

Machine Learning for Data Science (CS4786)

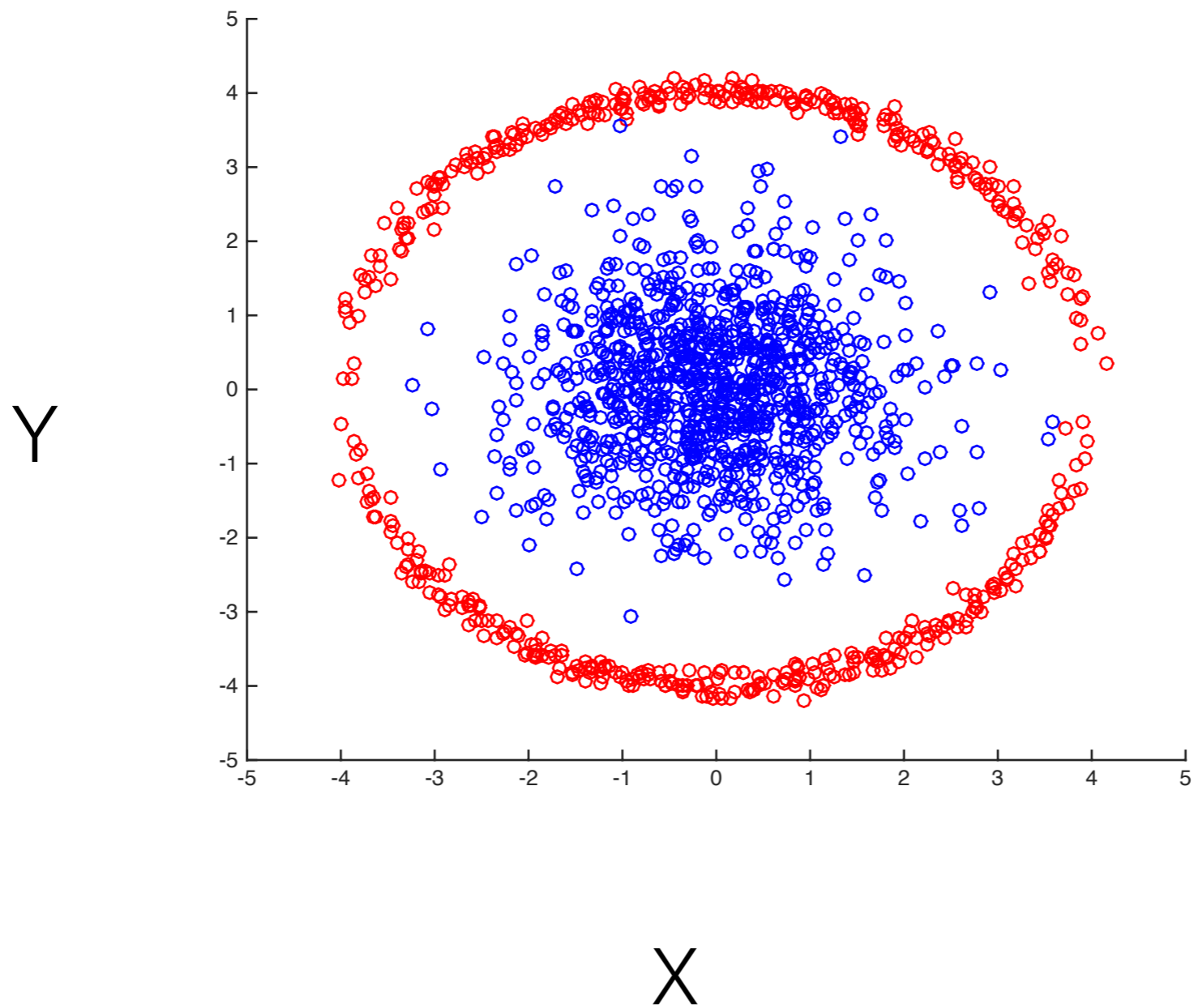
Lecture 7

Kernel PCA, Clustering

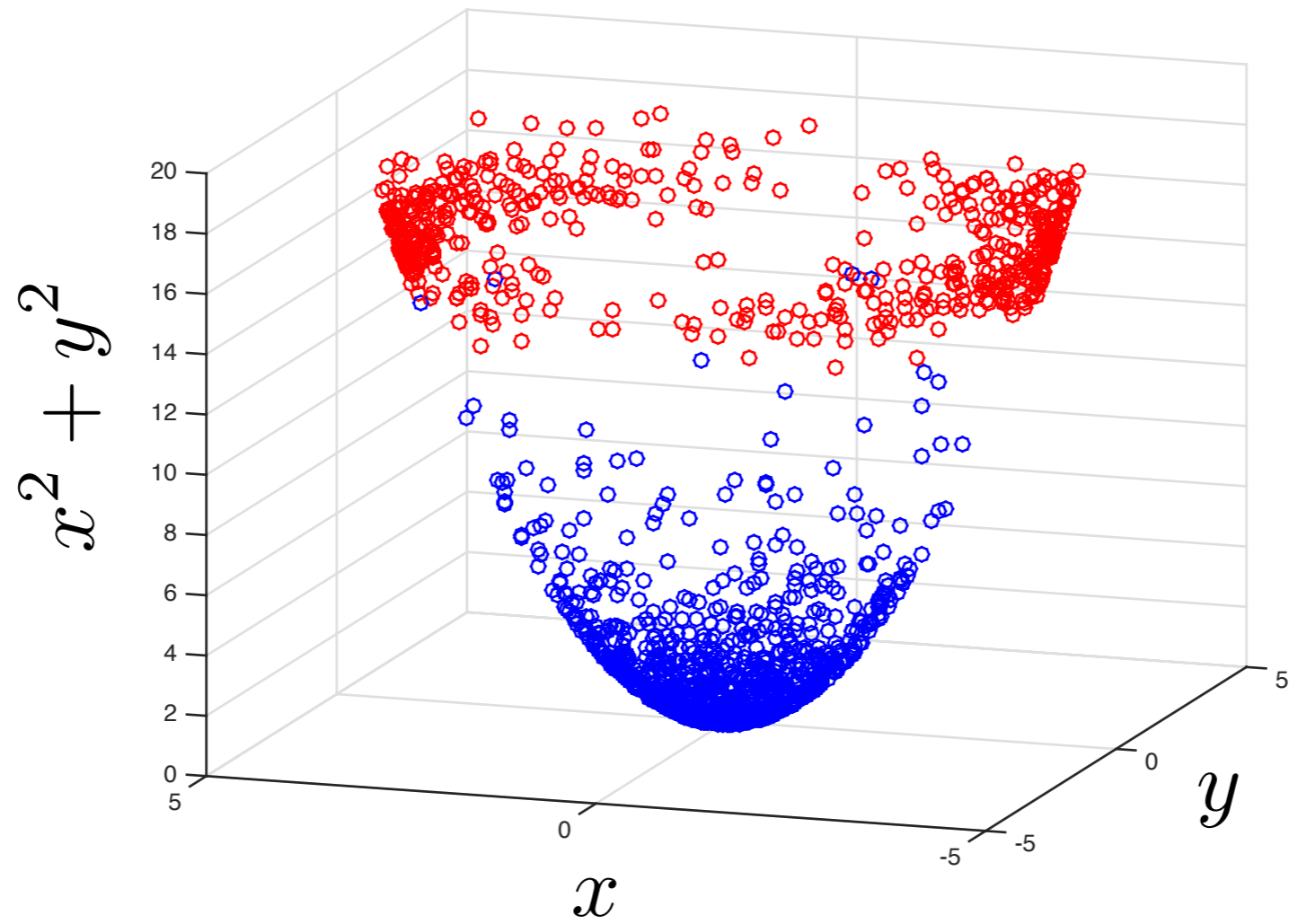
Course Webpage :

<http://www.cs.cornell.edu/Courses/cs4786/2016fa/>

EXAMPLE



EXAMPLE



KERNEL TRICK

- Essence of Kernel trick:
 - If we can write down an algorithm only in terms of $\Phi(\mathbf{x}_t)^\top \Phi(\mathbf{x}_s)$ for data points \mathbf{x}_t and \mathbf{x}_s
 - Then we don't need to explicitly enumerate $\Phi(\mathbf{x}_t)$'s but instead, compute $k(\mathbf{x}_t, \mathbf{x}_s) = \Phi(\mathbf{x}_t)^\top \Phi(\mathbf{x}_s)$ (even if Φ maps to infinite dimensional space)
- Example: RBF kernel $k(\mathbf{x}_t, \mathbf{x}_s) = \exp(-\sigma \|\mathbf{x}_t - \mathbf{x}_s\|_2^2)$, polynomial kernel $k(\mathbf{x}_t, \mathbf{x}_s) = (\mathbf{x}_t^\top \mathbf{y}_t)^p$
- Kernel function measures similarity between points.

LETS REWRITE PCA

- k^{th} column of W is eigenvector of covariance matrix
That is, $\lambda_k W_k = \Sigma W_k$. Rewriting, for centered X

$$\lambda_k W_k = \frac{1}{n} \left(\sum_{t=1}^n \mathbf{x}_t \mathbf{x}_t^{\top} \right) W_k = \frac{1}{n} \sum_{t=1}^n (\mathbf{x}_t^{\top} W_k) \mathbf{x}_t$$

W_k 's can be written as linear combination of \mathbf{x}_t 's, as

$$W_k = \sum_{t=1}^n \alpha_k[t] \mathbf{x}_t$$

where $\alpha_k[t] = \frac{1}{\lambda_k n} (\mathbf{x}_t^{\top} W_k)$

LETS REWRITE PCA

- We dont want to compute W_k itself as it could be in feature space (which is possibly infinite dimensional)!
- However, the projections are in K dimensions and we can hope to directly compute these as:

$$y_i[k] = \mathbf{x}_i^\top W_k = \mathbf{x}_i^\top \left(\sum_{t=1}^n \alpha_k[t] \mathbf{x}_t \right) = \sum_{t=1}^n \alpha_k[t] \mathbf{x}_i^\top \mathbf{x}_t$$

- Hence if we had $\alpha_k[t]$'s for all $k \in [K]$ and $t \in [n]$, we can compute projection only using inner products!

LETS REWRITE PCA

- We have that $W_k = \sum_{s=1}^n \alpha_k[s] \mathbf{x}_s$ and that $\alpha_k[t] = \frac{1}{\lambda_k n} (\mathbf{x}_t^\top W_k)$.
- Hence:

$$\alpha_k[t] = \frac{1}{\lambda_k n} \left(\mathbf{x}_t^\top \left(\sum_{s=1}^n \alpha_k[s] \mathbf{x}_s \right) \right) = \frac{1}{\lambda_k n} \sum_{s=1}^n \alpha_k[s] \mathbf{x}_t^\top \mathbf{x}_s$$

- Let \tilde{K} be a matrix such that $\tilde{K}_{s,t} = \mathbf{x}_t^\top \mathbf{x}_s$. Hence, $\alpha_k[t] = \frac{1}{\lambda_k n} \alpha_k^\top \tilde{K}_t$ and

$$\alpha_k = \frac{1}{\lambda_k n} \tilde{K} \alpha_k$$

where \tilde{K}_t is the t 'th column of \tilde{K} .

- Hence α_k is in the direction of eigen vector of \tilde{K}

LETS REWRITE PCA

- Further, since W_k is unit norm,

$$1 = \|W_k\|_2^2 = \left(\sum_{t=1}^n \alpha_k[t] \mathbf{x}_t \right)^\top \left(\sum_{s=1}^n \alpha_k[s] \mathbf{x}_s \right) = \alpha_k^\top \tilde{K} \alpha_k = n \gamma_k \alpha_k^\top \alpha_k$$

Hence $\|\alpha_k\|^2 = \frac{1}{n \gamma_k}$ where γ_k is the k 'th eigen value of matrix \tilde{K}

Can we compute \tilde{K} based only on inner products?

REWRITING PCA

- We assumed centered data, what if its not,

$$\begin{aligned}\tilde{K}_{s,t} &= \left(\mathbf{x}_t - \frac{1}{n} \sum_{u=1}^n \mathbf{x}_u \right)^\top \left(\mathbf{x}_s - \frac{1}{n} \sum_{u=1}^n \mathbf{x}_u \right) \\ &= \mathbf{x}_t^\top \mathbf{x}_s - \left(\frac{1}{n} \sum_{u=1}^n \mathbf{x}_u \right)^\top \mathbf{x}_s - \left(\frac{1}{n} \sum_{u=1}^n \mathbf{x}_u \right)^\top \mathbf{x}_t \\ &\quad + \frac{1}{n^2} \left(\sum_{u=1}^n \mathbf{x}_u \right)^\top \left(\sum_{v=1}^n \mathbf{x}_v \right) \\ &= \mathbf{x}_t^\top \mathbf{x}_s - \frac{1}{n} \sum_{u=1}^n \mathbf{x}_u^\top \mathbf{x}_s - \frac{1}{n} \sum_{u=1}^n \mathbf{x}_u^\top \mathbf{x}_t + \frac{1}{n^2} \sum_{u=1}^n \sum_{v=1}^n \mathbf{x}_u^\top \mathbf{x}_v\end{aligned}$$

REWRITING PCA

- Equivalently, if **Kern** is the matrix ($\text{Kern}_{t,s} = x_t^\top x_s$),

$$\tilde{K} = \text{Kern} - \frac{(\mathbf{1}_{n \times n} \times \text{Kern})}{n} - \frac{(\text{Kern} \times \mathbf{1}_{n \times n})}{n} + \frac{(\mathbf{1}_{n \times n} \times \text{Kern} \times \mathbf{1}_{n \times n})}{n^2}$$

PCA REWRITTEN

- Compute $\tilde{K} = \text{Kern} - \mathbf{1} \text{ Kern}/n - \text{Kern} \mathbf{1}/n + \mathbf{1} \text{ Kern} \mathbf{1}/n^2$
- Compute top K eigen vectors P_1, \dots, P_K along with eigen values $\gamma_1, \dots, \gamma_K$ for the matrix \tilde{K}
- Rescale each P_k by the inverse of the square-root of corresponding eigen values ie. $\alpha_k = P_k / \sqrt{n\gamma_k}$
- Compute projections by setting

$$y_i[k] = \sum_{t=1}^n \alpha_k[t] \tilde{K}_{t,i}$$

or in other words $Y = \tilde{K} \times [\alpha_1, \dots, \alpha_K]$

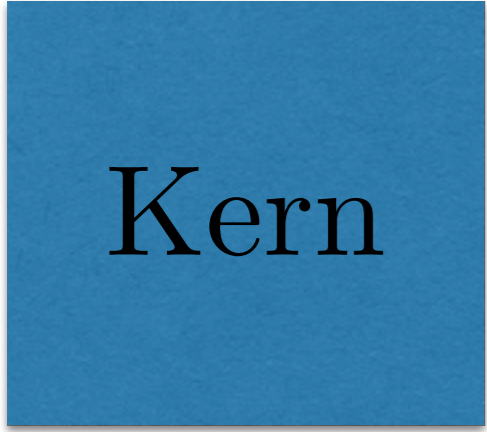
KERNEL PCA

All we need to be able to compute, to perform PCA are $\mathbf{x}_t^\top \mathbf{x}_s$

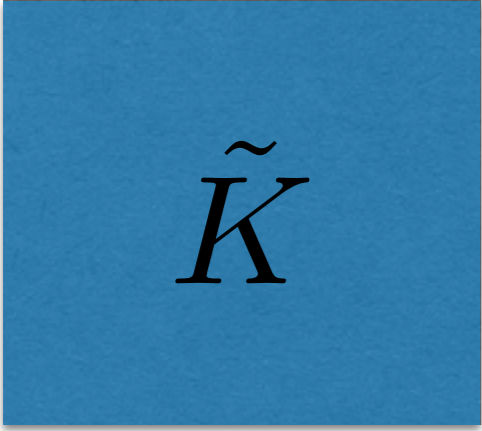
Replace $\mathbf{x}_t^\top \mathbf{x}_s$ with $\Phi(\mathbf{x}_t)^\top \Phi(\mathbf{x}_s) = k(x_t, x_s)$ to perform PCA
in feature space

KERNEL PCA

1.


$$= \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ k(x_{n-1}, x_1) & k(x_{n-1}, x_2) & \dots & k(x_{n-1}, x_n) \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix}$$

2.


$$= \text{Kern} - \frac{1}{n} (\mathbf{1} \text{ Kern} + \text{Kern} \mathbf{1}) + \frac{1}{n^2} \mathbf{1} \text{ Kern} \mathbf{1}$$

KERNEL PCA

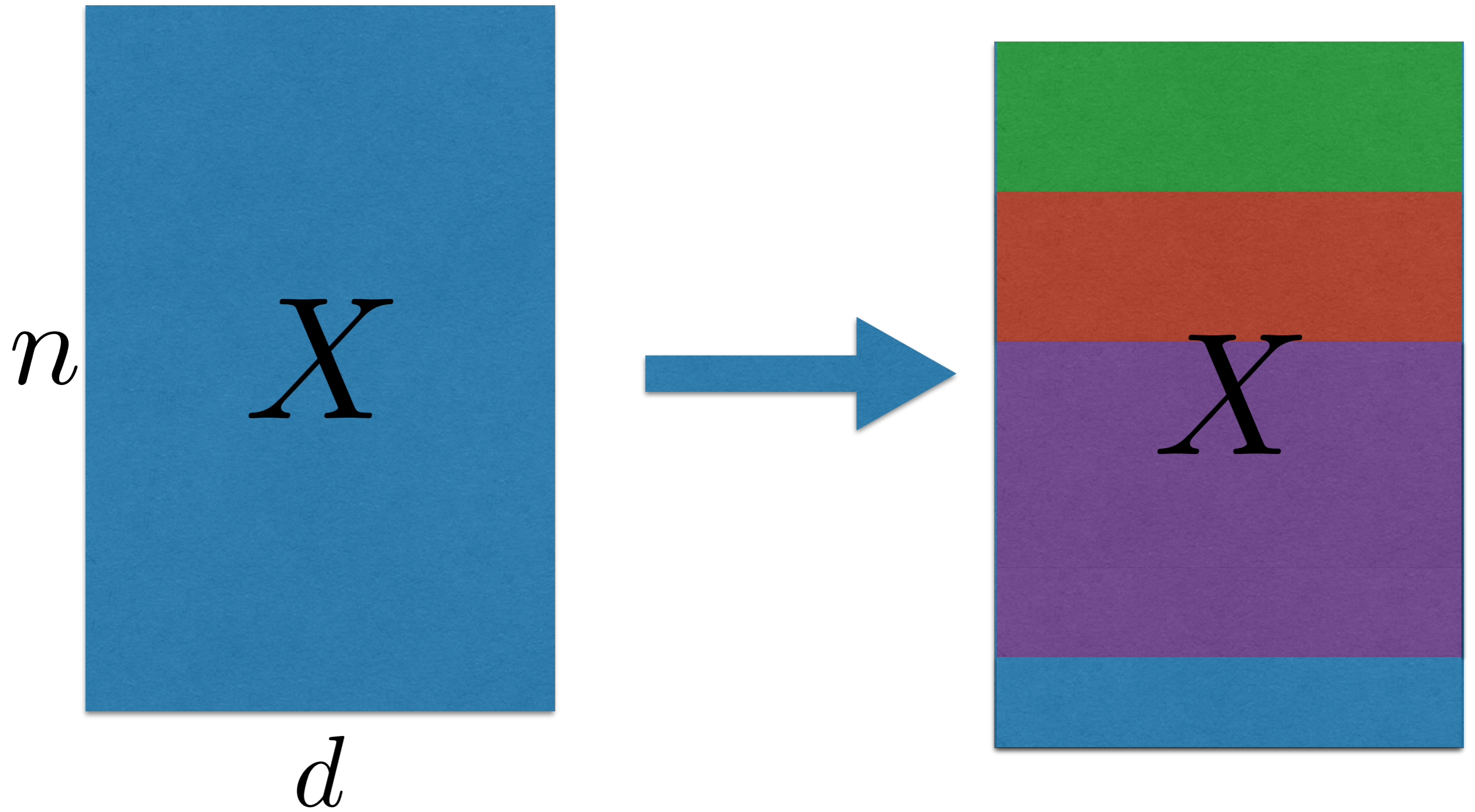
$$3. \begin{bmatrix} n \\ P \\ K \end{bmatrix}, \gamma = \text{eigs} \left(\begin{bmatrix} \tilde{K} \\ K \end{bmatrix} \right)$$

$$4. \begin{bmatrix} n \\ \alpha \\ K \end{bmatrix} = n \begin{bmatrix} P_1 & \dots & P_K \\ \hline \sqrt{n\gamma_1} & & \sqrt{n\gamma_K} \\ \hline \vdots & & \vdots \end{bmatrix}$$

$$5. \begin{bmatrix} n \\ Y \\ K \end{bmatrix} = n \begin{bmatrix} \tilde{K} \\ n \end{bmatrix} \times \begin{bmatrix} \alpha \\ K \end{bmatrix}$$

Demo

CLUSTERING



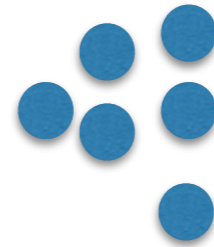
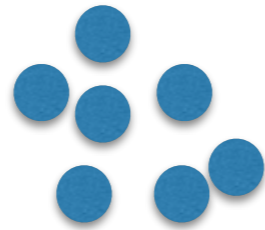
CLUSTERING

- Grouping sets of data points s.t.
 - points in same group are similar
 - points in different groups are dissimilar
- A form of unsupervised classification where there are no predefined labels

CLUSTERING

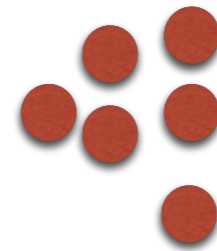
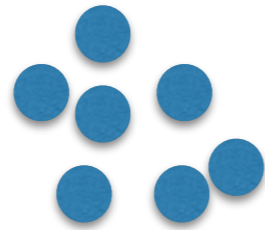
- Partition data into K disjoint groups
- Compression or Quantization
 - Compress n points into K representatives/groups
- Visualization or Understanding
 - Taxonomy: Animals Vs plants Vs Microbes, Science Vs Math Vs Social Sciences
 - Segmentation: different types of customers, students etc. Find natural groupings in data
- What this does not include: items belonging to more than one type

EXAMPLES



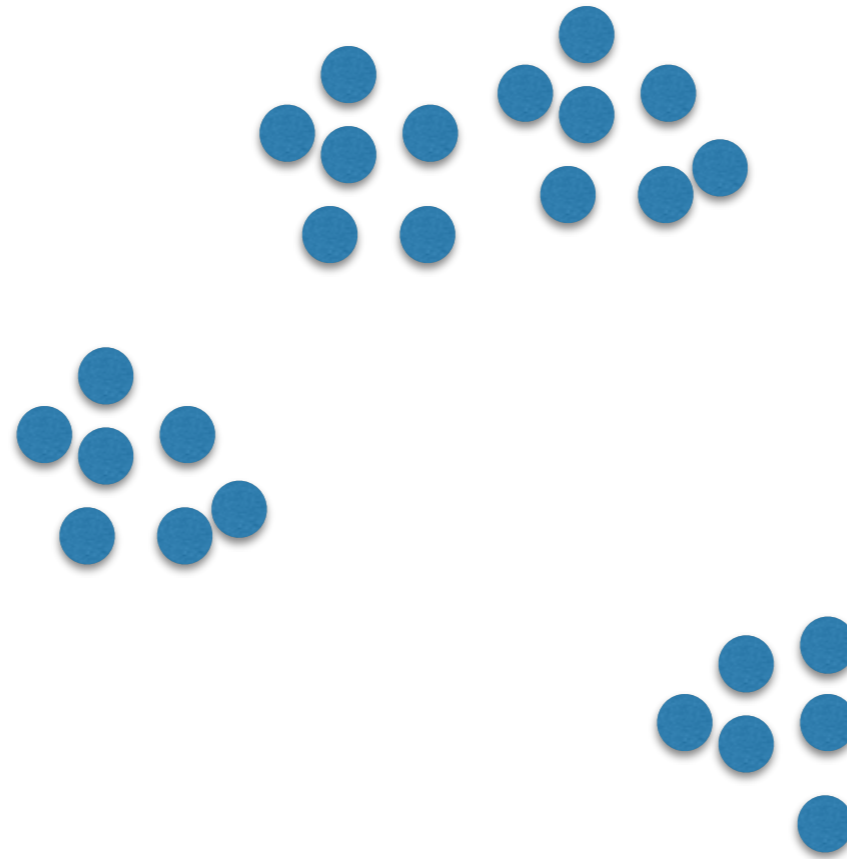
What are the clusters?

EXAMPLES



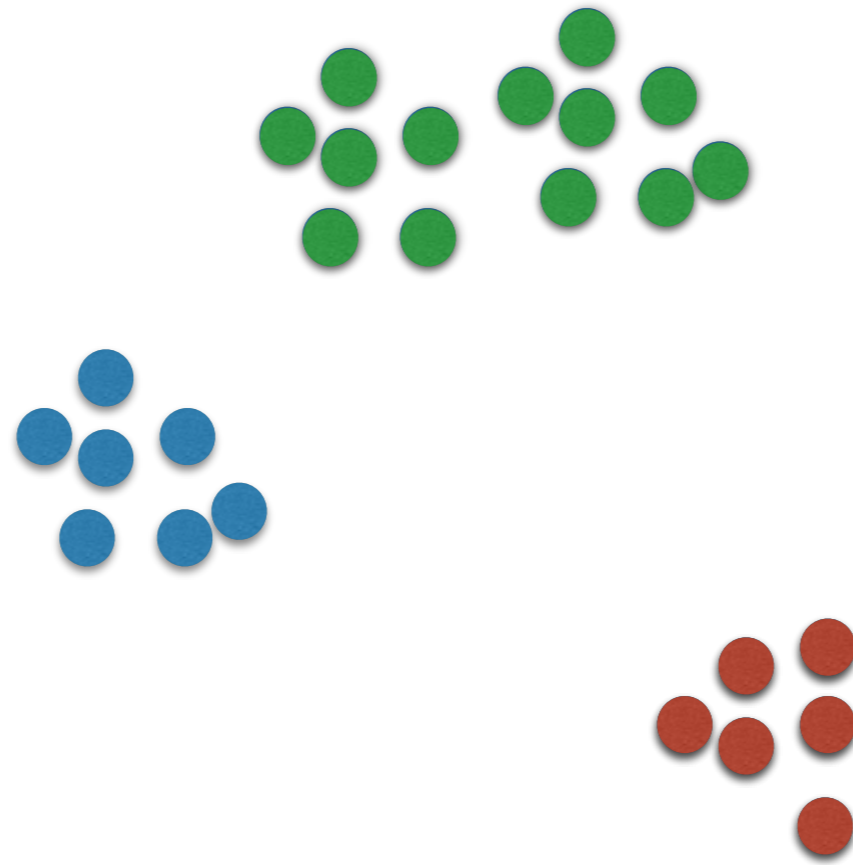
What are the clusters?

EXAMPLES



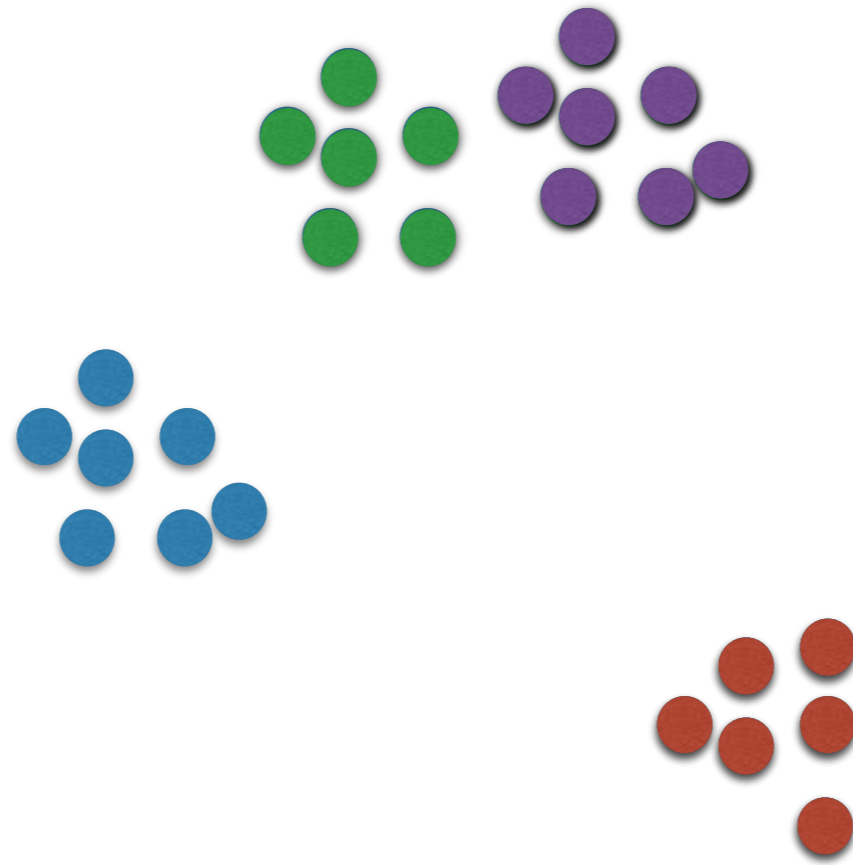
What are the clusters?

EXAMPLES



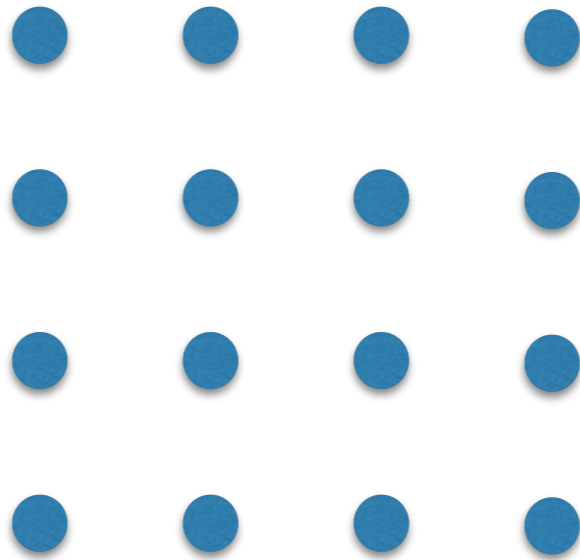
What are the clusters?

EXAMPLES



What are the clusters?

EXAMPLES



What are the clusters?

SOME NOTATIONS

- K -ary clustering is a partition of $\mathbf{x}_1, \dots, \mathbf{x}_n$ into K groups
- For now assume the magical K is given to use
- Clustering given by C_1, \dots, C_K , the partition of data points.
- Given a clustering, we shall use $c(\mathbf{x}_t)$ to denote the cluster identity of point \mathbf{x}_t according to the clustering.
- Let n_j denote $|C_j|$, clearly $\sum_{j=1}^K n_j = n$.

Can we formalize criterion/ objectives for clustering?

- . Assume points are represented as vectors
- . Use Euclidean distances for now
- . Similar points in same cluster
- . Points across clusters are dissimilar

CLUSTERING CRITERION

- 1 Minimize within-cluster scatter

$$M_1 = \sum_{j=1}^K \sum_{\mathbf{x}_s, \mathbf{x}_t \in C_j} \|\mathbf{x}_s - \mathbf{x}_t\|_2^2$$

- 2 Maximize between-cluster scatter

$$M_2 = \sum_{\mathbf{x}_s, \mathbf{x}_t: c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \|\mathbf{x}_s - \mathbf{x}_t\|_2^2$$

- 3 Minimize weighted within-cluster variance: $\mathbf{r}_j = \frac{1}{n_j} \sum_{\mathbf{x} \in C_j} \mathbf{x}$

$$M_3 = \sum_{j=1}^K n_j \sum_{\mathbf{x}_t \in C_j} \|\mathbf{x}_t - \mathbf{r}_j\|_2^2$$

- 4 Maximize smallest between-cluster distance

$$M_4 = \min_{\mathbf{x}_s, \mathbf{x}_t: c(\mathbf{x}_s) \neq c(\mathbf{x}_t)} \|\mathbf{x}_s - \mathbf{x}_t\|_2^2$$

- 5 Minimize largest within-cluster distance

$$M_5 = \max_{j \in [K]} \max_{\mathbf{x}_s, \mathbf{x}_t \in C_j} \|\mathbf{x}_s - \mathbf{x}_t\|_2^2$$

CLUSTERING CRITERION

6 Minimize within-cluster average scatter

$$M_6 = \sum_{j=1}^K \frac{1}{n_j} \sum_{\mathbf{x}_s, \mathbf{x}_t \in C_j} \|\mathbf{x}_s - \mathbf{x}_t\|_2^2$$

7 Minimize within-cluster variance: $\mathbf{r}_j = \frac{1}{n_j} \sum_{\mathbf{x} \in C_j} \mathbf{x}$

$$M_7 = \sum_{j=1}^K \sum_{\mathbf{x}_t \in C_j} \|\mathbf{x}_t - \mathbf{r}_j\|_2^2$$

How different are these
various criterion?