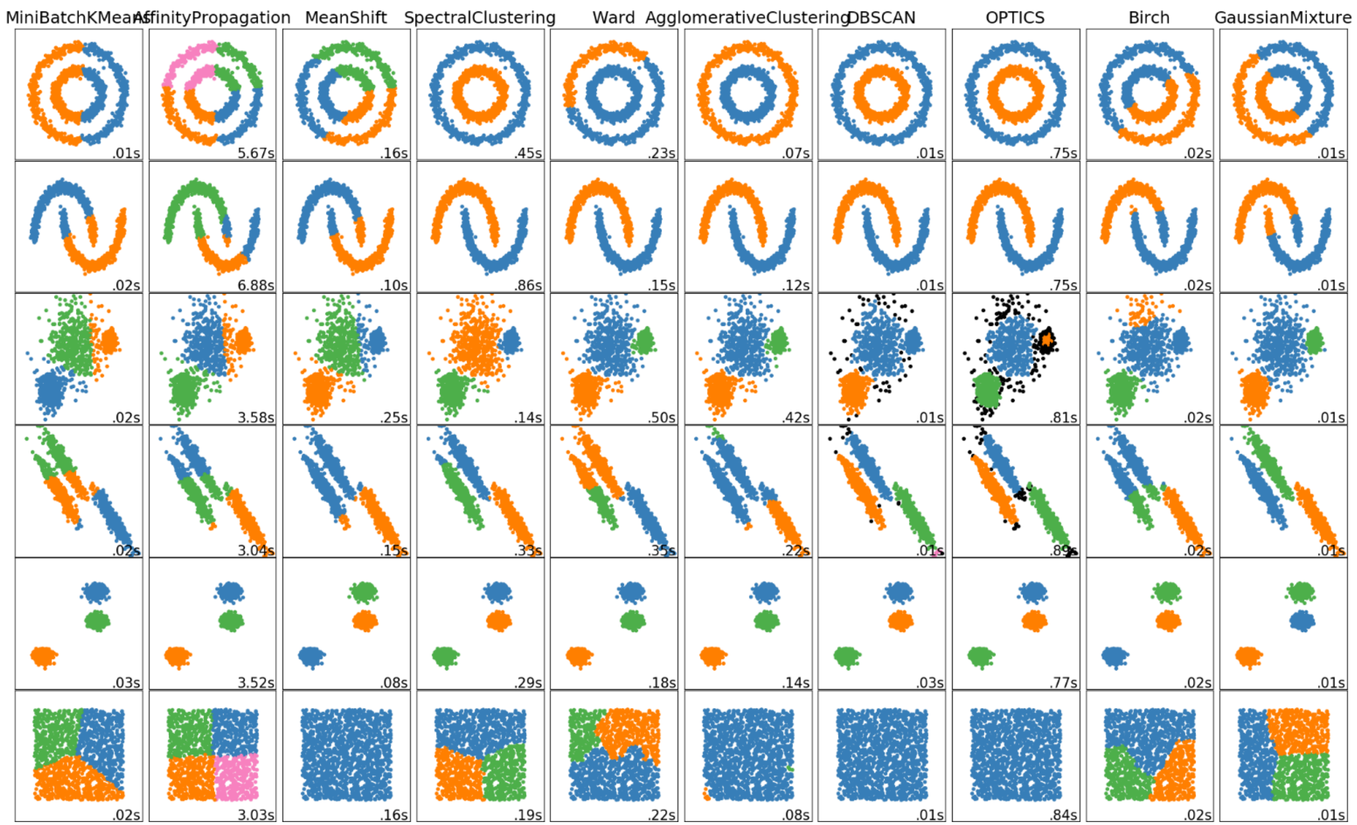

1 3 2 0
3 0 1 0
(COSINE DISTANCE)

K-NN: K-Nearest Neighbors



Source: [TowardsDataScience]

Clustering



Source: scikit-learn.org

K-Means

k = number of clusters

The k-means clustering algorithm

$x^{(i)}$ = data samples

In the clustering problem, we are given a training set $\{x^{(1)}, \ldots, x^{(n)}\}$, and want to group the data into a few cohesive "clusters." Here, $x^{(i)} \in \mathbb{R}^d$ as usual; but no labels $y^{(i)}$ are given. So, this is an unsupervised learning problem.

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^d$ randomly.

2. Repeat until convergence: {

   For every i, set

   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

   $c^{(i)}$ = cluster assignment

   For each j, set

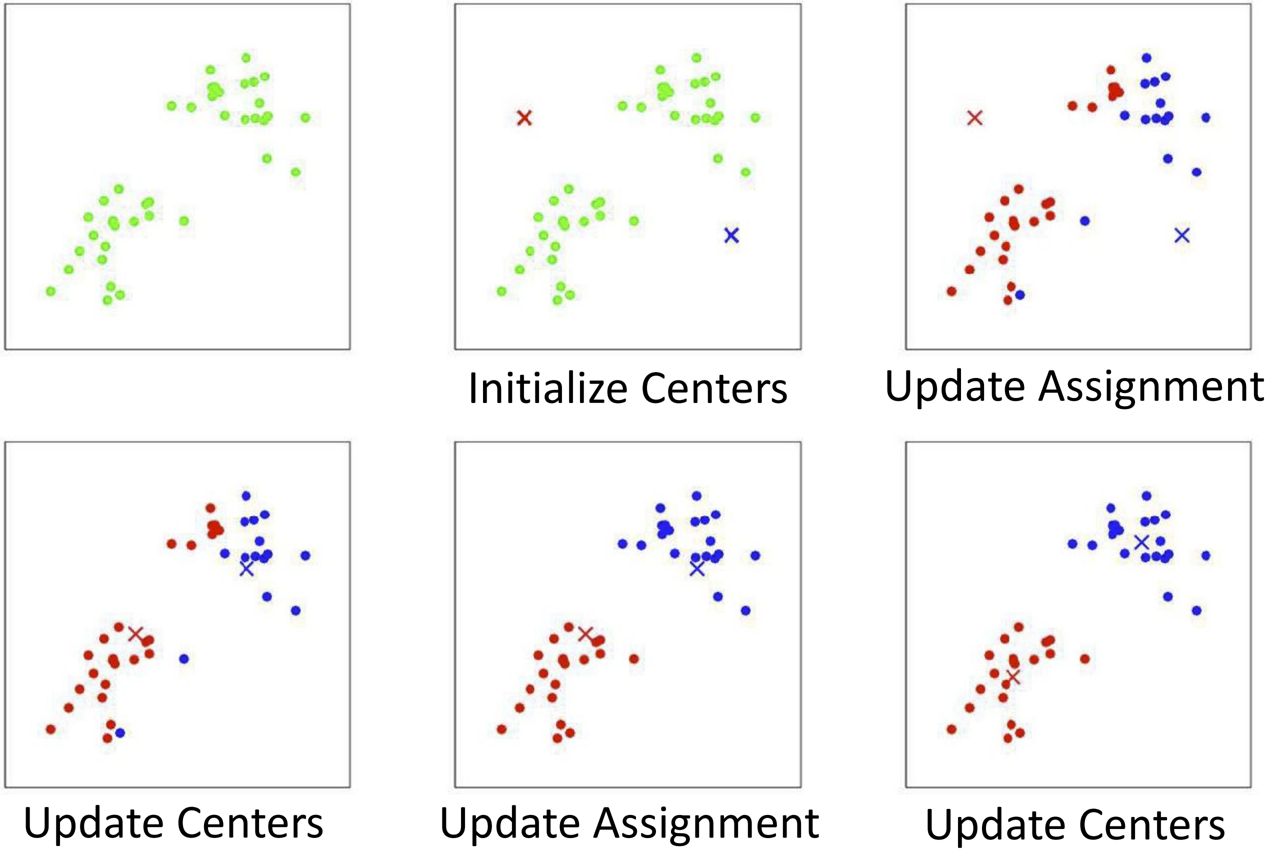
   $$\mu_j := \frac{\sum_{i=1}^n 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^n 1\{c^{(i)} = j\}}.$$

   μ_j = cluster means/"centers"

   }

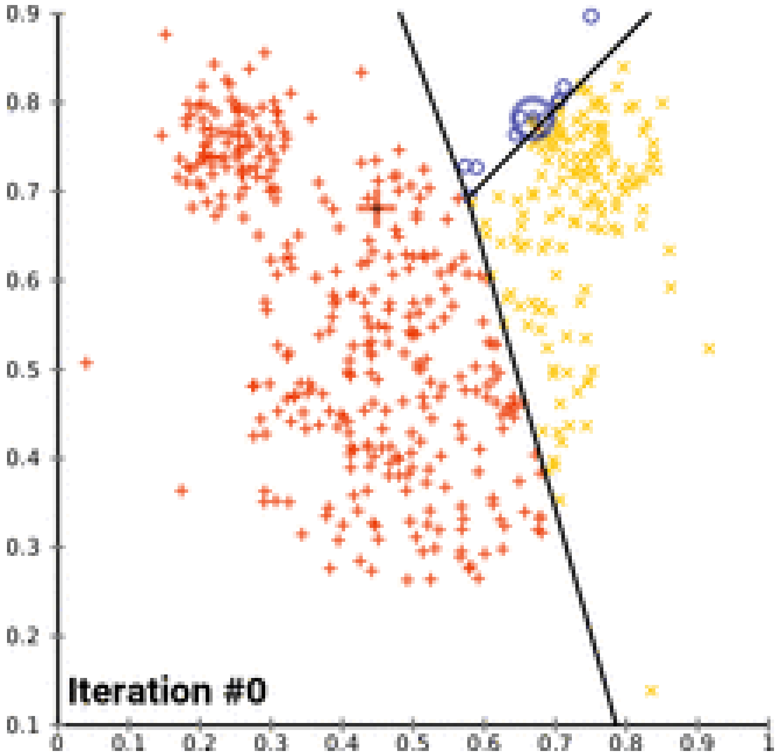
Adapted from Andrew Ng, Stanford – [CS 229 Course Notes]

K-Means



Adapted from Andrew Ng, Stanford – [CS 229 Course Notes]()

K-Means



Mention:

- Voronoi Diagram
- Computational Geometry

from [Wikipedia]()

E-M: Expectation-Maximization

Repeat until convergence {

(E-step) For each i, set

$x^{(i)}$ = data samples

$$Q_i(z^{(i)}) := p(z^{(i)} | x^{(i)}; \theta).$$

Q_i = cluster distribution

$z^{(i)}$ = soft "assignment"

θ = parameters (clusters)

(M-step) Set

number of clusters = n

$$\theta := \arg\max_\theta \sum_{i=1}^{n} \mathrm{ELBO}(x^{(i)}; Q_i, \theta)$$

ELBO = "evidence lower bound"

$$= \arg\max_\theta \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}.$$

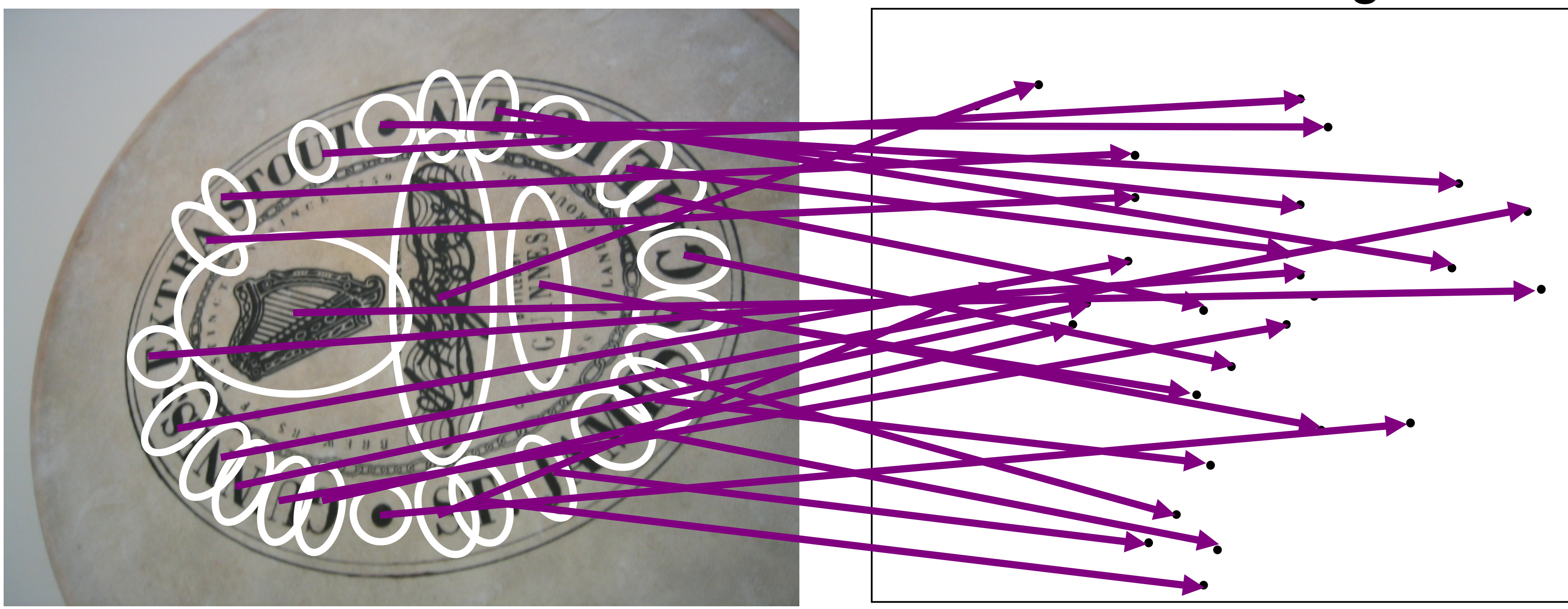
}

Adapted from Andrew Ng, Stanford – [CS 229 Course Notes]

Application: Visual Words (Vector Quantization)

Visual words: main idea

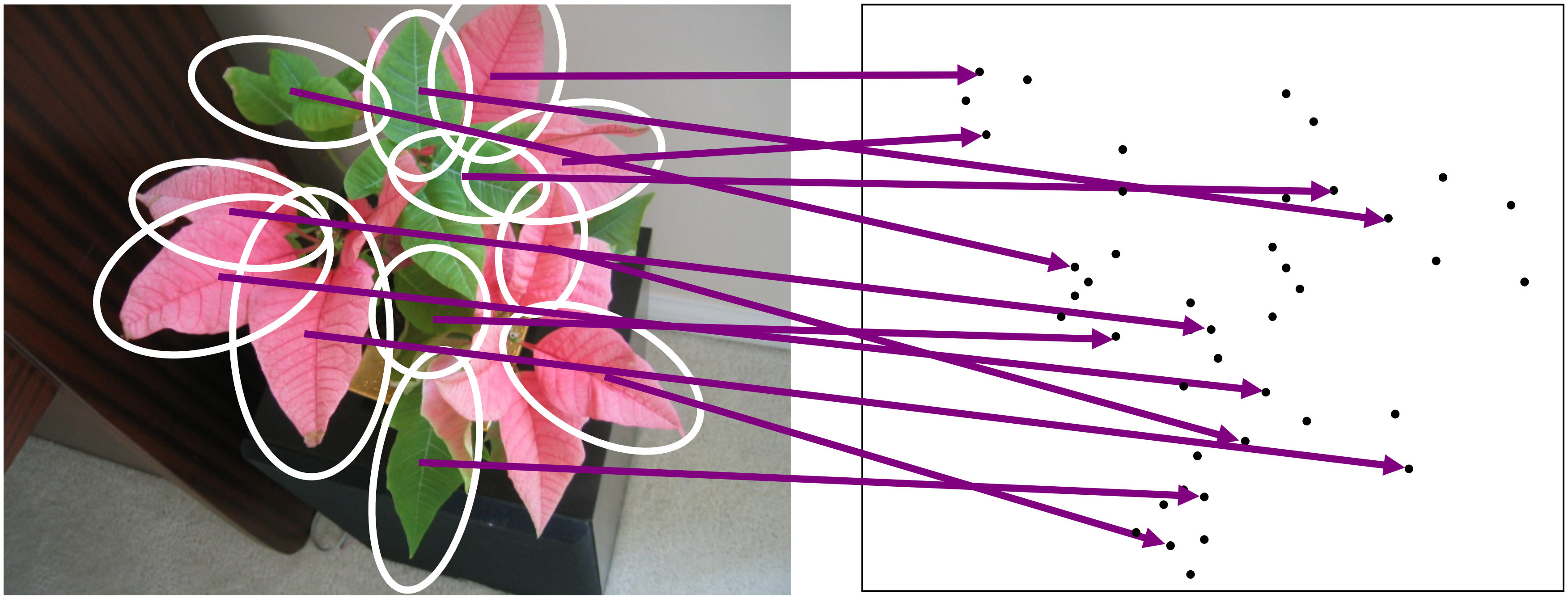
- Extract some local features from a number of images …



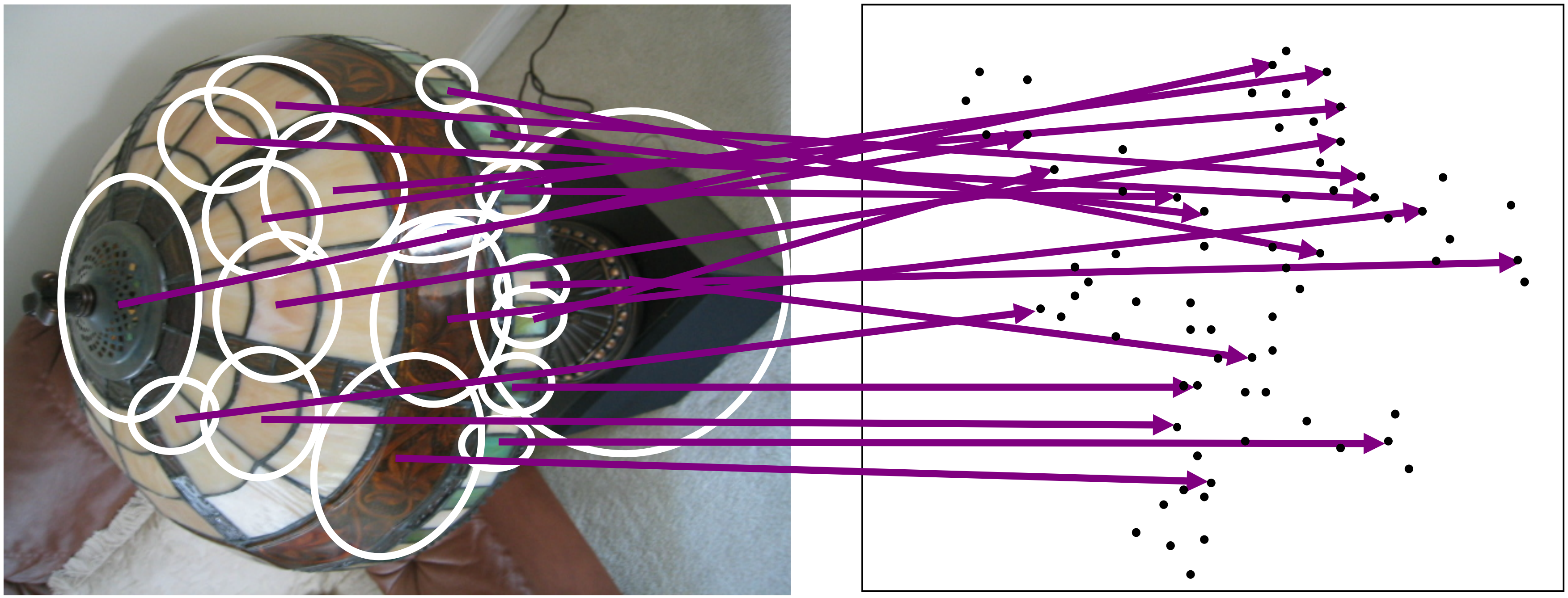
e.g., SIFT descriptor space: each point is 128-dimensional

Slide credit: D. Nister, CVPR 2006

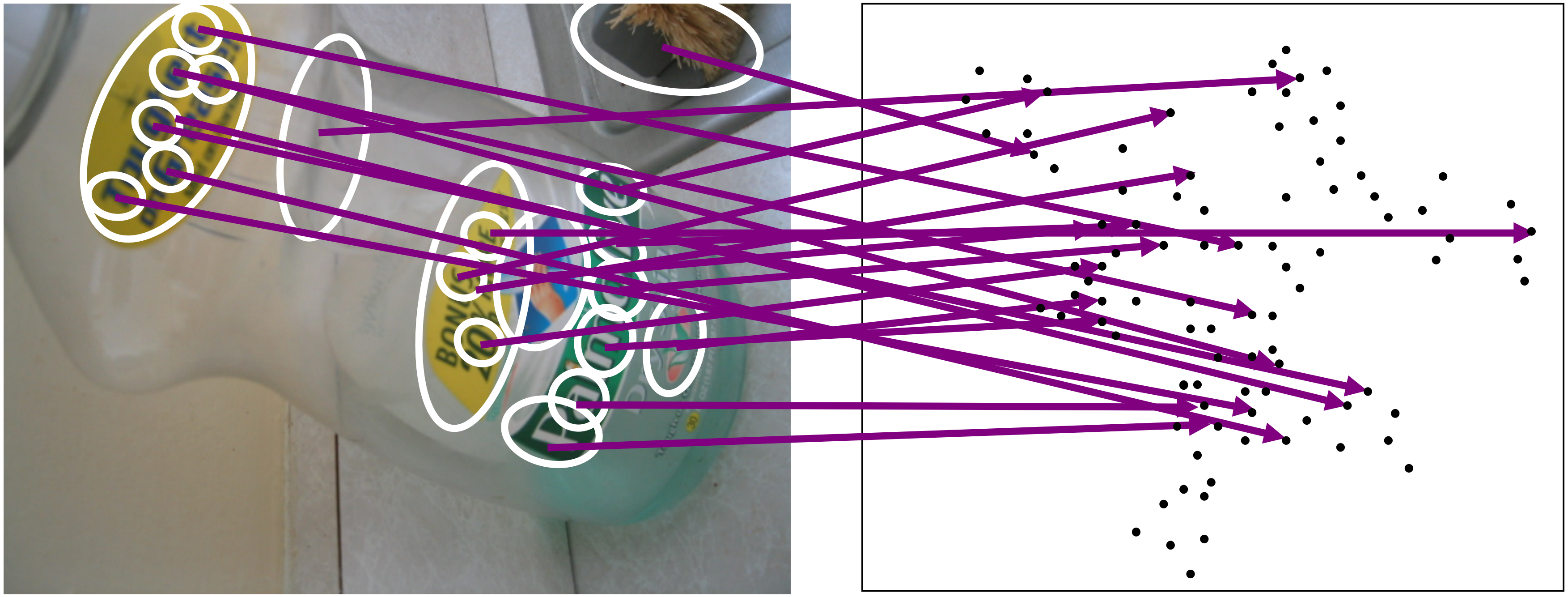
Visual words: main idea

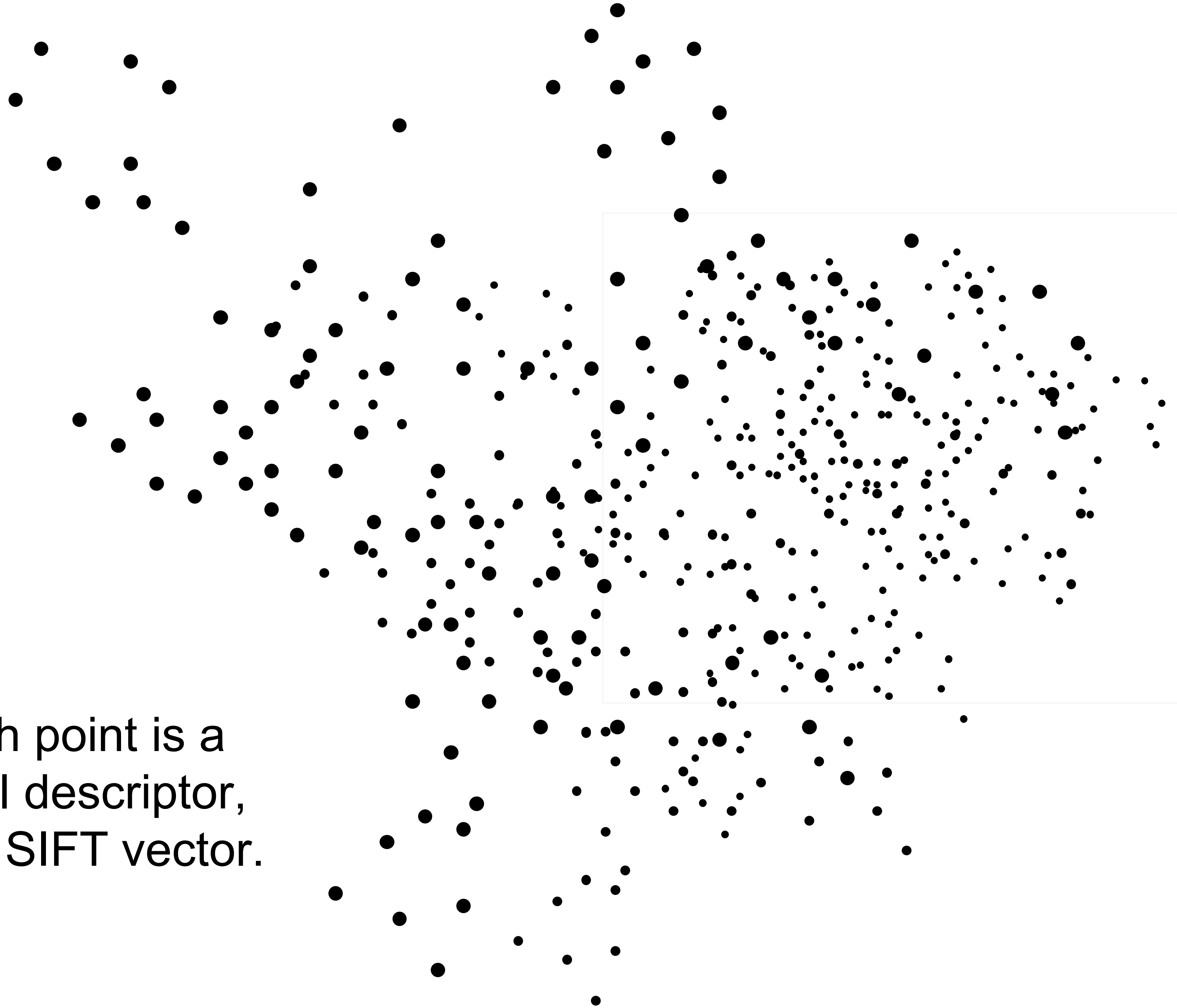


Visual words: main idea

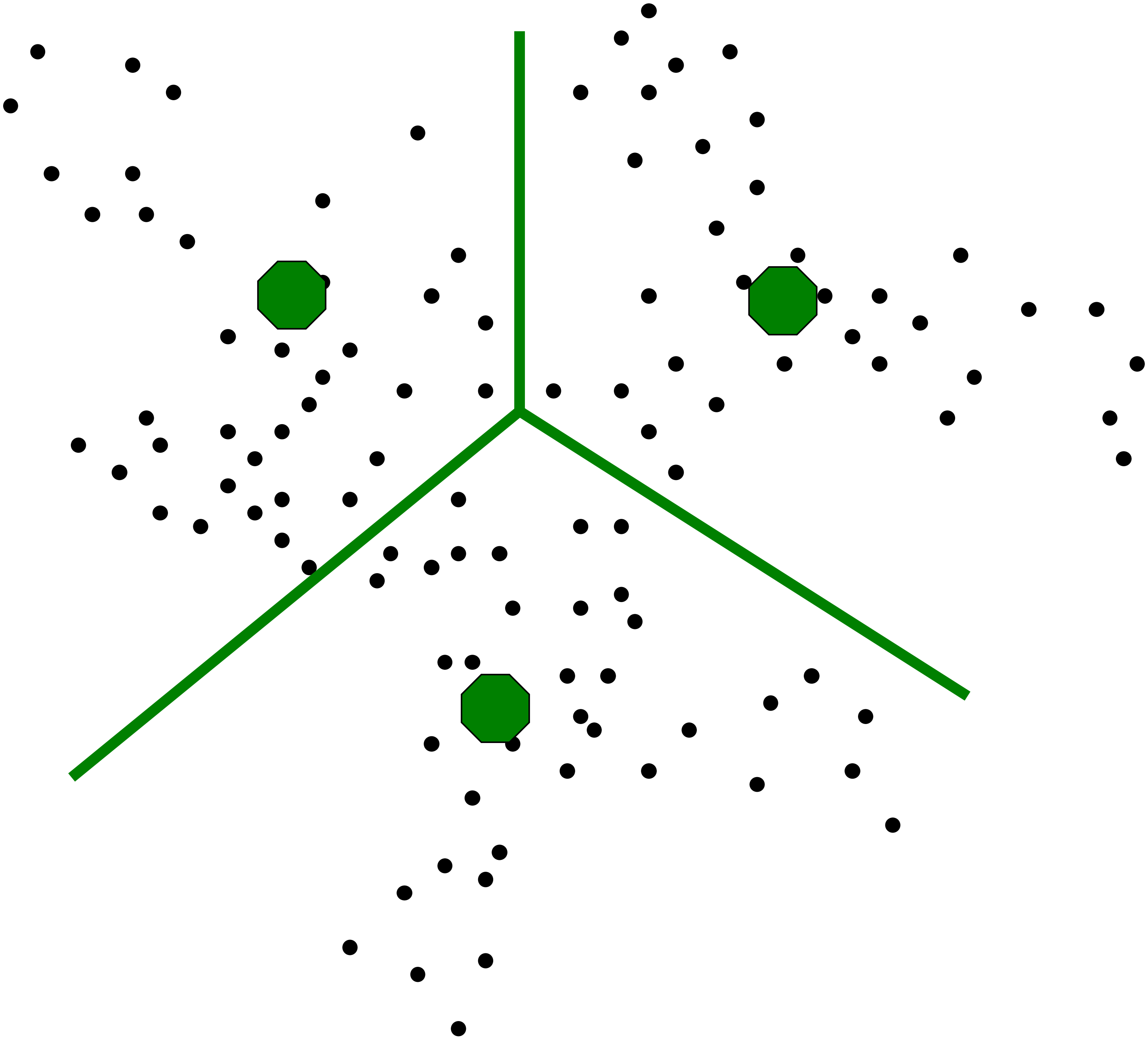


Visual words: main idea



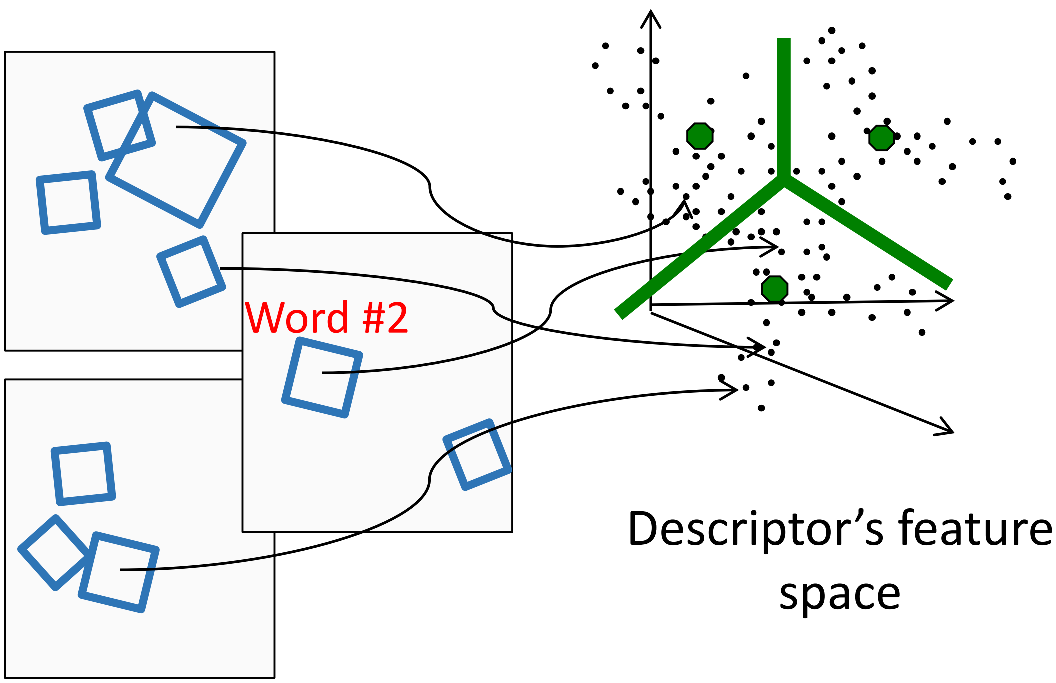


Each point is a local descriptor, e.g. SIFT vector.



Visual words

- Map high-dimensional descriptors to tokens/words by quantizing the feature space

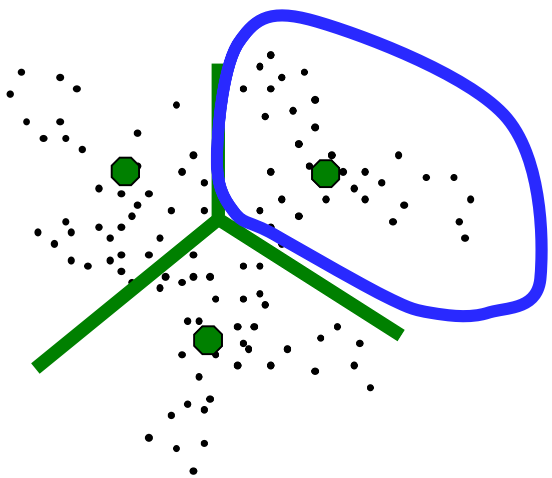


- Quantize via clustering, let cluster centers be the prototype "words"
- Determine which word to assign to each new image region by finding the closest cluster center.

Kristen Grauman

Visual words

- Example: each group of patches belongs to the same visual word



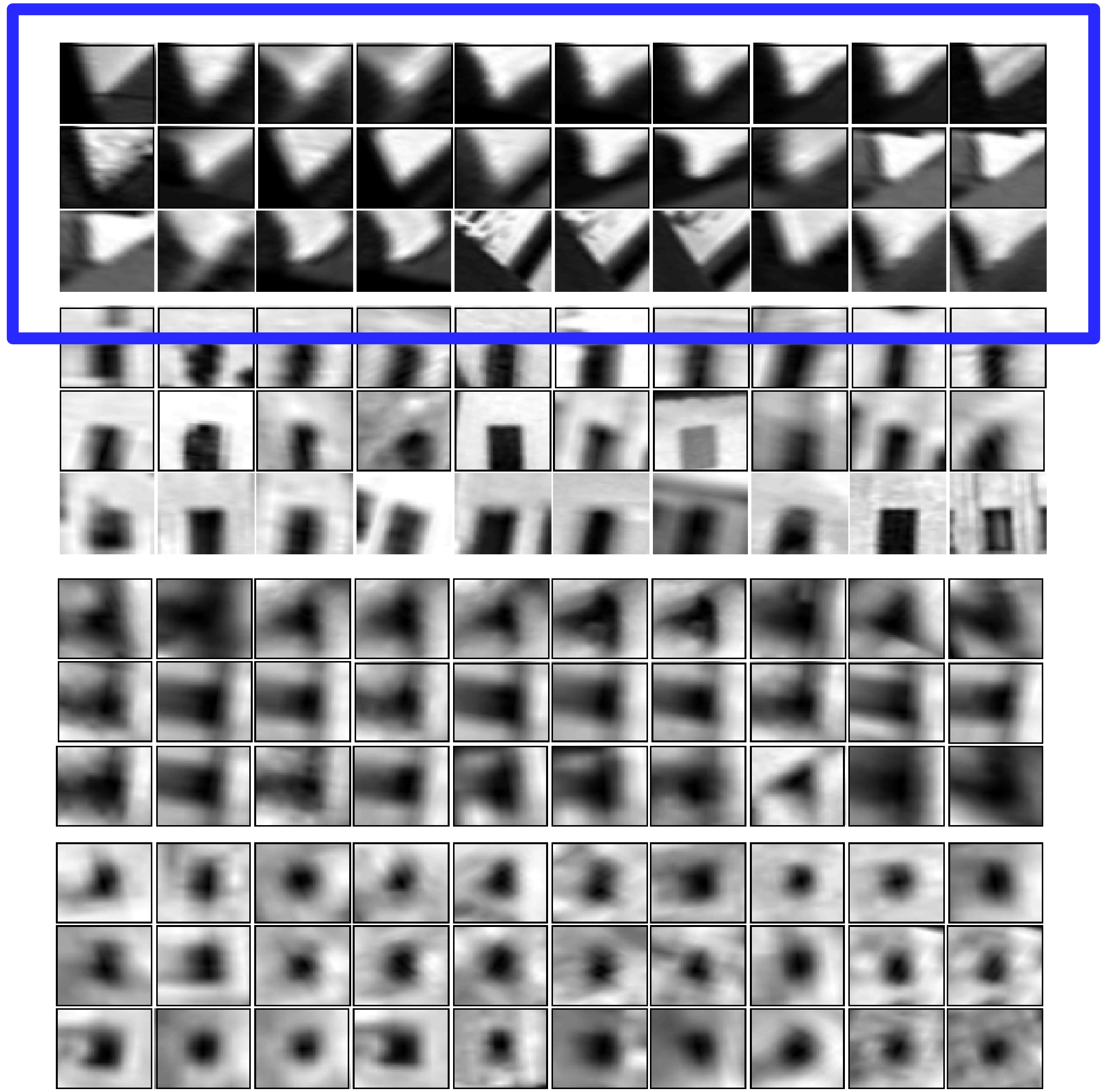
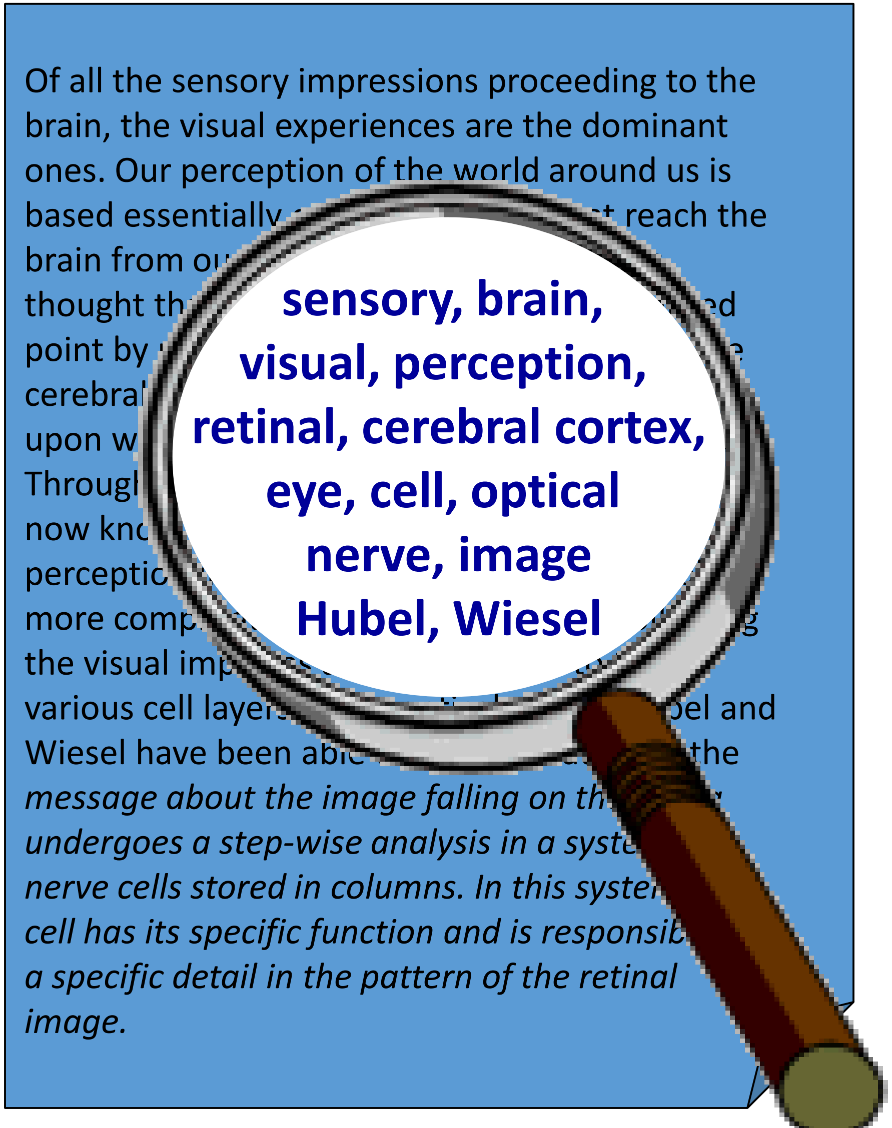
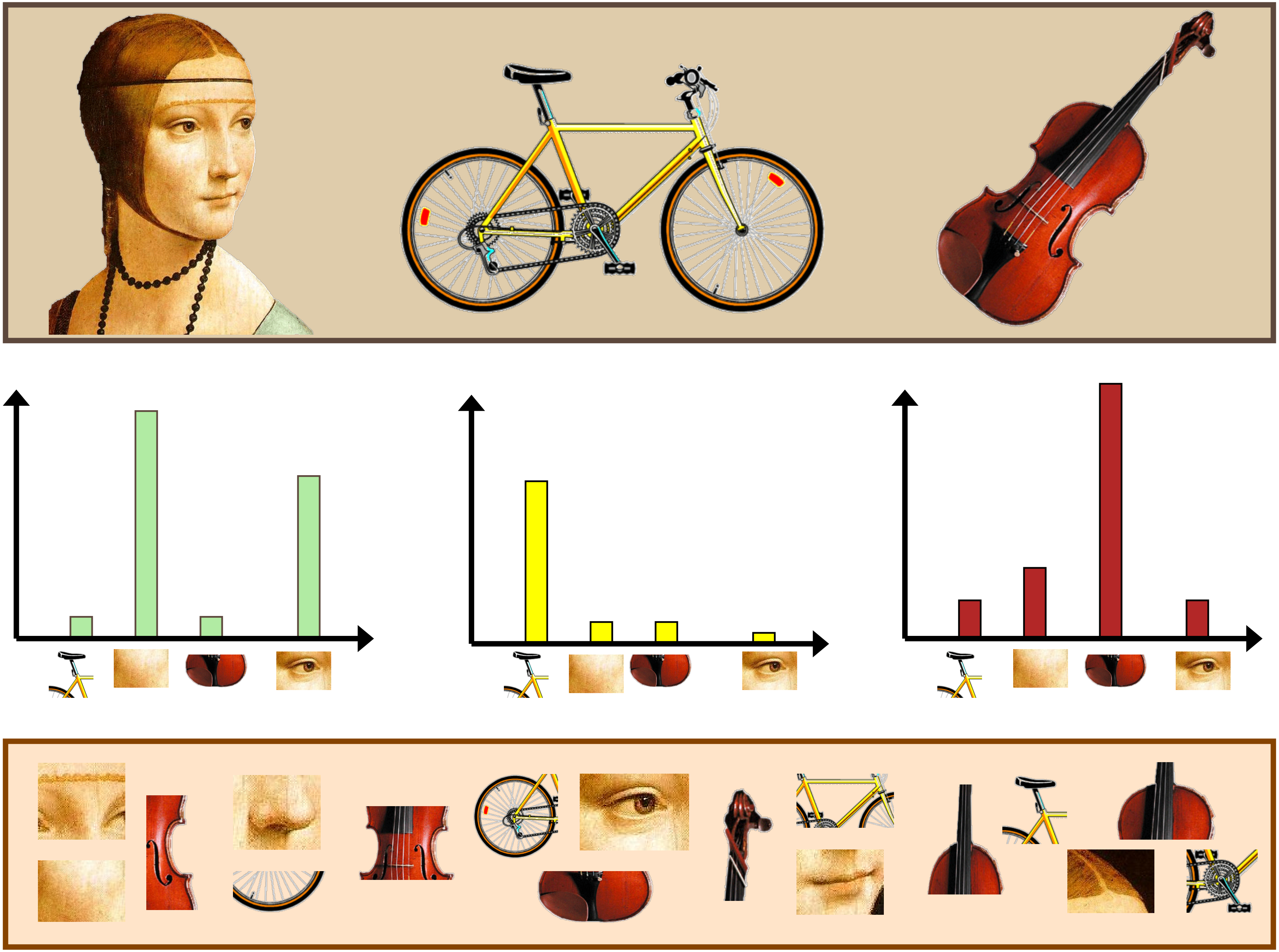


Figure from Sivic & Zisserman, ICCV 2003

Kristen Grauman

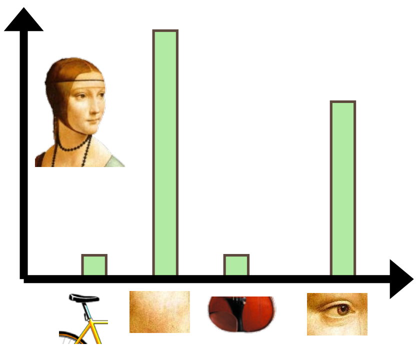
Analogy to documents

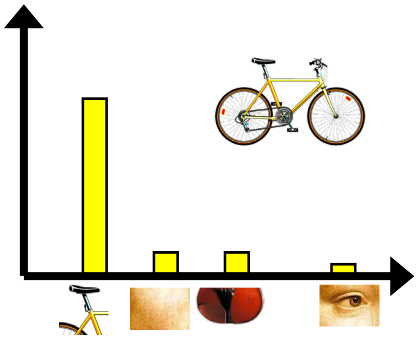


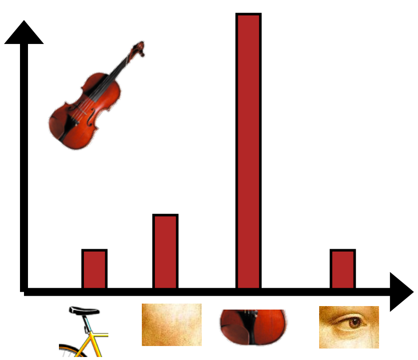


Bags of visual words

- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.



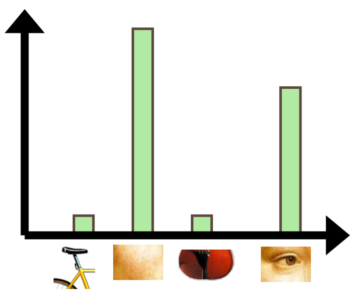




Comparing bags of words

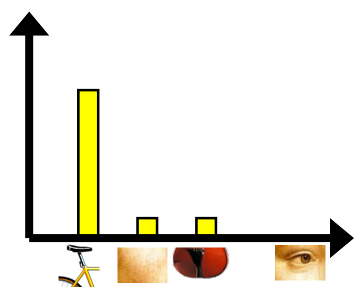
- Rank frames by normalized scalar product between their (possibly weighted) occurrence counts---*nearest neighbor* search for similar images.

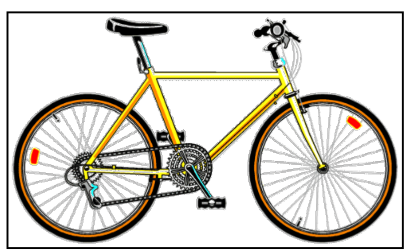
[1 8 1 4]



$\vec{d}_j$

[5 1 1 0]





$\vec{q}$

$$sim(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|}$$

$$= \frac{\sum_{i=1}^{V} d_j(i) * q(i)}{\sqrt{\sum_{i=1}^{V} d_j(i)^2} * \sqrt{\sum_{i=1}^{V} q(i)^2}}$$

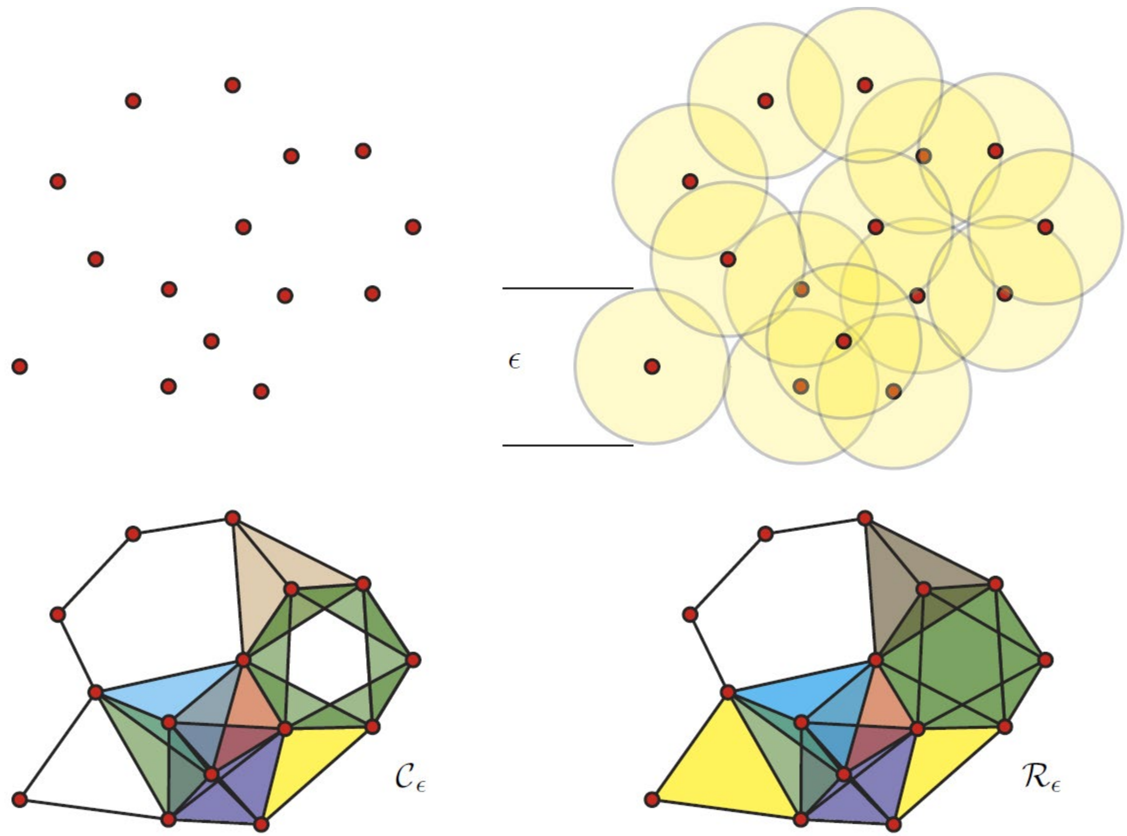
for vocabulary of *V* words

Kristen Grauman

Mean-Shift Clustering

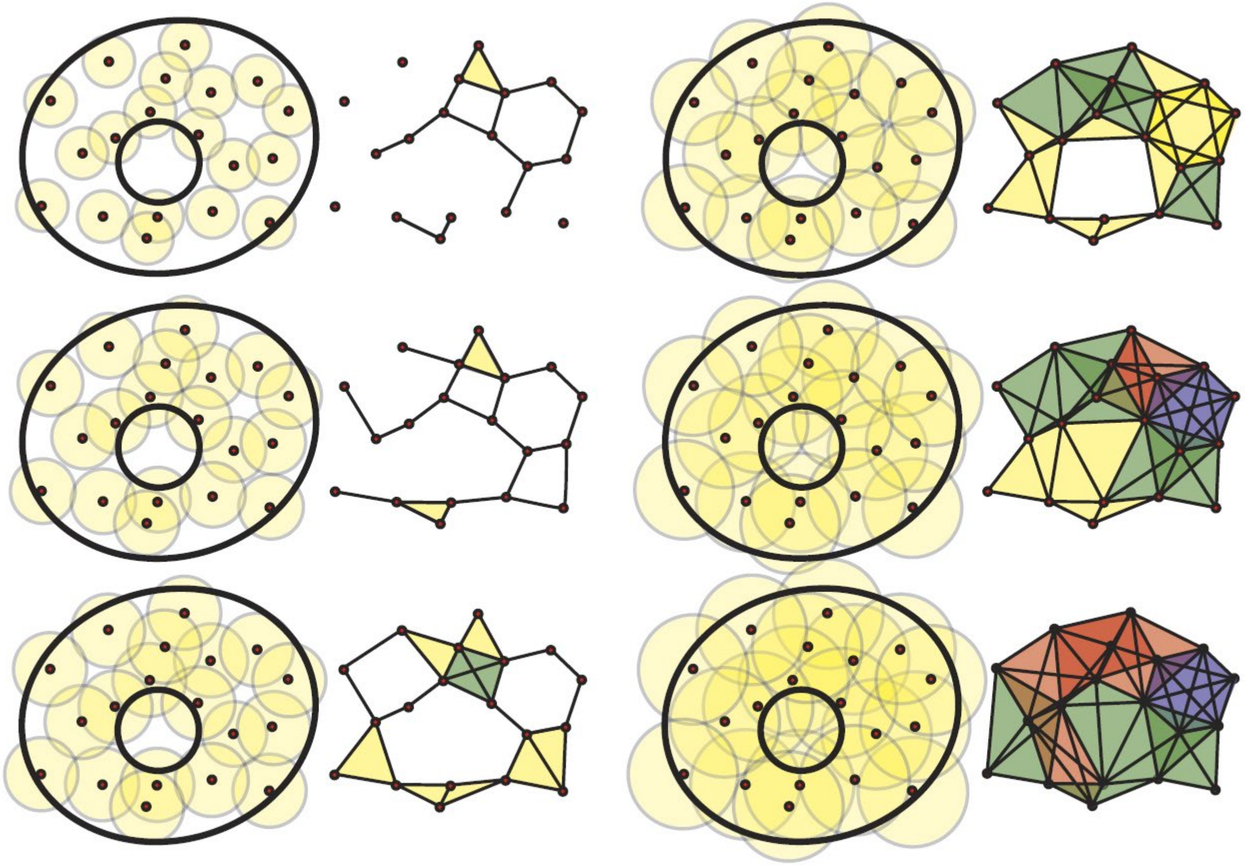
Excellent Explanation by Matt Nedrich

Persistent Homology



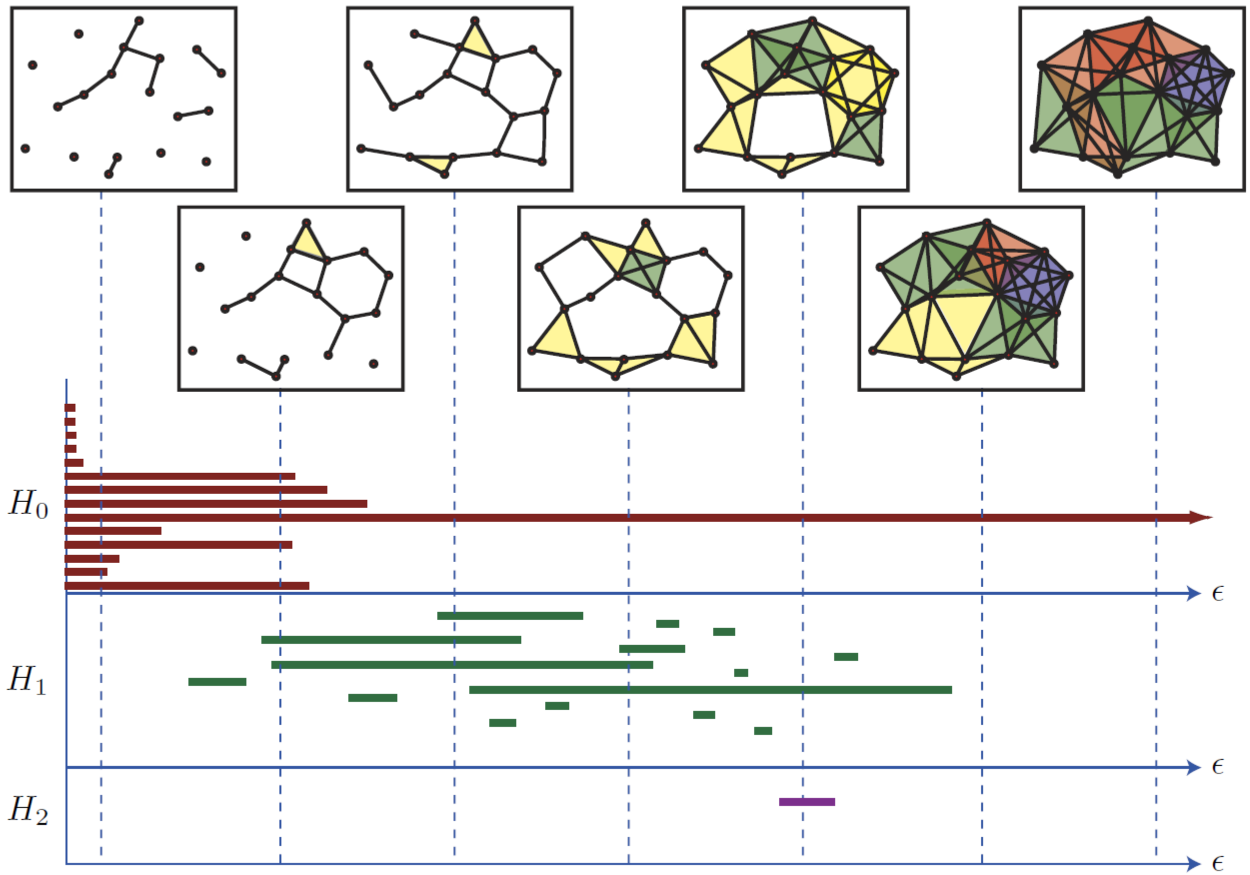
From [Robert Ghrist](#) (2008)

Persistent Homology



From Robert Ghrist (2008)

Persistent Homology



From Robert Ghrist (2008)

Metric Learning

Slides included from:

Brian Kulis, Boston University:

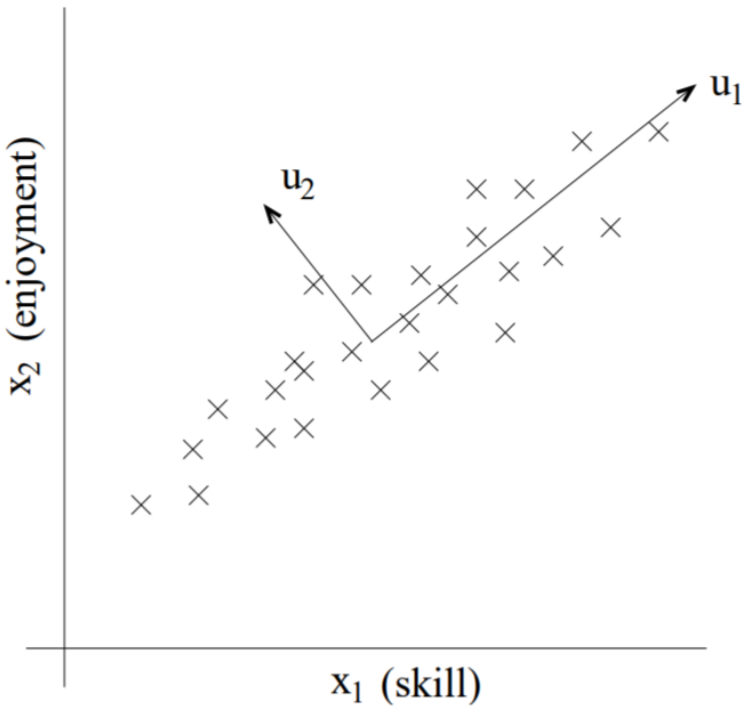
2010 ICML Tutorial / 2013 Survey

Unsupervised vs. Supervised

- Unsupervised Examples
  - Dimensionality Reduction
    - PCA
  - Density Estimation
    - KDE
  - Mahalanobis Distance

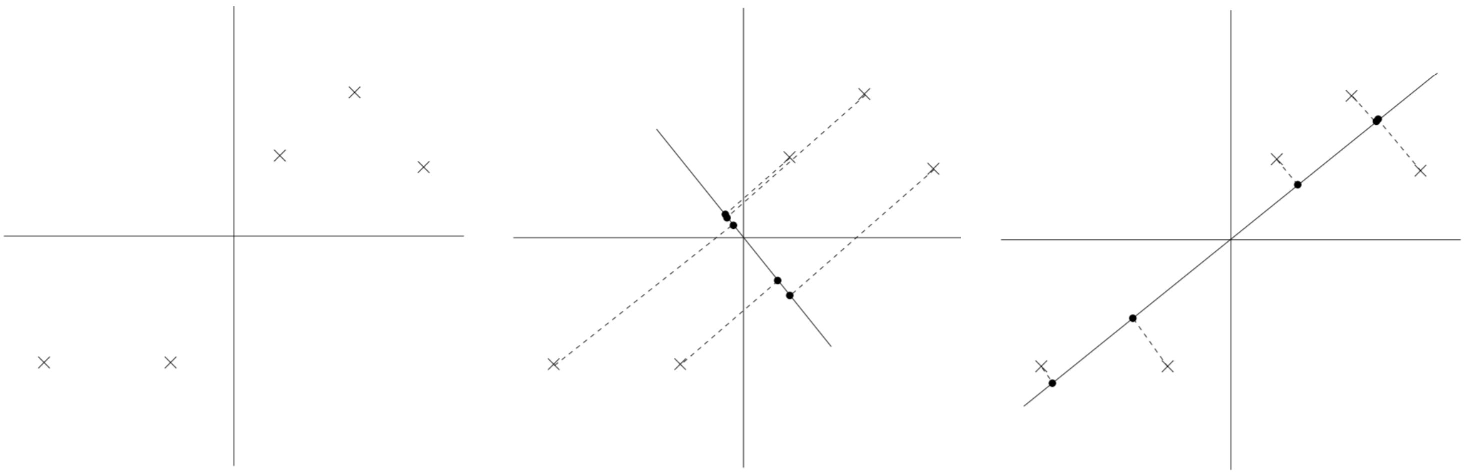
- Supervised Examples
  - LMNN (Triplet Loss)
  - Kernel Methods

PCA: Principal Component Analysis



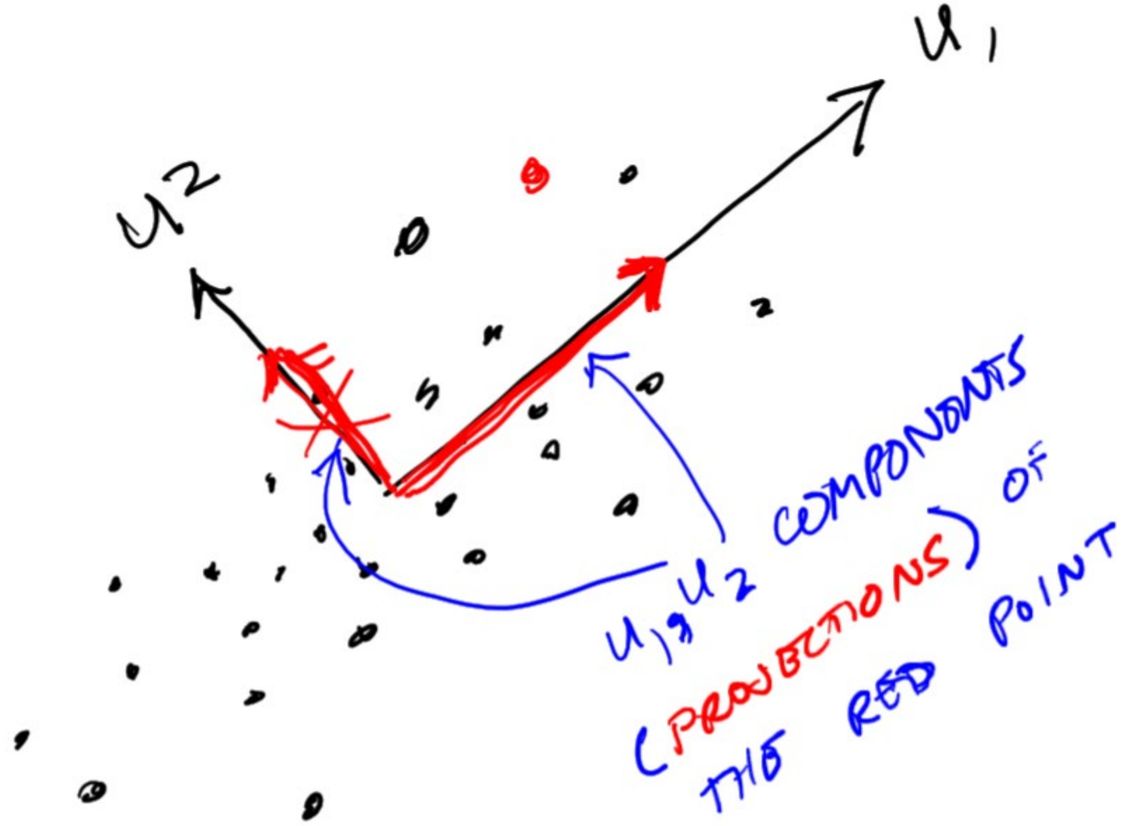
Adapted from Andrew Ng, Stanford – [CS 229 Course Notes]

PCA: Principal Component Analysis

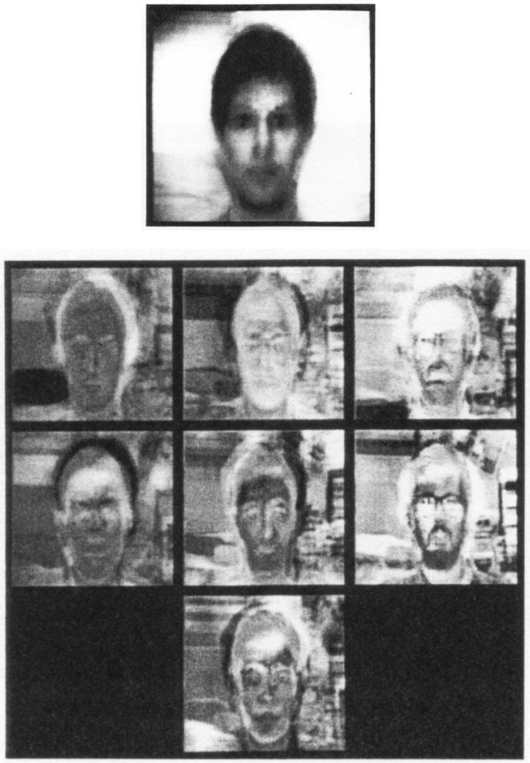


Adapted from Andrew Ng, Stanford – [CS 229 Course Notes]

(WHITEBOARD)

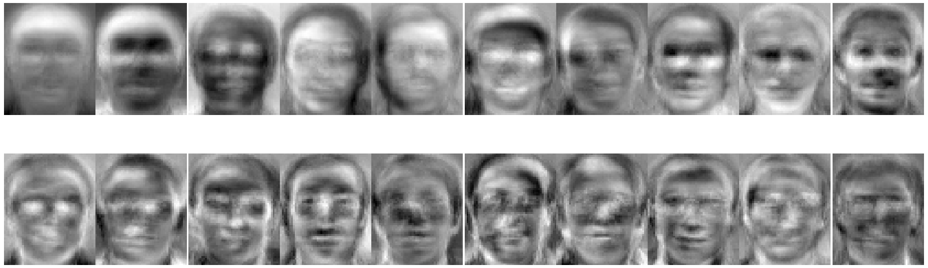


PCA Application - Eigenfaces



Turk and Pentland, Eigenfaces for Recognition. Journal of Cognitive Neuroscience, 1991

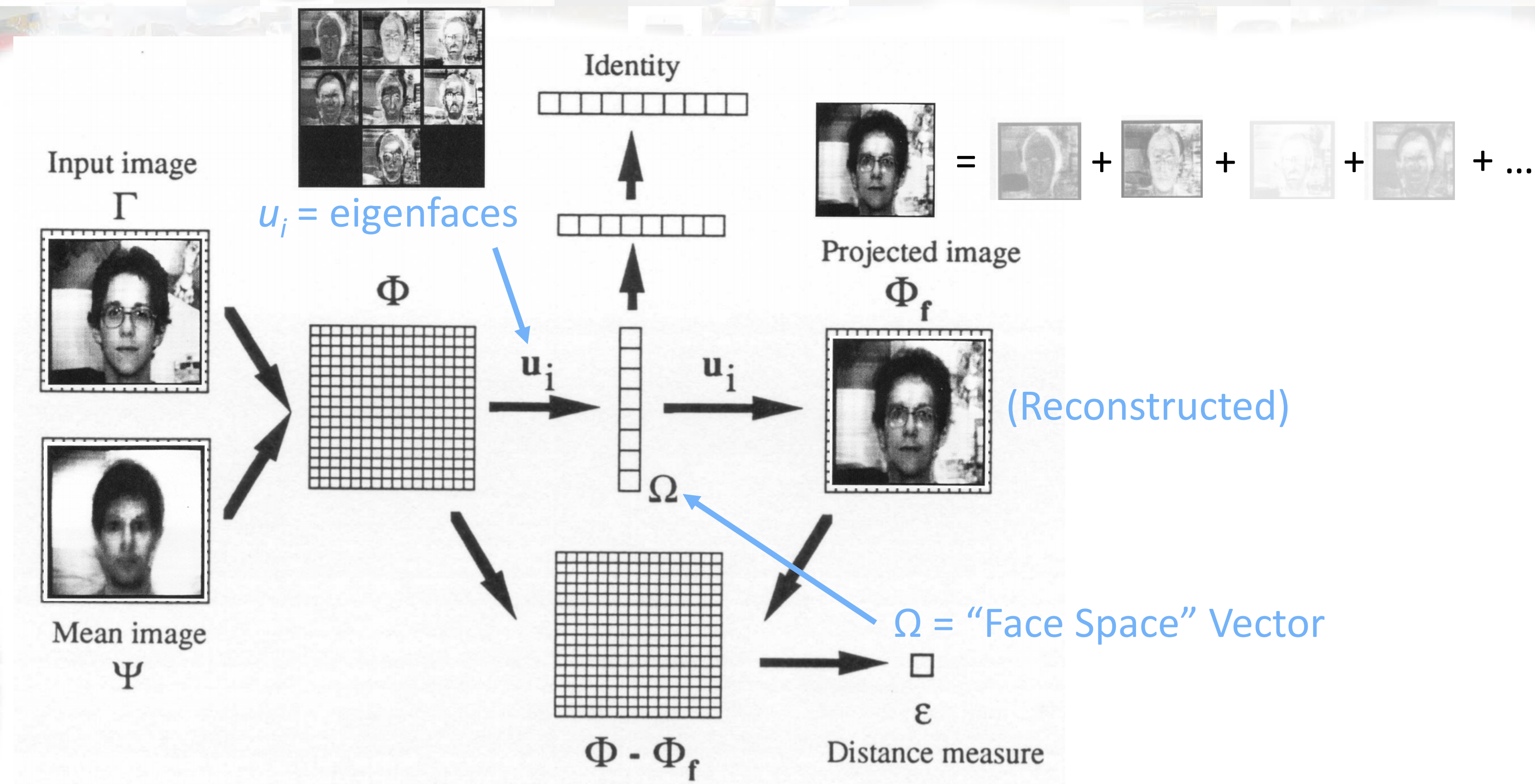
PCA Application - Eigenfaces



from [Sebastian Norena]

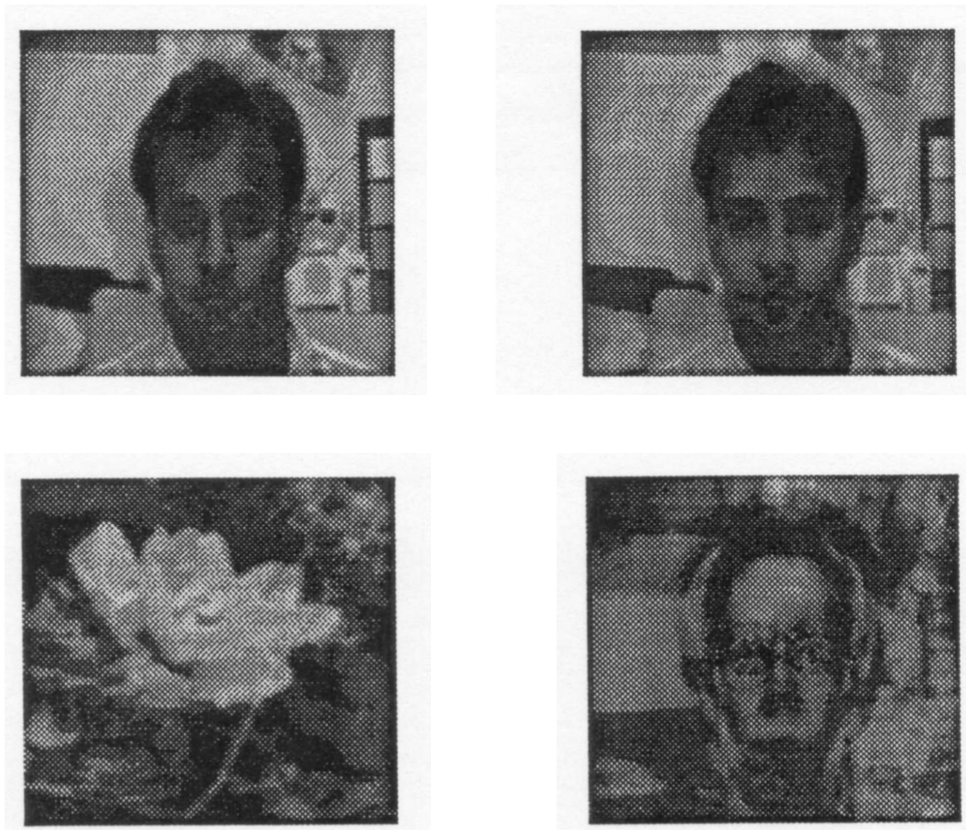
from [Deepesh Raj]

PCA Application - Eigenfaces



Turk and Pentland, Eigenfaces for Recognition. Journal of Cognitive Neuroscience, 1991

PCA Application - Eigenfaces



Turk and Pentland, Eigenfaces for Recognition. Journal of Cognitive Neuroscience, 1991

Mahalanobis Distance

ON THE GENERALIZED DISTANCE IN STATISTICS.

By P. C. MAHALANOBIS.

(*Read January 4, 1936.*)

1. A normal (Gauss-Laplacian) statistical population in P-variates is usually described by a P-dimensional frequency distribution :—

$$df = \text{const.} \times e^{-\frac{1}{2\alpha}\left[A_{11}(x_1-\alpha_1)^2 + A_{22}(x_2-\alpha_2)^2 + \ldots\ldots + 2A_{12}(x_1-\alpha_1)(x_2-\alpha_2) + \ldots\ldots\right]} \,.\, dx_1 \,.\, dx_2 \,..\, dx_P \qquad (1 \cdot 0)$$

where

$\alpha_1, \alpha_2 \ldots\ldots \alpha_P$ = the population (mean) values of the P-variates $x_1, x_2 \ldots\ldots x_P$ .. .. (1·1)

$\alpha_{ii} = \sigma_i^2$, are the respective variances .. .. (1·2)

$\alpha_{ij} = \sigma_i \,.\, \sigma_j \,.\, \rho_{ij}$, where ρ_{ij} = the coefficient of correlation between the ith and jth variates .. .. (1·3)

α is the determinant $|\alpha_{ij}|$ defined more fully in (2·2), and A_{ij} is the minor of α_{ij} in this determinant.

A P-variate normal population is thus completely specified by the set of

Mahalanobis Distance

- Assume the data is represented as N vectors of length d: $X = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$
- Squared Euclidean distance

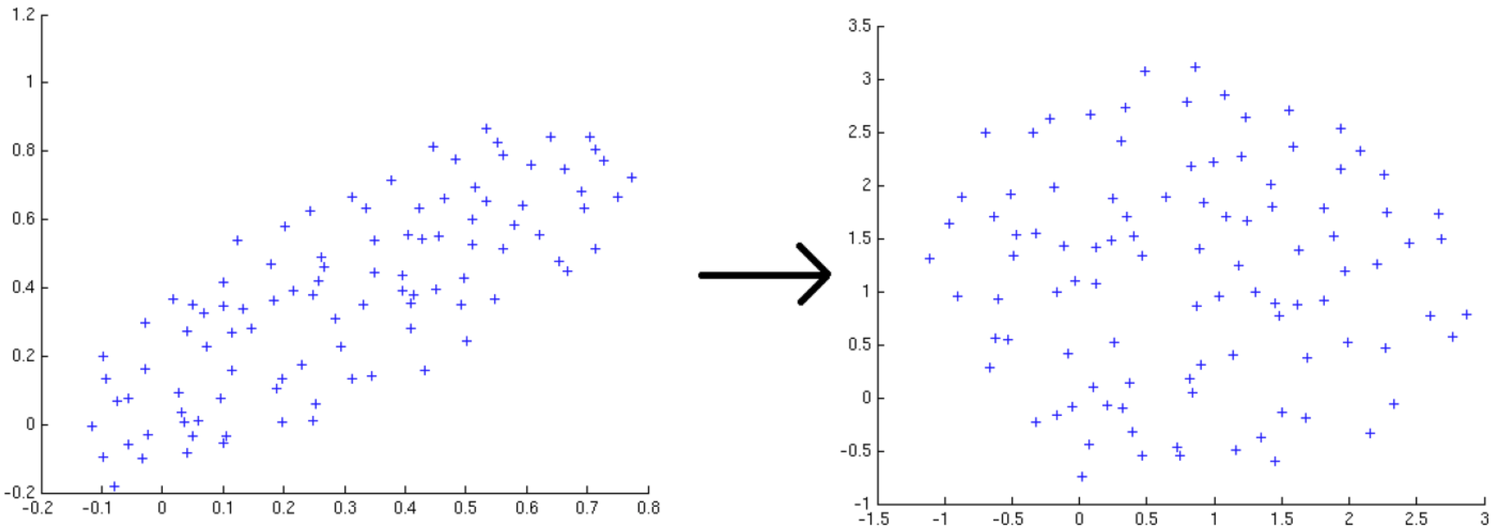
$$\begin{aligned} d(\mathbf{x}_1, \mathbf{x}_2) &= \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \\ &= (\mathbf{x}_1 - \mathbf{x}_2)^T(\mathbf{x}_1 - \mathbf{x}_2) \end{aligned}$$

- Let $\Sigma = \sum_{i,j}(\mathbf{x}_i - \mu)(\mathbf{x}_j - \mu)^T$
- The "original" Mahalanobis distance:

$$d_M(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 - \mathbf{x}_2)^T \Sigma^{-1} (\mathbf{x}_1 - \mathbf{x}_2)$$

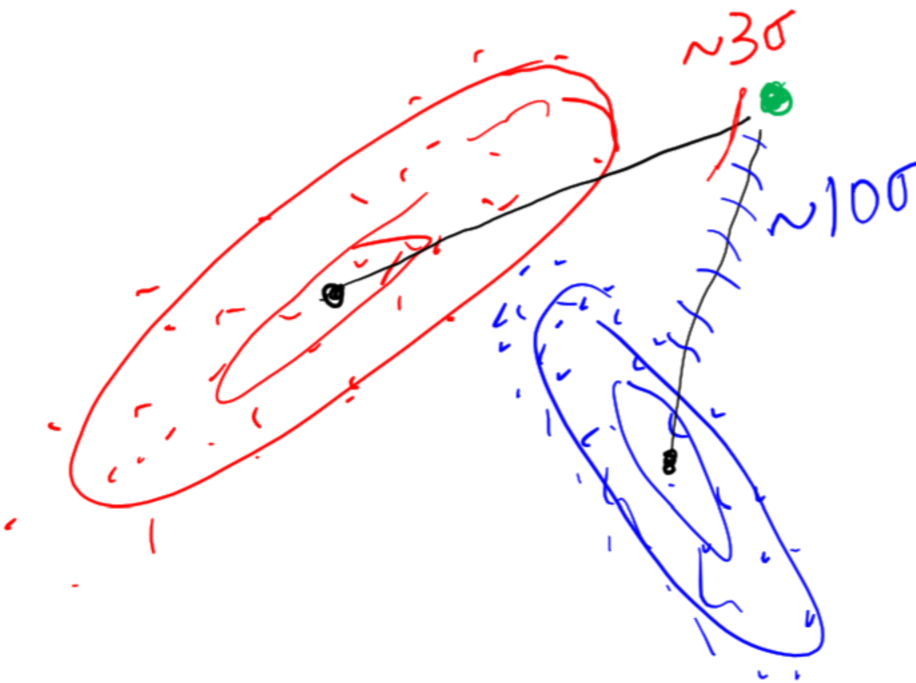
Mahalanobis Distance

- Equivalent to applying a *whitening transform*



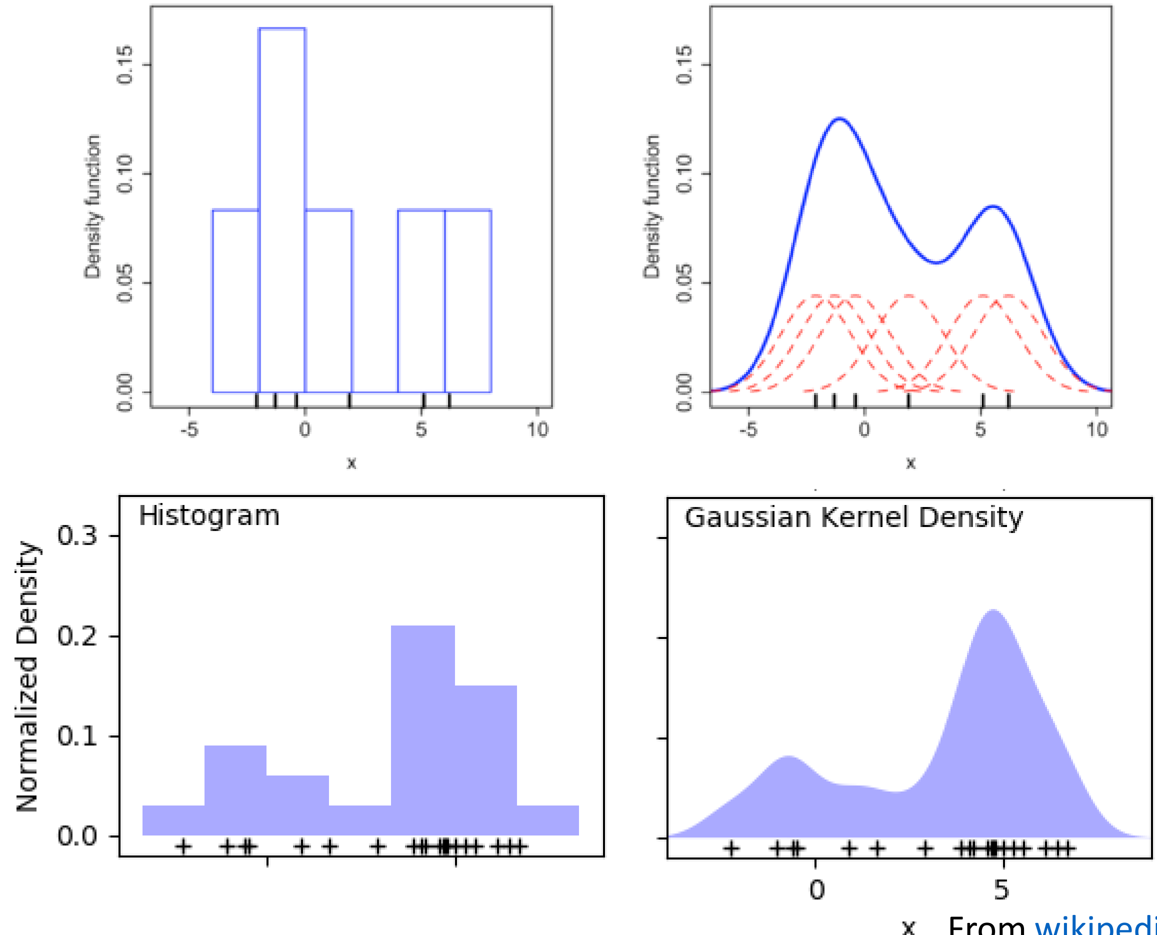
(WHITEBOARD)

MAHALANOBIS DISTANCE



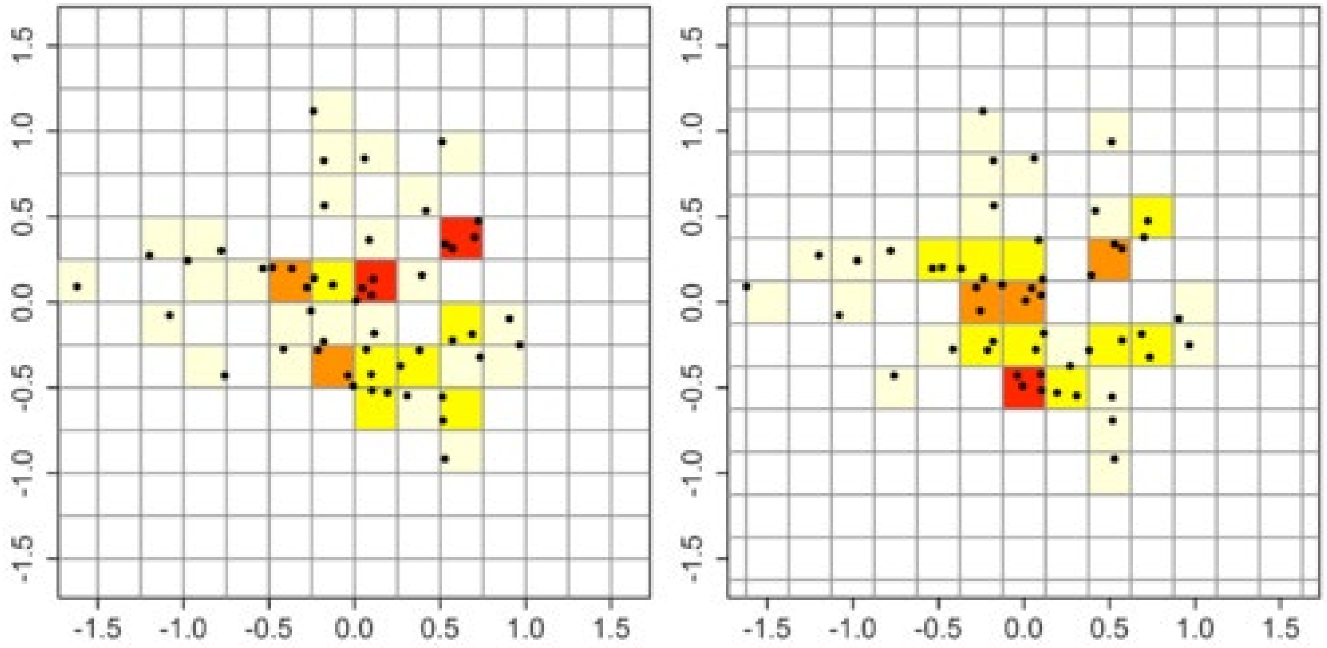
SAME EUCLIDEAN DIST. FROM MEANS BUT QUITE DIFF'T RELATIVE TO COVARIANCE

KDE: Kernel Density Estimation



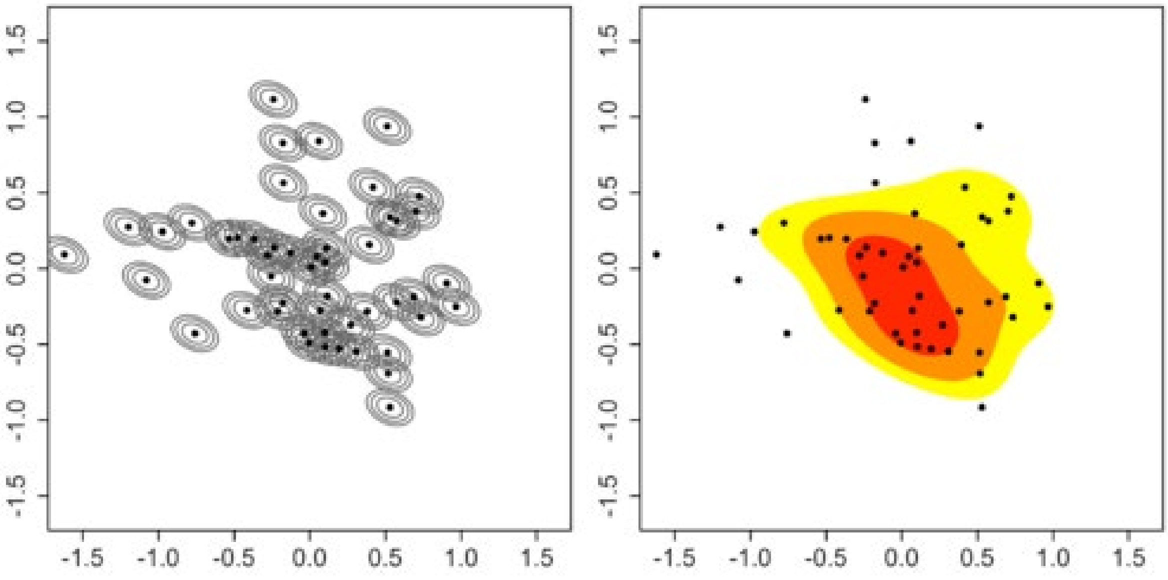
From [wikipedia]() and [scikit-learn.org]()

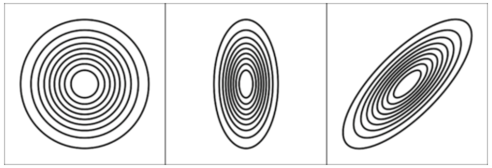
KDE: Kernel Density Estimation



From Wikipedia

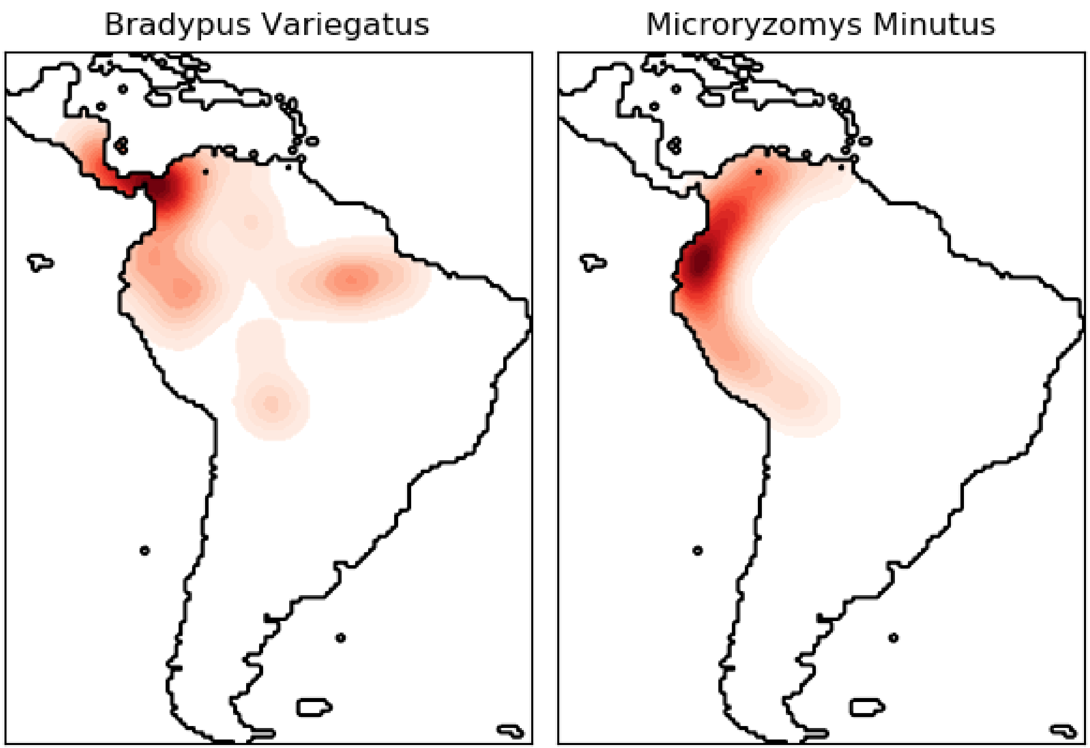
KDE: Kernel Density Estimation





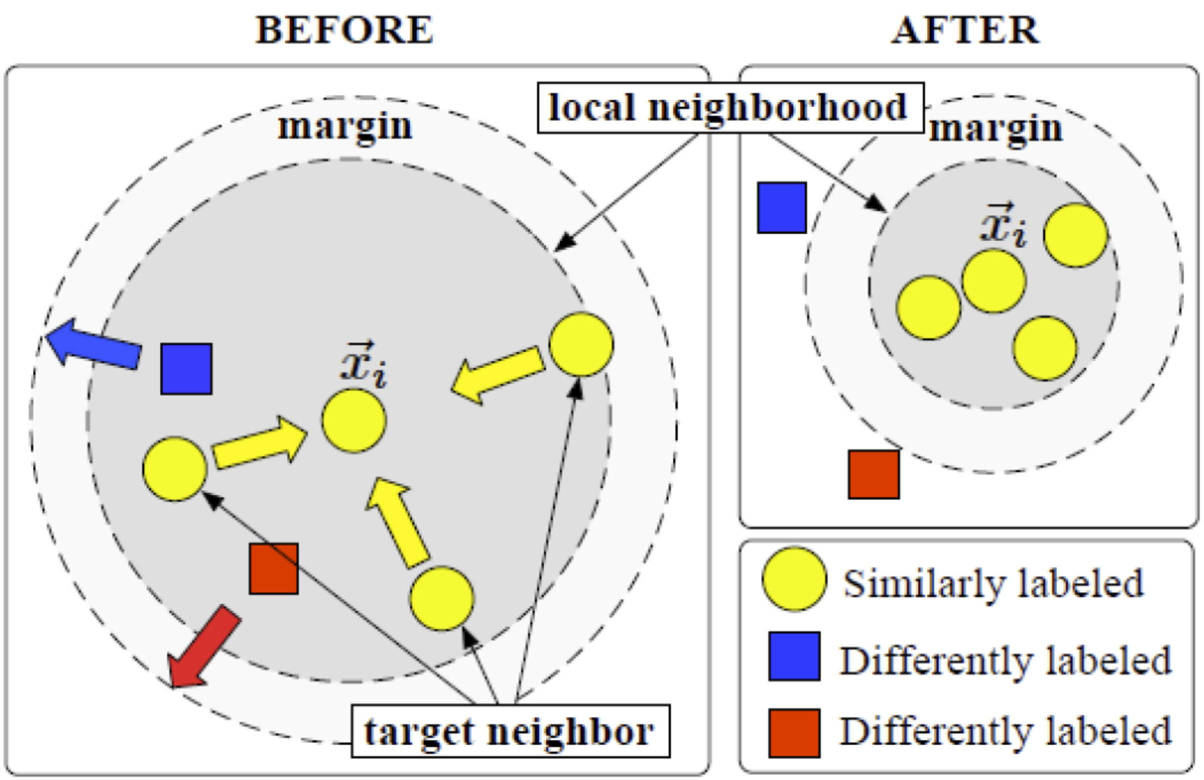
From Wikipedia

KDE: Kernel Density Estimation



From wikipedia and scikit-learn.org

LMNN: Large-Margin Nearest-Neighbor



LMNN: Large-Margin Nearest-Neighbor

- Problem Formulation
  - Also define set $\mathcal{S}$ of pairs of points $(\mathbf{x}_i, \mathbf{x}_j)$ such that $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighbors in the same class
  - Want to minimize sum of distances of pairs of points in $\mathcal{S}$
  - Also want to satisfy the relative distance constraints
- Mathematically:

$$\begin{aligned} \min_{A} \quad & \textstyle\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} d_A(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & d_A(\mathbf{x}_i, \mathbf{x}_k) - d_A(\mathbf{x}_i, \mathbf{x}_j) \geq 1 \quad \forall(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) \in \mathcal{R} \\ & A \succeq 0. \end{aligned}$$