Machine Learning: Unsupervised Learning

Konstantin Tretyakov
http://kt.era.ee

IFI Summer School 2014

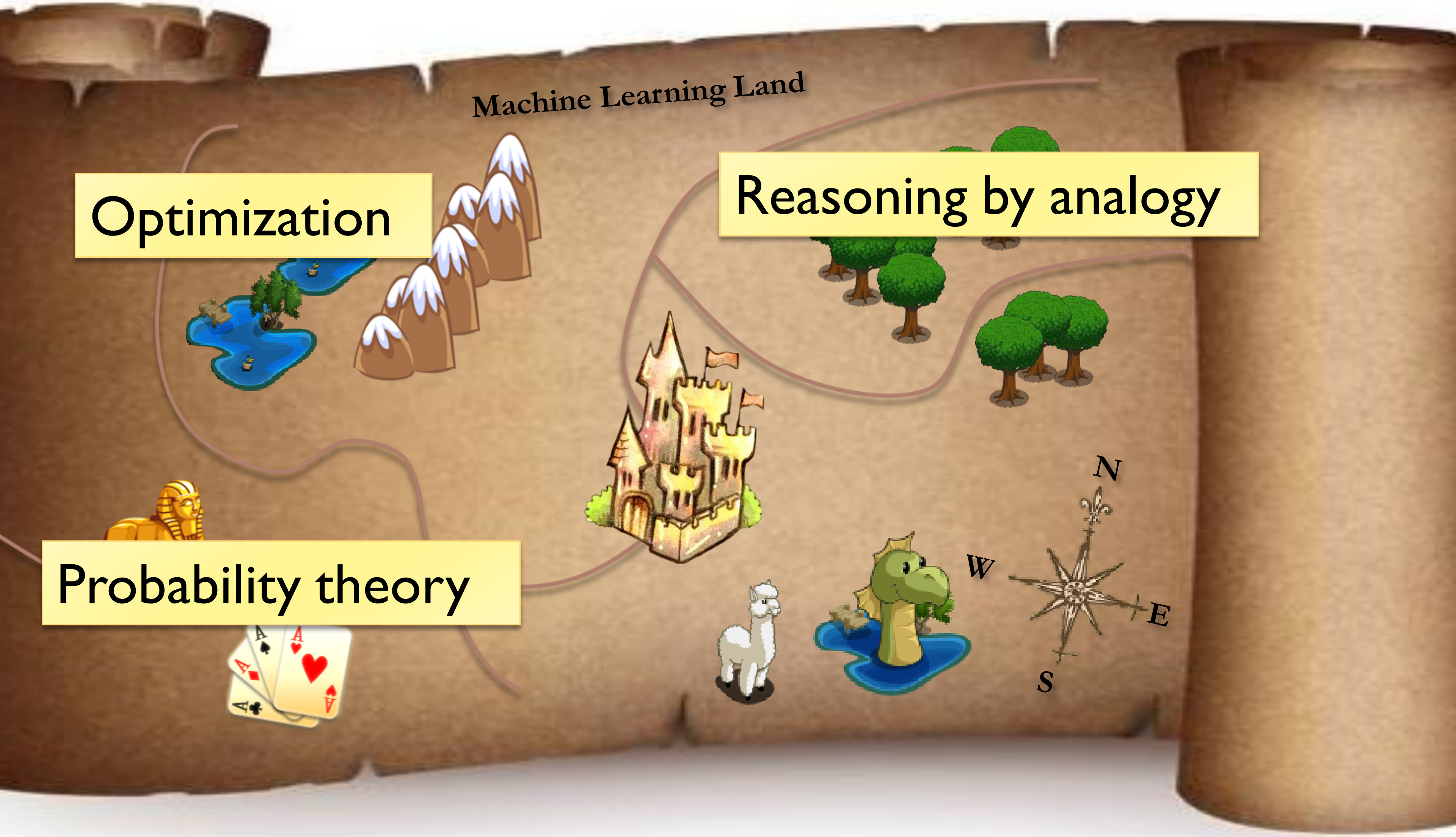
STACC Software Technology and Applications Competence Center

BIIT

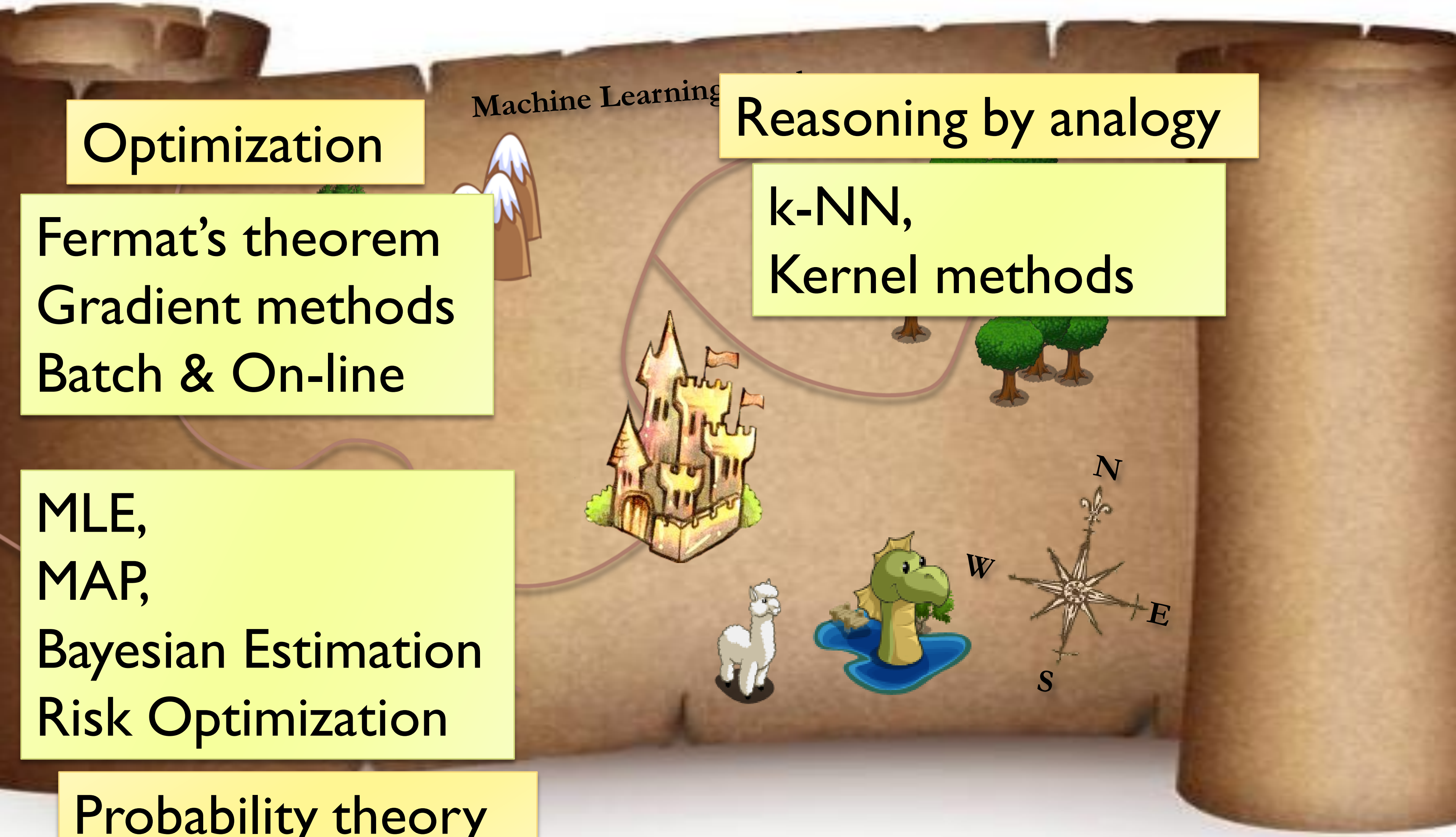
TARTU ÜLIKOOL · UNIVERSITAS TARTUENSIS · 1632

So far…

So far…


Machine Learning Land
Optimization
Reasoning by analogy
Probability theory
N
W
E
S

So far…



Optimization

- Fermat's theorem
- Gradient methods
- Batch & On-line

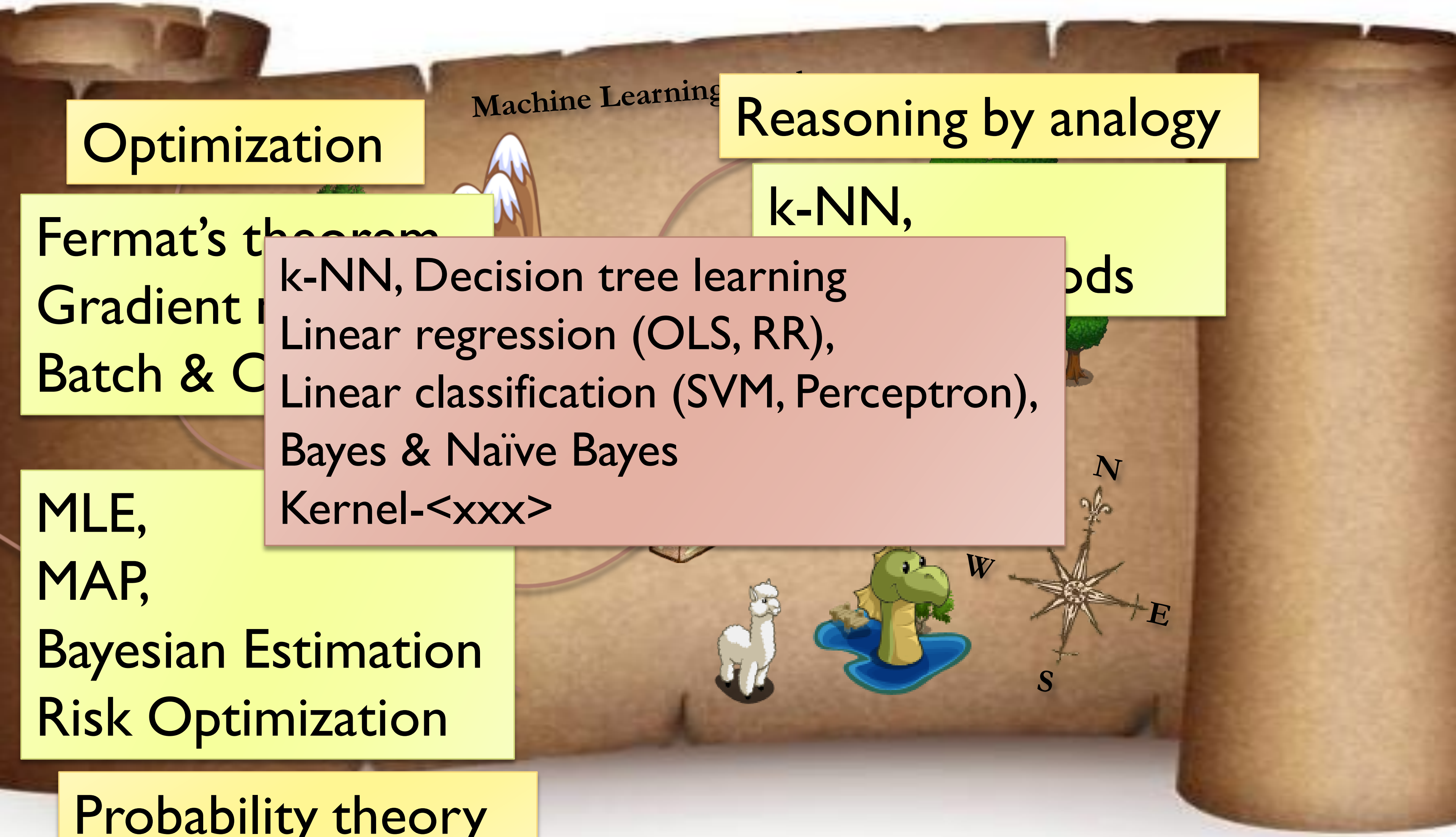
- MLE,
- MAP,
- Bayesian Estimation
- Risk Optimization

Probability theory

Reasoning by analogy

- k-NN,
- Kernel methods

So far…



Optimization

Fermat's th
Gradient
Batch & O

MLE,
MAP,
Bayesian Estimation
Risk Optimization

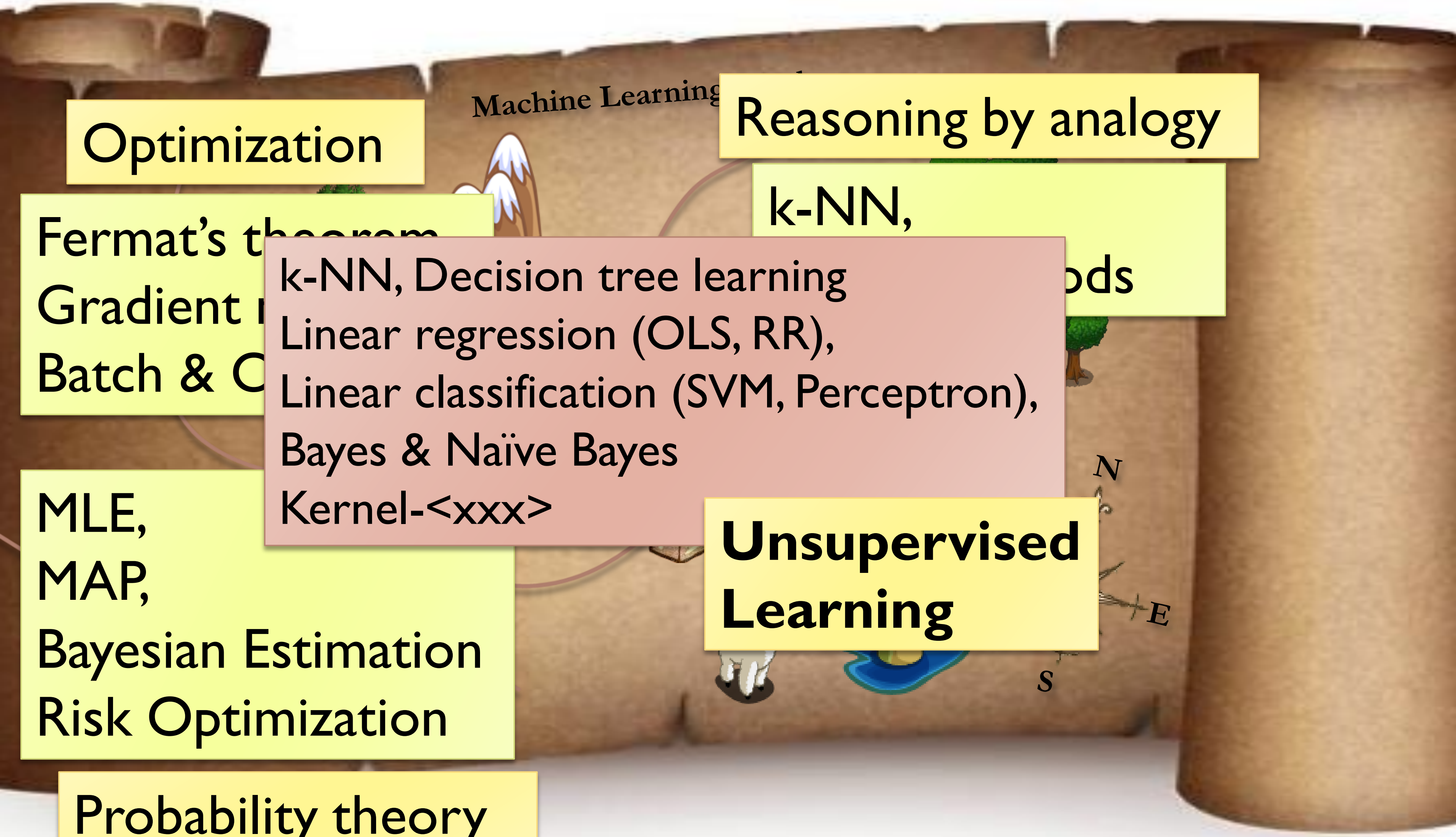
Probability theory

Reasoning by analogy

k-NN,
ods

k-NN, Decision tree learning
Linear regression (OLS, RR),
Linear classification (SVM, Perceptron),
Bayes & Naïve Bayes
Kernel-<xxx>

Next



Optimization

Fermat's t
Gradient
Batch & C

MLE,
MAP,
Bayesian Estimation
Risk Optimization

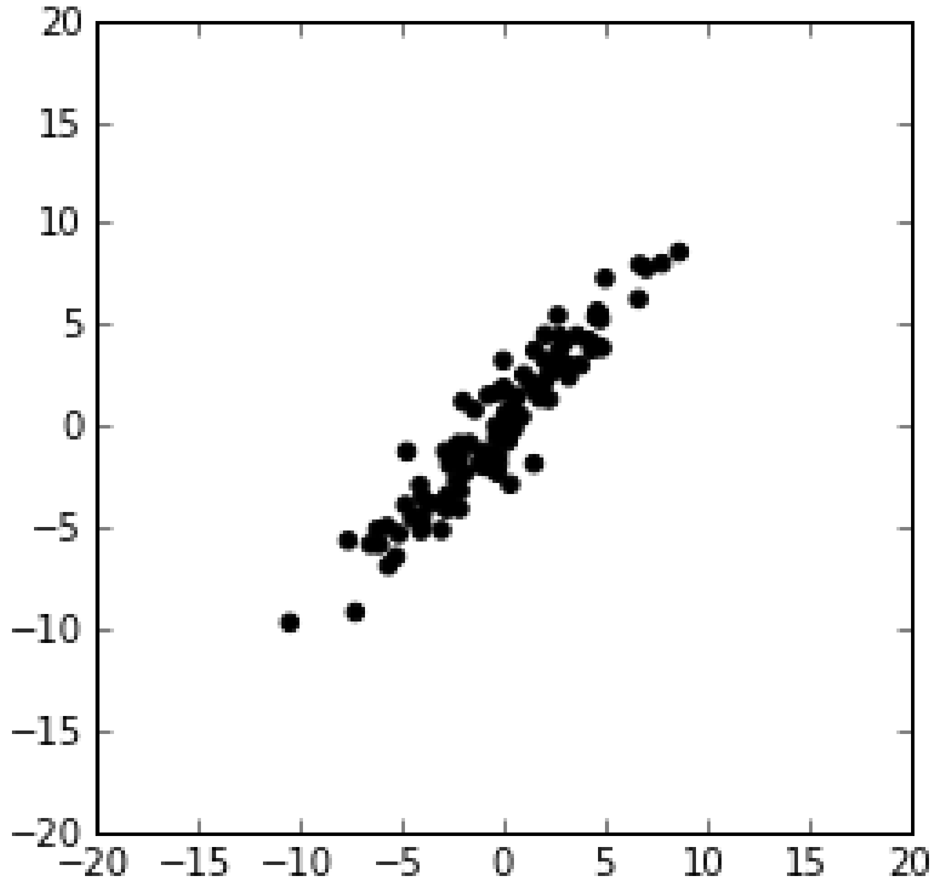
Probability theory

Reasoning by analogy

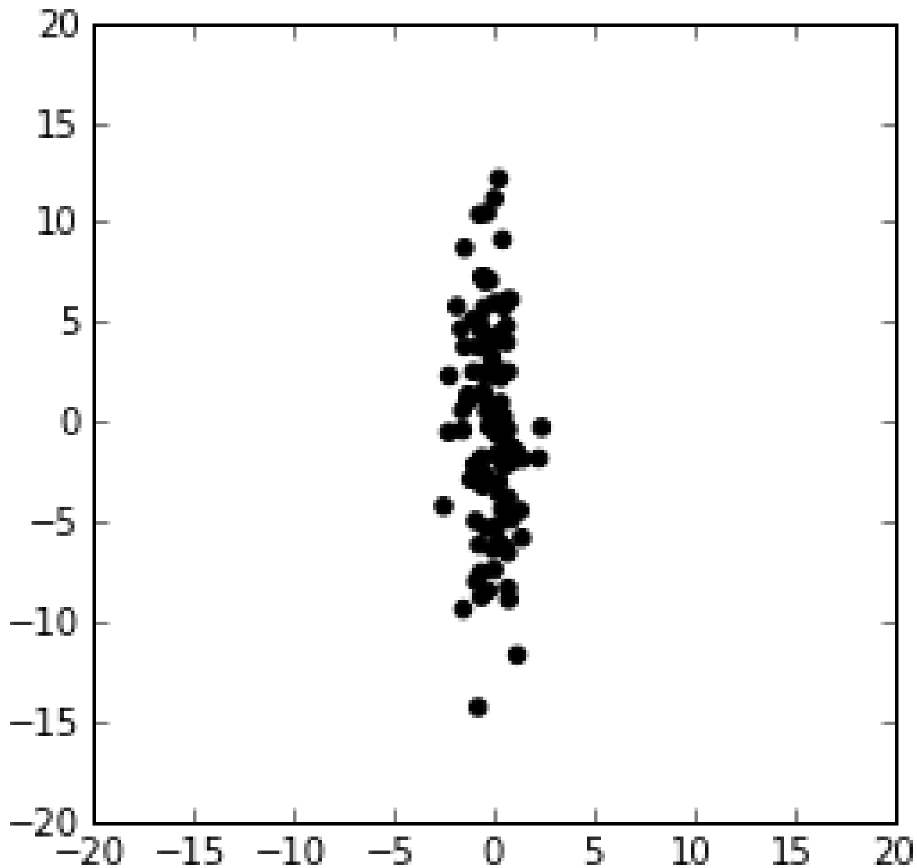
k-NN,
ods

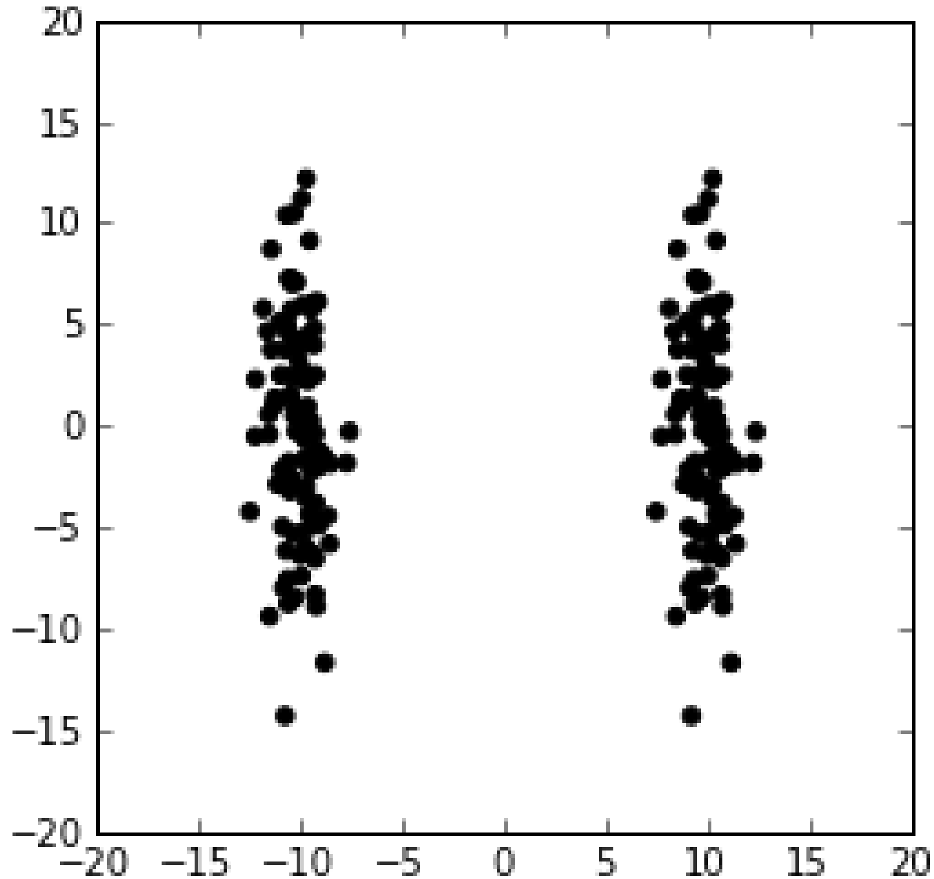
k-NN, Decision tree learning
Linear regression (OLS, RR),
Linear classification (SVM, Perceptron),
Bayes & Naïve Bayes
Kernel-<xxx>

Unsupervised Learning



20
15
10
5
0
−5
−10
−15
−20
−20
−15
−10
−5
0
5
10
15
20


20
15
10
5
0
−5
−10
−15
−20
−20
−15
−10
−5
0
5
10
15
20


20
15
10
5
0
−5
−10
−15
−20
−20 −15 −10 −5 0 5 10 15 20

AATAACGGCCCGATGAGGAAACGAACGGTCGCACT

AAGATGAGACATGTCCCGAAAGGTGCATAAGTTAT

GGACGAAAAACTTTCTTCGCCCTTTGATGTGCCCC

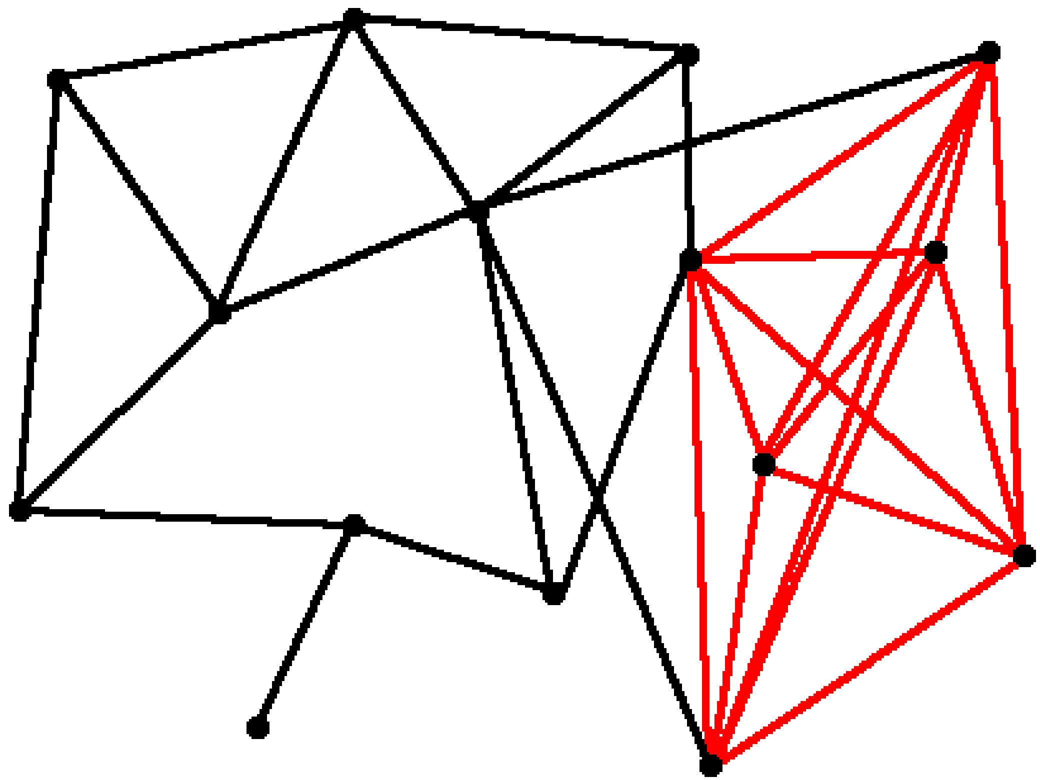
AGCGCGGGATGAGGATCAGCCCCCGCATTAGTTCA

ATATGCGAGCTTTCGCGCTCGGAAAGGGCAATAAA

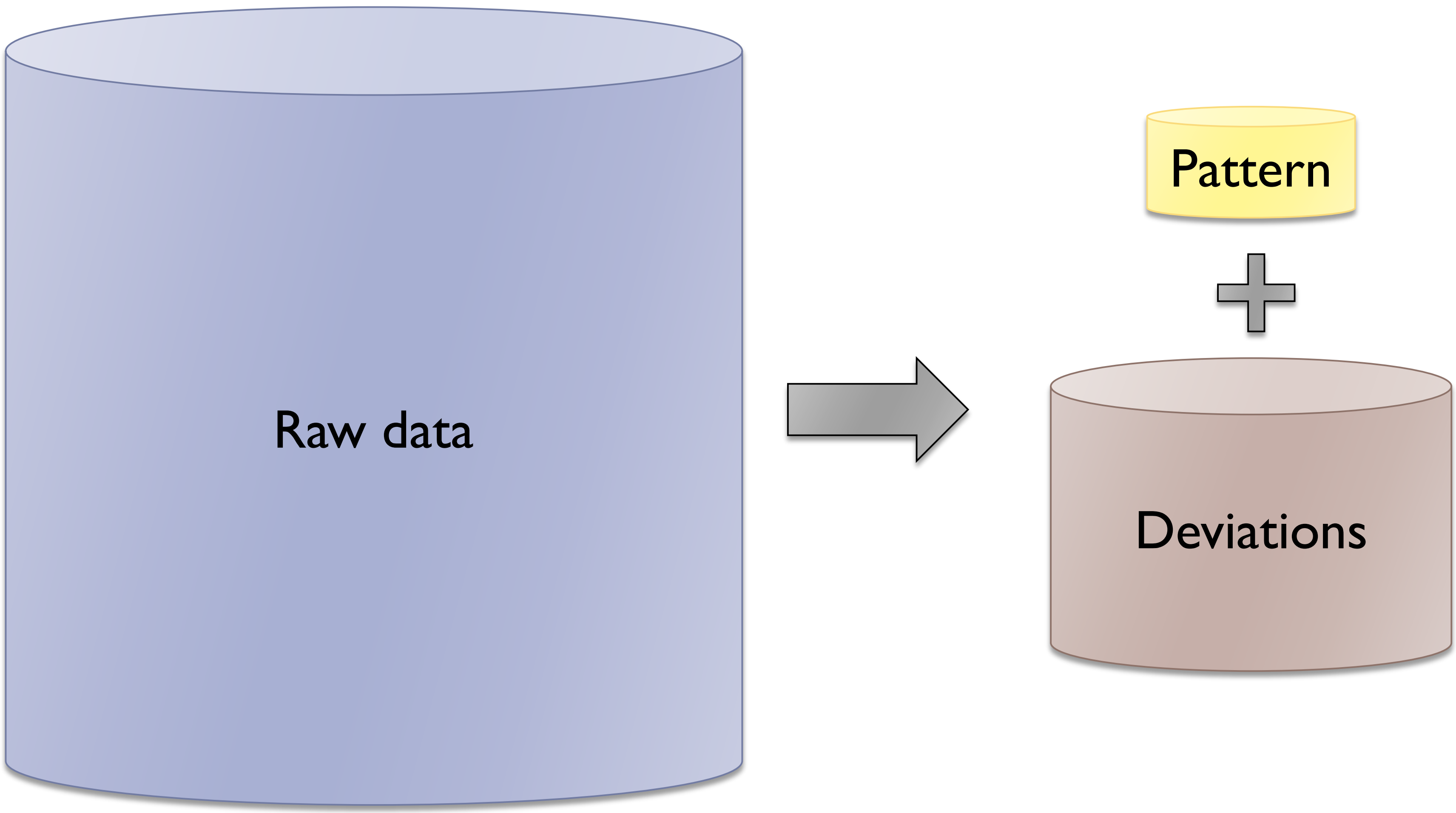
GCGACGGCCCCGATGAGGGGTGTTACTAGATTGGA

TGGGTGGTTCAGATCTCGGCTTACCCCCTTTATCA

ACCCTGCTACAGACTCGTTGAGAATGCTACGGATC

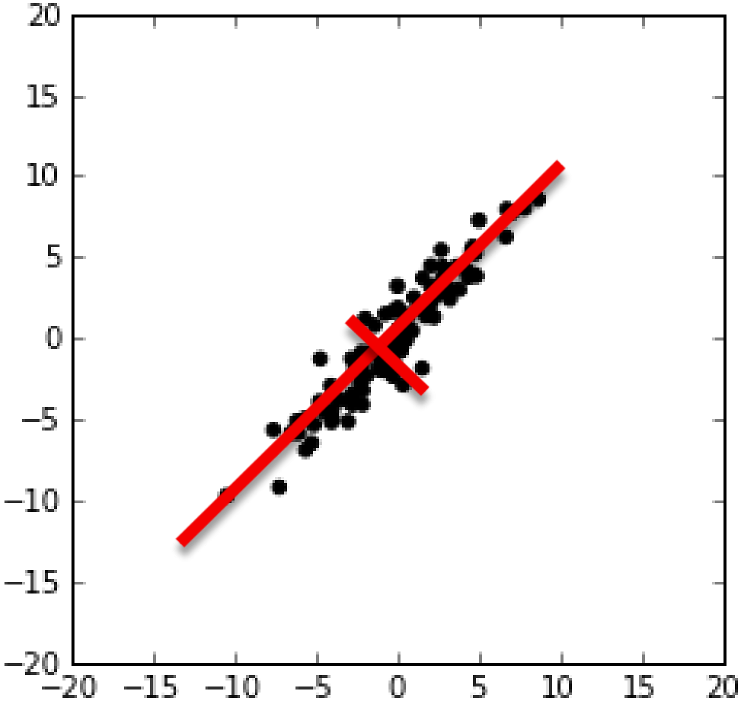


Data Mining

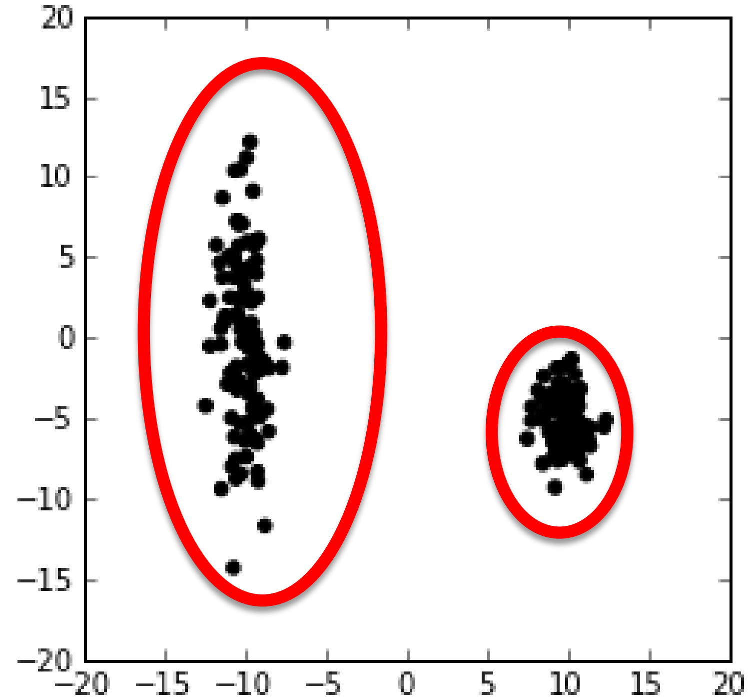


Unsupervised learning patterns

Decomposition



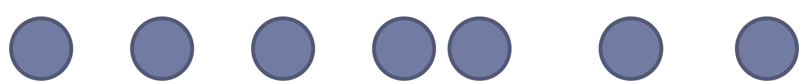
Clustering



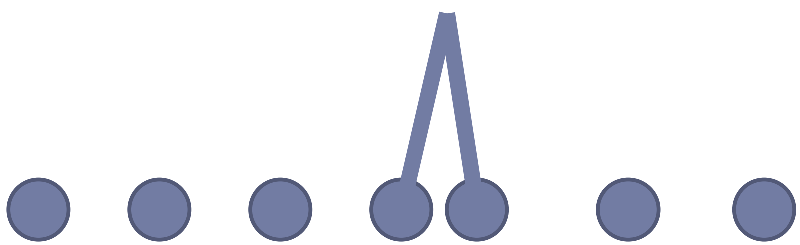
Quiz

- Why would one need clustering?

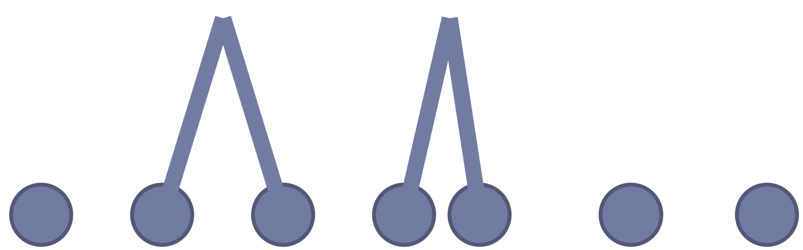
Hierarchical clustering



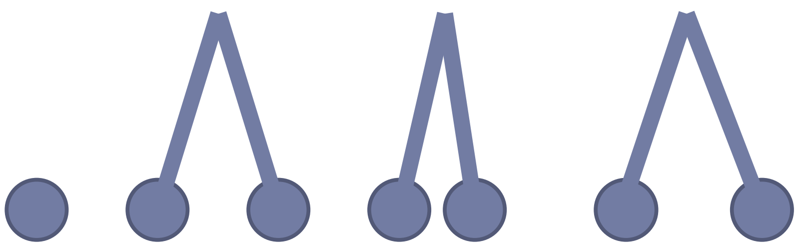
Hierarchical clustering



Hierarchical clustering

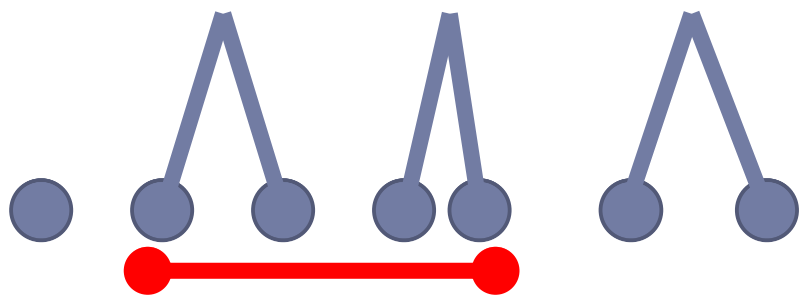


Hierarchical clustering



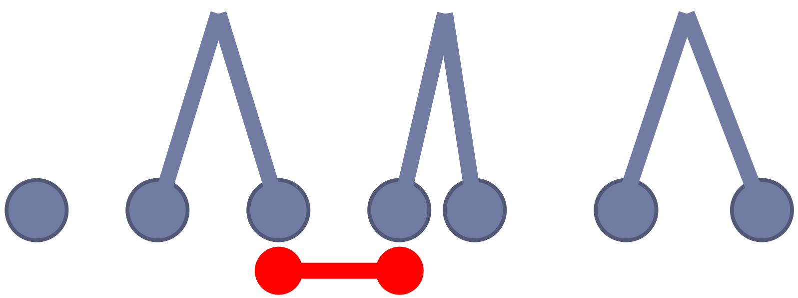
Hierarchical clustering

Complete linkage



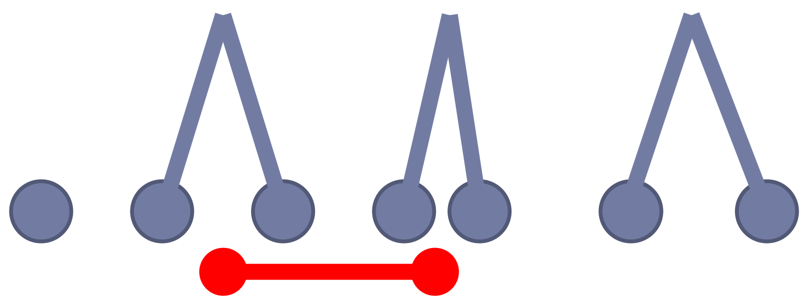
Hierarchical clustering

Single linkage



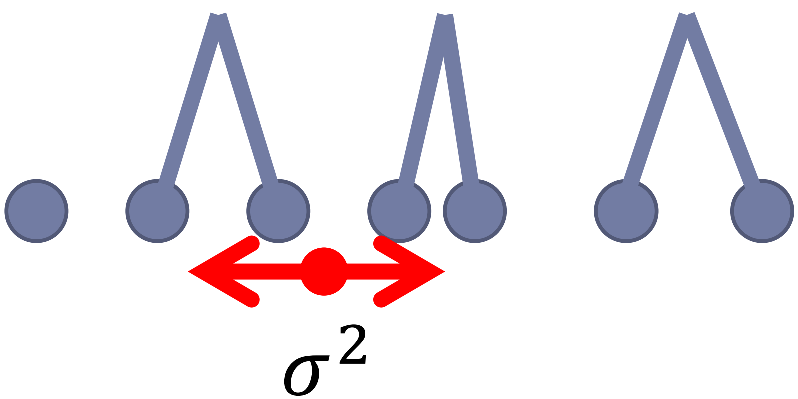
Hierarchical clustering

Average linkage



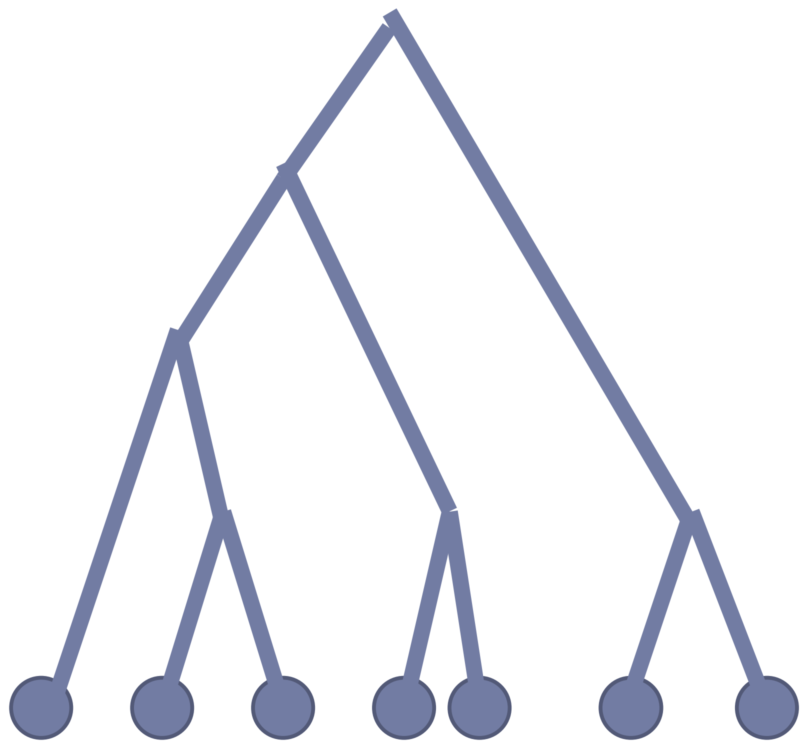
Hierarchical clustering

Ward linkage

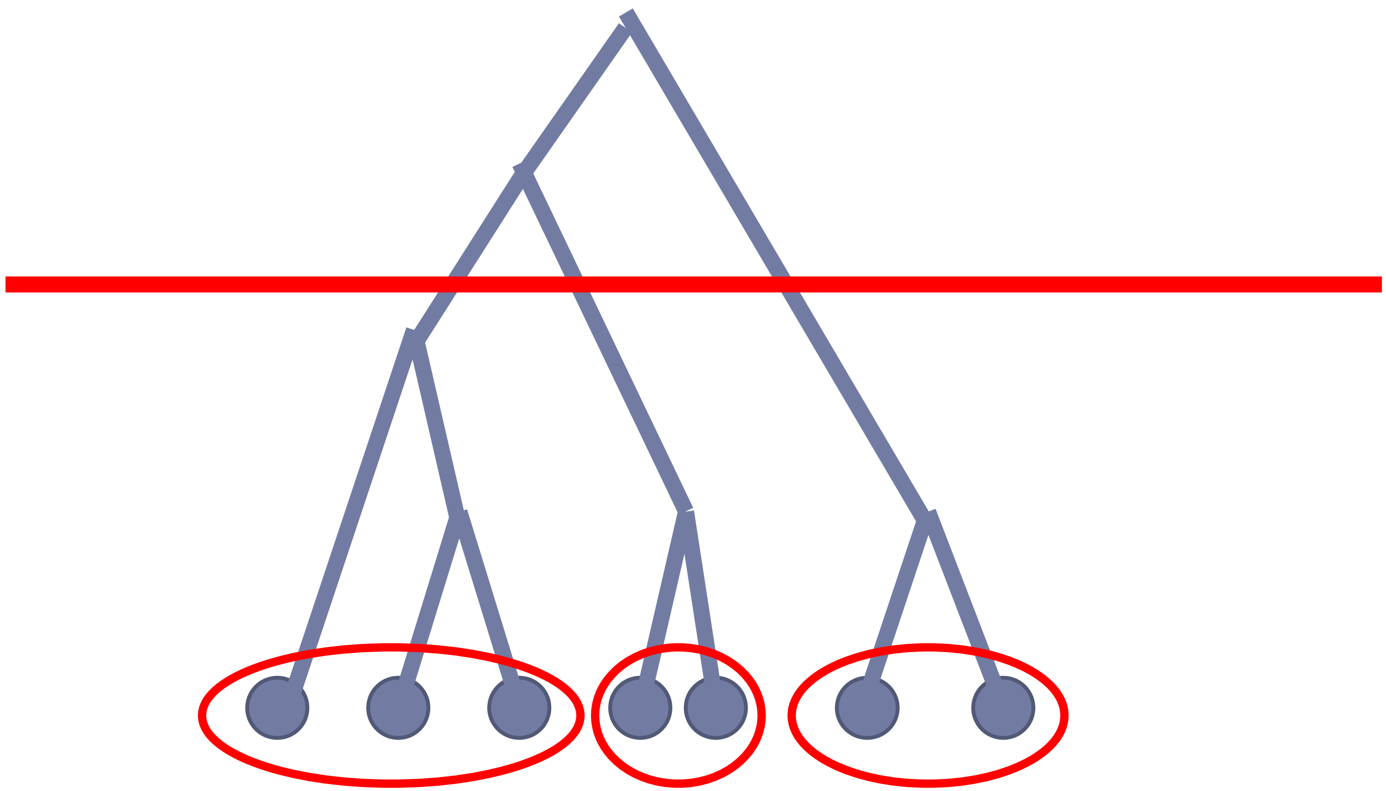


Hierarchical clustering

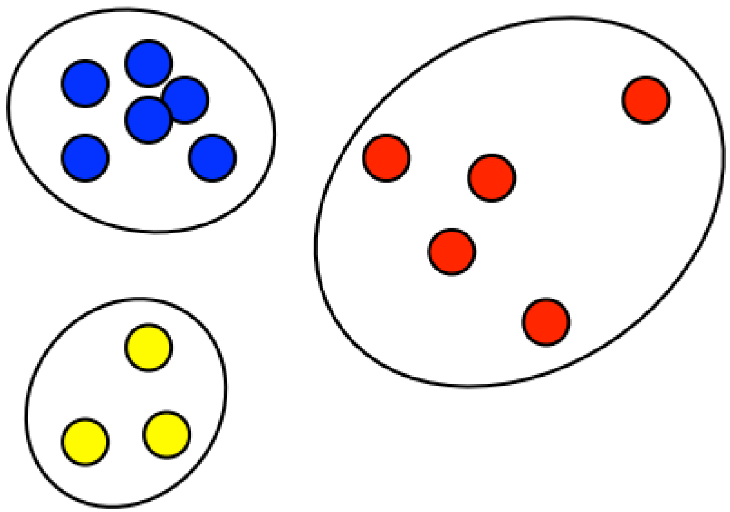
Hierarchical clustering



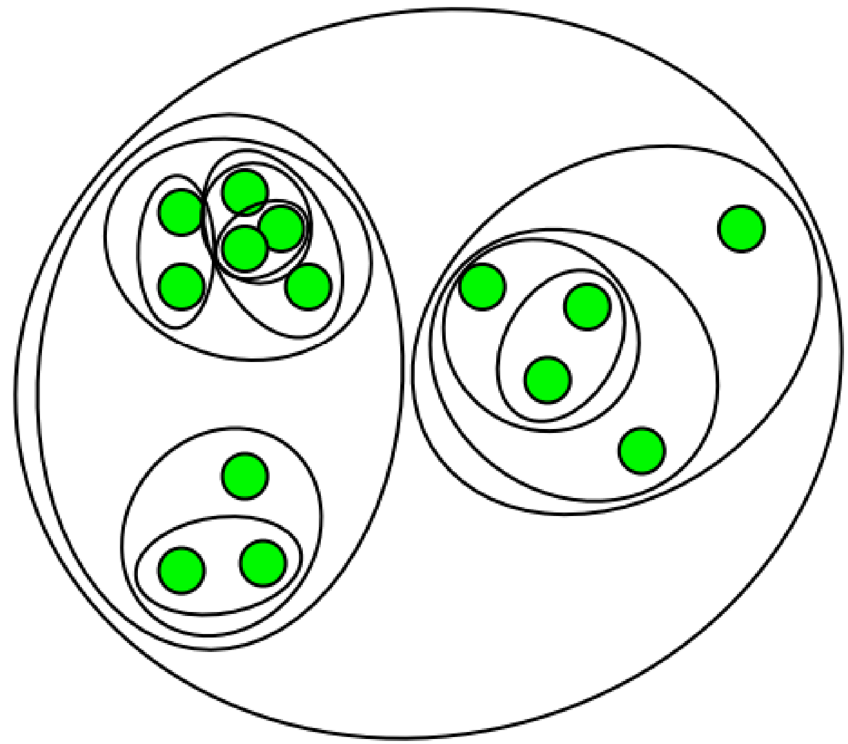
Hierarchical clustering



Partitional vs Hierarchical

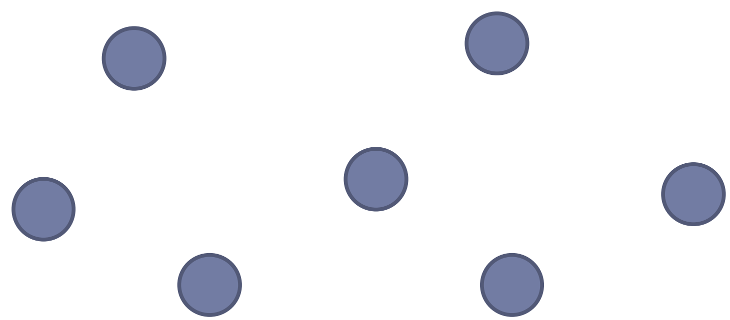


Partitional clustering finds a fixed number of clusters

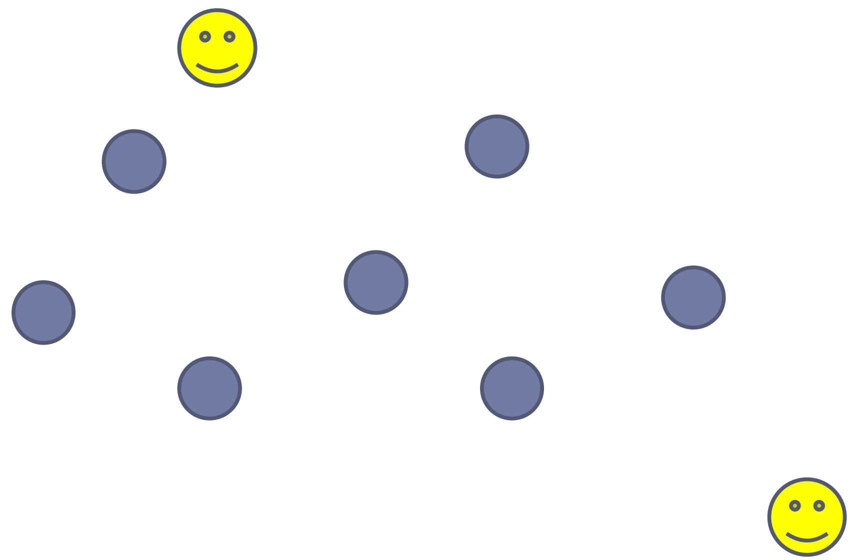


Hierarchical clustering creates a series of clusterings contained in each other

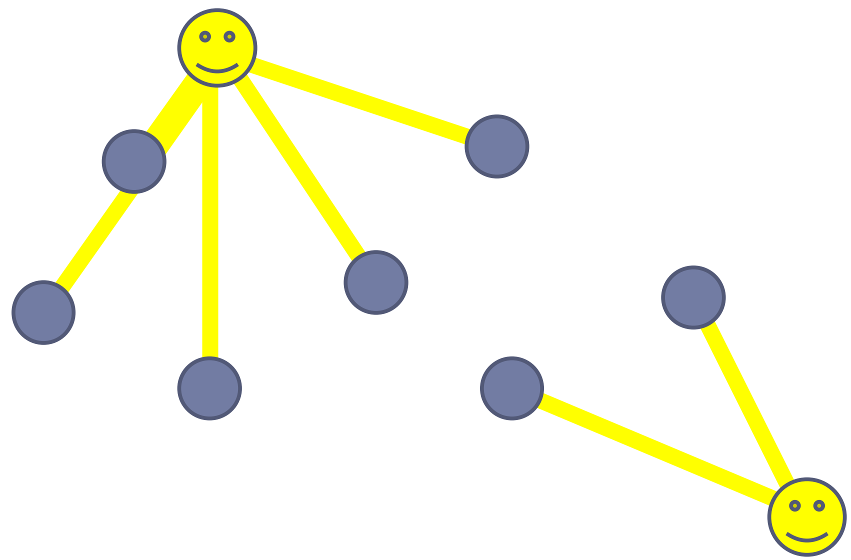
K-means



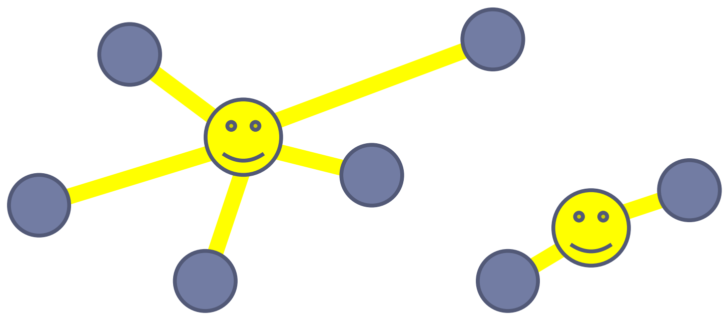
K-means



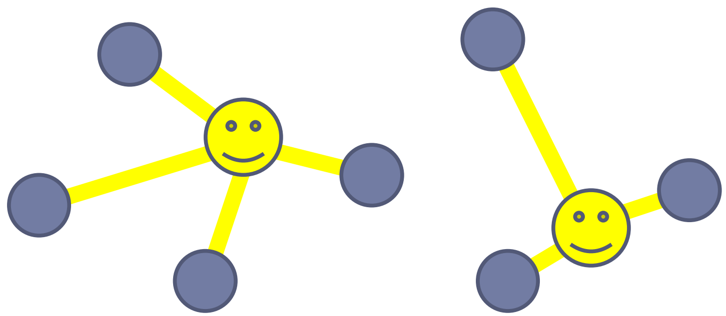
K-means



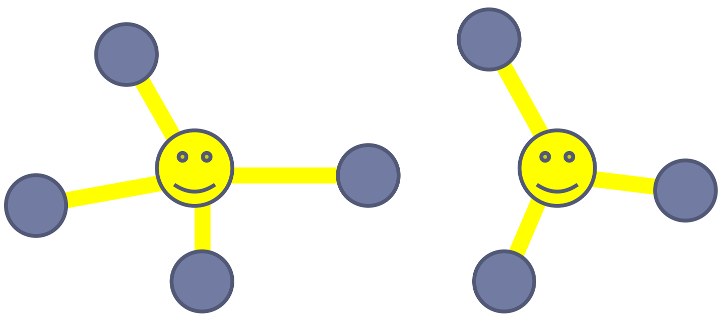
K-means



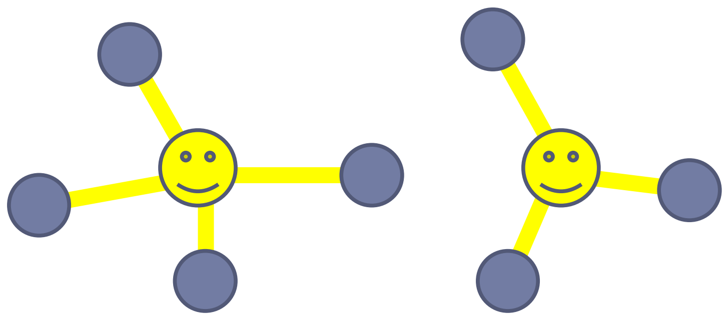
K-means



K-means



K-means



$$\mathrm{argmin}_{c_1,\ldots,c_K} \sum_i \left\| x_i - c_{\mathrm{closest\_to}\,(i)} \right\|^2$$

K-means

$$\text{argmin}_{c_1,\ldots,c_K} \sum_i \left\| x_i - c_{\text{closest\_to}(i)} \right\|^2$$

- **Need to find cluster centers c_k.**

$$c_1 = ?, c_2 = ?, \ldots, c_K = ?$$

K-means

$$\text{argmin}_{c_1,\dots,c_K} \sum_i \left\| x_i - c_{\text{closest\_to}(i)} \right\|^2$$

- **Need to find cluster centers c_k.**

$$c_1 = ?, c_2 = ?, \dots, c_K = ?$$

- **Introduce *latent variables* (one for each x_i)**

$$a_i = \text{closest\_cluster\_center}(i)$$

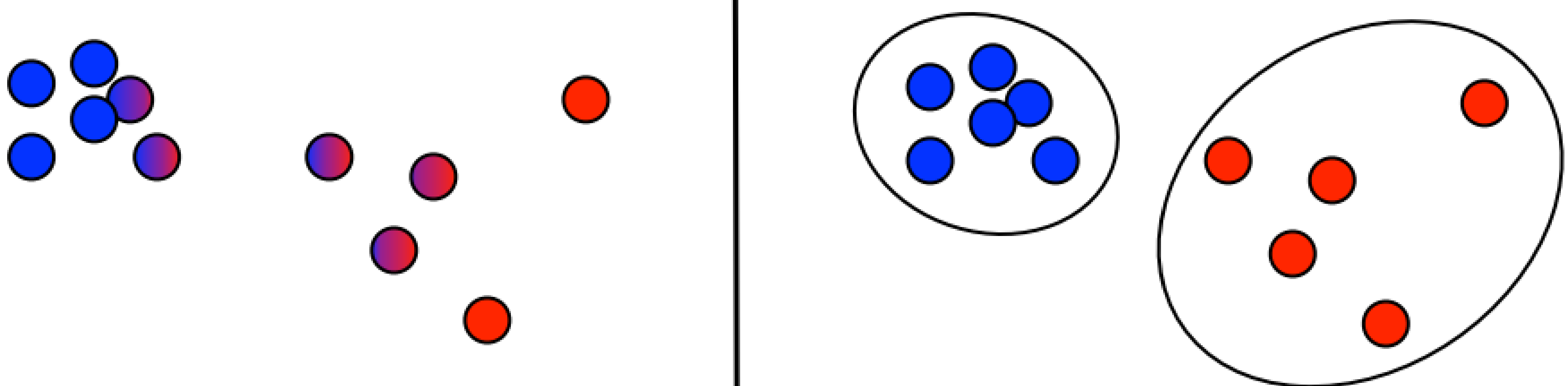
$$a_1 = ?, a_2 = ?, a_3 = ?, \dots, a_n = ?$$

K-means

$$\operatorname{argmin}_{c_1,\ldots,c_K} \sum_i \left\| x_i - c_{\mathrm{closest\_to}(i)} \right\|^2$$

- **For fixed c_k we can find optimal a_i**
- **For fixed a_i we can find optimal c_k.**
- **Iterate to convergence.**

Fuzzy vs Hard



Each object belongs to each cluster with some weight (the weight can be zero)

Each object belongs to exactly one cluster

- Slide © M.Kull, http://courses.cs.ut.ee/2012/ml/

Gaussian Mixture Modeling

$$\boldsymbol{X} \sim [N(\boldsymbol{\mu}_1, \sigma_1^2) \text{ or } N(\boldsymbol{\mu}_2, \sigma_2^2)]$$

Given X, estimate $\boldsymbol{\mu_i}, \boldsymbol{\sigma_i^2}$

Gaussian Mixture Modeling

$$\boldsymbol{X} \sim [N(\boldsymbol{\mu}_1, \sigma_1^2) \quad \text{or} \quad N(\boldsymbol{\mu}_2, \sigma_2^2)]$$

Given $\boldsymbol{X}$, estimate $\boldsymbol{\mu_i}, \boldsymbol{\sigma_i^2}$

➡ **MLE**

Gaussian Mixture Modeling

$$\boldsymbol{X} \sim [N(\boldsymbol{\mu}_1, \sigma_1^2) \;\text{ or }\; N(\boldsymbol{\mu}_2, \sigma_2^2)]$$

Given X, estimate $\boldsymbol{\mu_i}, \boldsymbol{\sigma_i^2}$

→ **MLE**

→ **Expectation-Maximization (EM)**

SKLearn's Clustering

```
from sklearn.cluster
import
    Ward,
    KMeans,
    DBScan,
    MeanShift,
    SpectralClustering,
    AffinityPropagation
```

SKLearn's Clustering

```
from sklearn.cluster
import
    Ward,
    KMeans,
    DBScan,
    MeanShift,

    SpectralClustering,
    AffinityPropagation
```

Use feature vectors

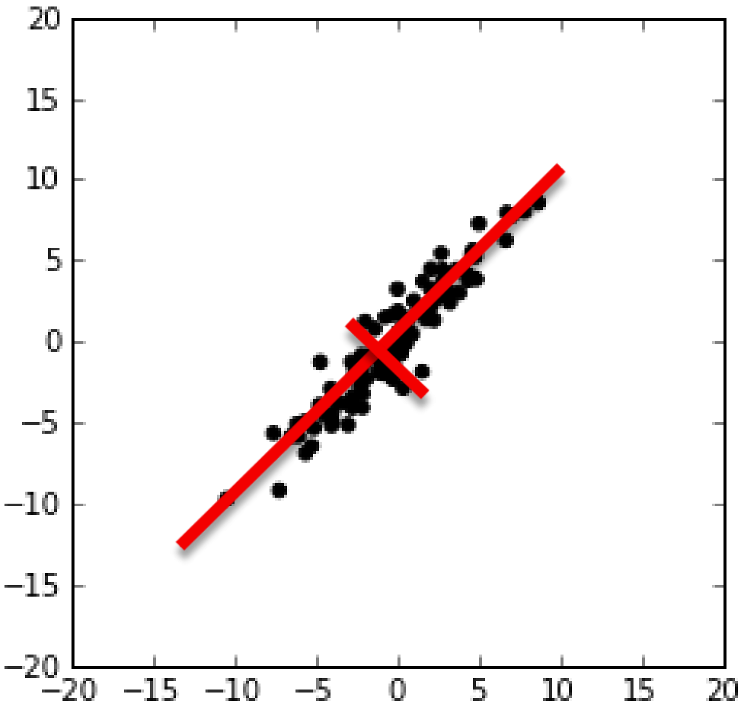
Use distance matrix

Quiz

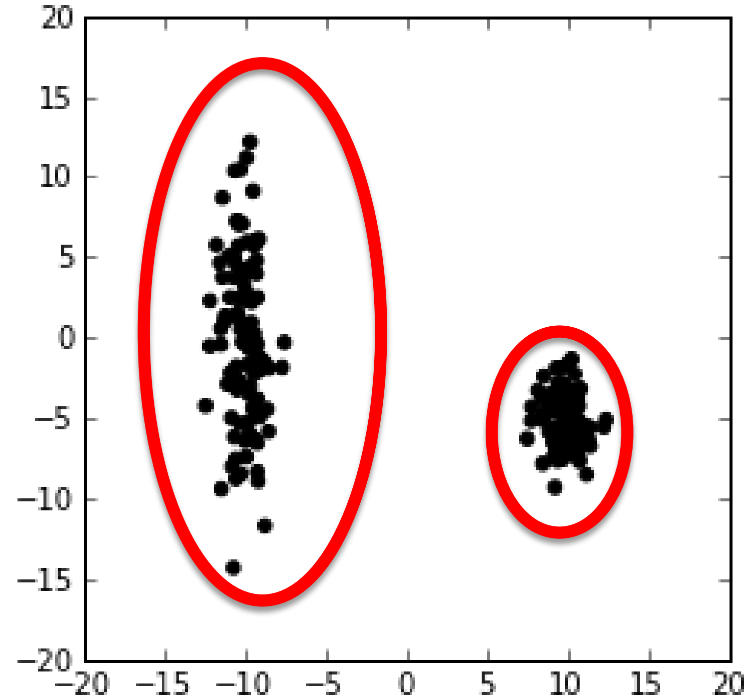
- Fuzzy clustering means that ________
- K-means finds a set of cluster centers, which have the smallest ________________
- K-means can get stuck in a local minimum (Y/N)?

Unsupervised learning patterns

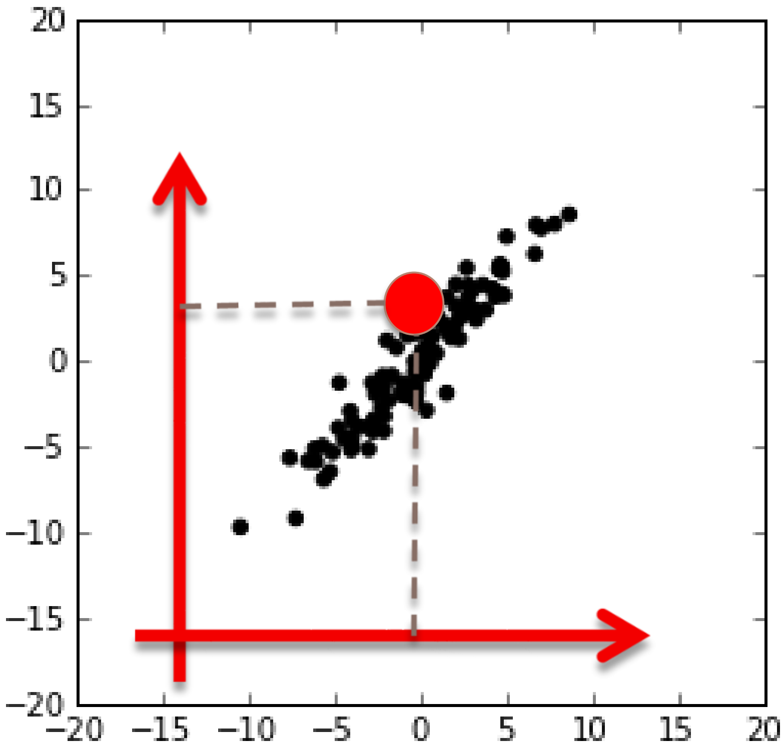
Decomposition



Clustering

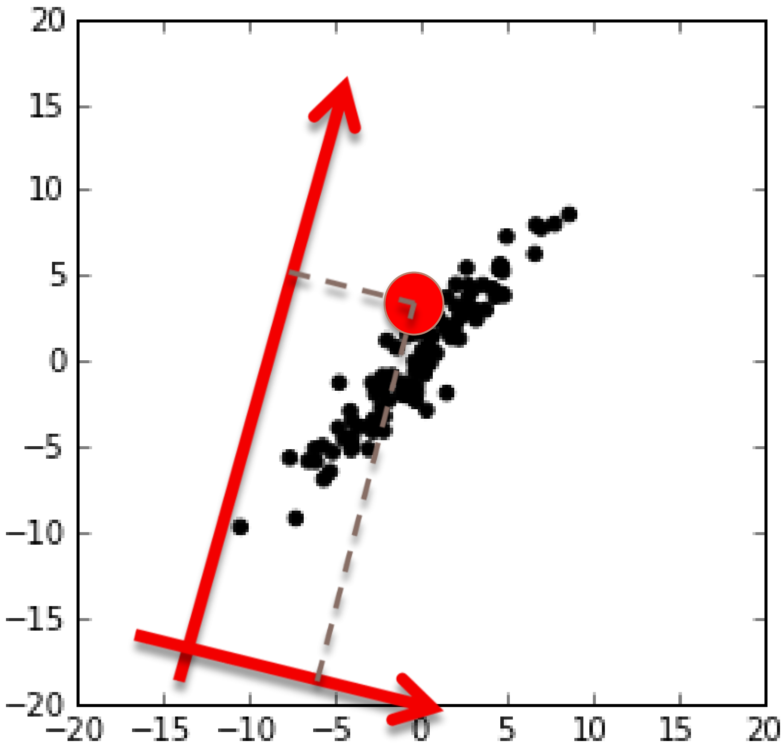


Canonical basis



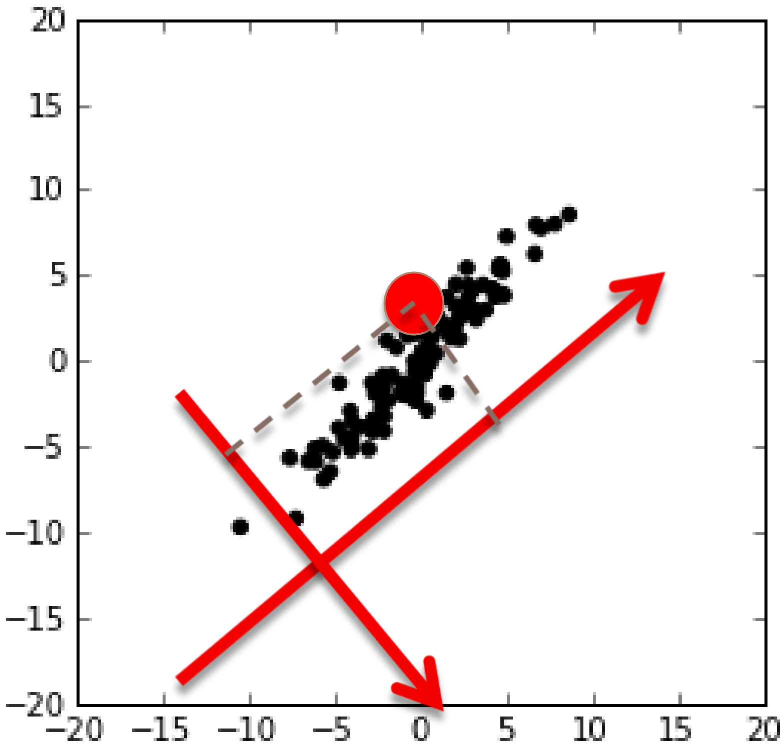
$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Alternative basis



$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \alpha \begin{pmatrix} 0.9 \\ -0.1 \end{pmatrix} + \beta \begin{pmatrix} 0.1 \\ 0.9 \end{pmatrix}$$

Alternative basis



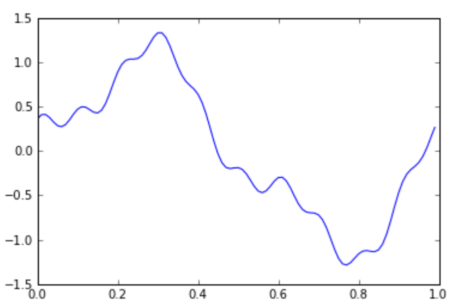
$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \alpha \begin{pmatrix} 0.4 \\ -0.6 \end{pmatrix} + \beta \begin{pmatrix} 0.6 \\ 0.4 \end{pmatrix}$$

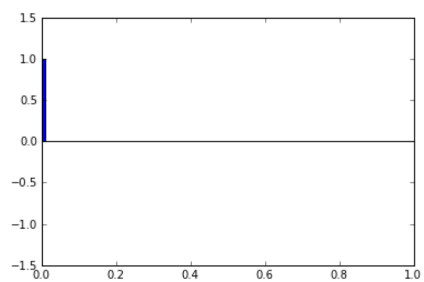
Linear Decomposition

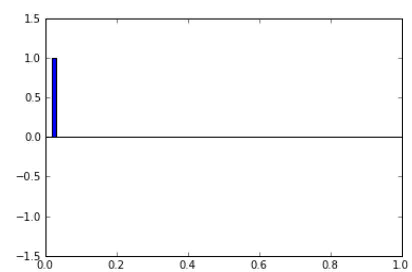
$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \cdots \\ x_{100000} \end{pmatrix} = \alpha_1 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \alpha_2 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \cdots + \alpha_m \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

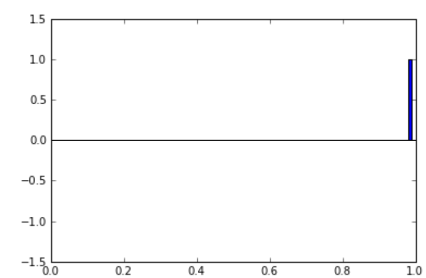
Linear Decomposition

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \cdots \\ x_{100000} \end{pmatrix} = \alpha_1 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \alpha_2 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \cdots + \alpha_m \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$



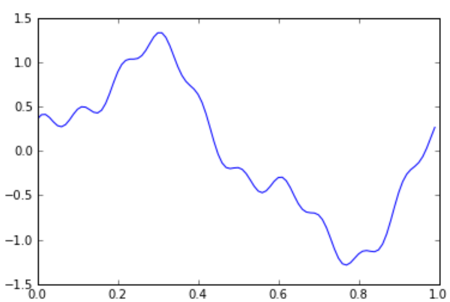


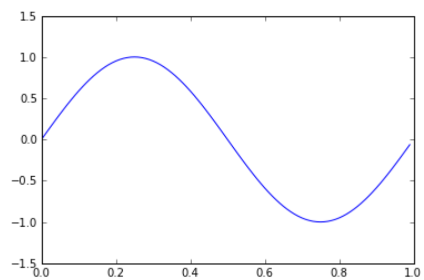


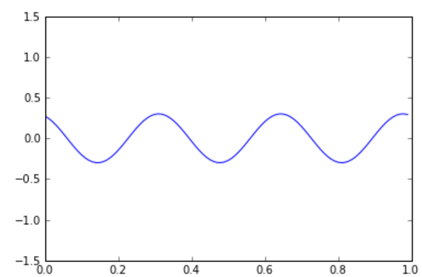


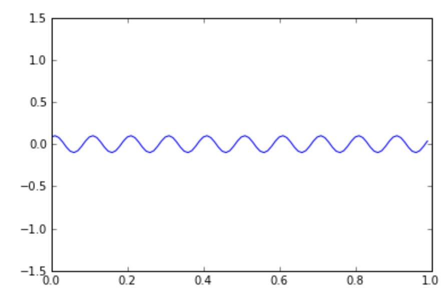
Linear Decomposition

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \dots \\ x_{100000} \end{pmatrix} = \alpha_1 \begin{pmatrix} 0.0 \\ 0.1 \\ 0.1 \\ 0.2 \\ \vdots \\ 0.0 \end{pmatrix} + \alpha_2 \begin{pmatrix} 0.3 \\ 0.2 \\ 0.2 \\ 0.1 \\ \vdots \\ 0.3 \end{pmatrix} + \alpha_m \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.0 \\ \vdots \\ 0.0 \end{pmatrix}$$



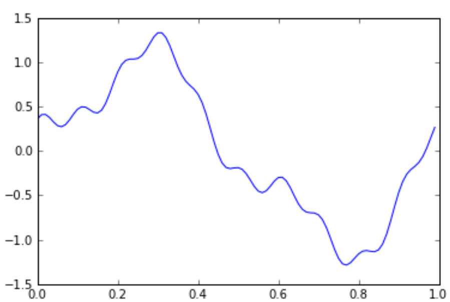






Linear Decomposition

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \dots \\ x_{100000} \end{pmatrix} = \alpha_1 \begin{pmatrix} 0.0 \\ 0.1 \\ 0.1 \\ 0.2 \\ \vdots \\ 0.0 \end{pmatrix} + \alpha_2 \begin{pmatrix} 0.3 \\ 0.2 \\ 0.2 \\ 0.1 \\ \vdots \\ 0.3 \end{pmatrix} + \alpha_m \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.0 \\ \vdots \\ 0.0 \end{pmatrix}$$



Linear Decomposition

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \cdots \\ x_{100000} \end{pmatrix} = \alpha_1 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \alpha_2 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \cdots + \alpha_m \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$



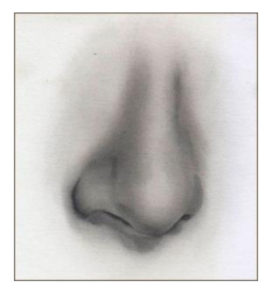

Linear Decomposition

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \dots \\ x_{100000} \end{pmatrix} = \alpha_1 \begin{pmatrix} 0.0 \\ 0.1 \\ 0.1 \\ 0.2 \\ \vdots \\ 0.0 \end{pmatrix} + \alpha_2 \begin{pmatrix} 0.3 \\ 0.2 \\ 0.2 \\ 0.1 \\ \vdots \\ 0.3 \end{pmatrix} + \alpha_m \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.0 \\ \vdots \\ 0.0 \end{pmatrix}$$




Linear Decomposition

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \dots \\ x_{100000} \end{pmatrix} = \alpha_1 \begin{pmatrix} 0.0 \\ 0.1 \\ 0.1 \\ 0.2 \\ \vdots \\ 0.0 \end{pmatrix} + \alpha_2 \begin{pmatrix} 0.3 \\ 0.2 \\ 0.2 \\ 0.1 \\ \vdots \\ 0.3 \end{pmatrix} + \alpha_m \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.0 \\ \vdots \\ 0.0 \end{pmatrix}$$



Linear Decomposition

$$\boldsymbol{x} = \alpha_1 \boldsymbol{v_1} + \alpha_2 \boldsymbol{v_2} + \cdots + \alpha_m \boldsymbol{v_m}$$

Linear Decomposition

$$\boldsymbol{x} = \alpha_1 \boldsymbol{v_1} + \alpha_2 \boldsymbol{v_2} + \cdots + \alpha_m \boldsymbol{v_m}$$

$$\boldsymbol{x} = \begin{pmatrix} \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & & \boldsymbol{v}_{\boldsymbol{m}} \\ \vdots & \vdots & \cdots & \vdots \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix}$$

Linear Decomposition

$$\boldsymbol{x} = \alpha_1 \boldsymbol{v_1} + \alpha_2 \boldsymbol{v_2} + \cdots + \alpha_m \boldsymbol{v_m}$$

$$\boldsymbol{x} = \boldsymbol{V\alpha}$$

Linear Decomposition

$$\boldsymbol{x} = \alpha_1 \boldsymbol{v_1} + \alpha_2 \boldsymbol{v_2} + \cdots + \alpha_m \boldsymbol{v_m}$$

$$\boldsymbol{x} = \boldsymbol{V\alpha}$$

$$\boldsymbol{\alpha} = ?$$

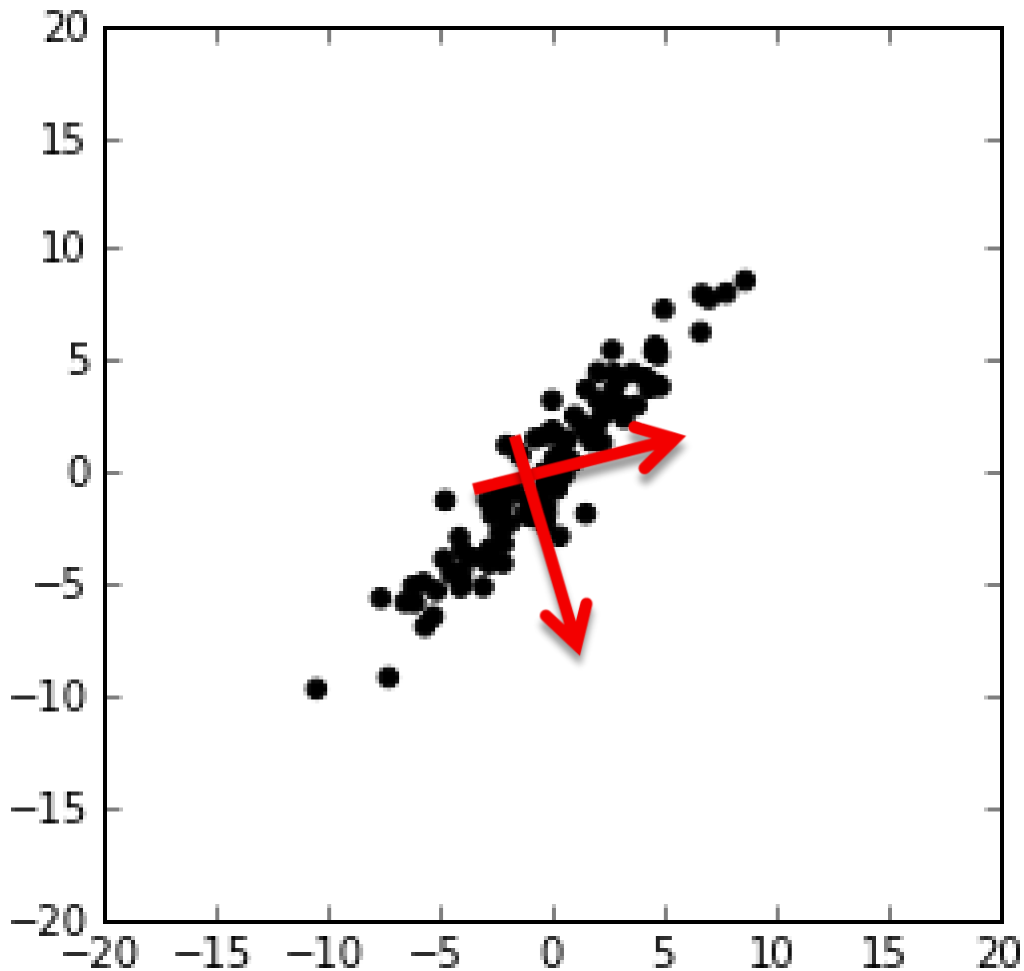
Linear Decomposition

$$\boldsymbol{x} = \alpha_1 \boldsymbol{v_1} + \alpha_2 \boldsymbol{v_2} + \cdots + \alpha_m \boldsymbol{v_m}$$

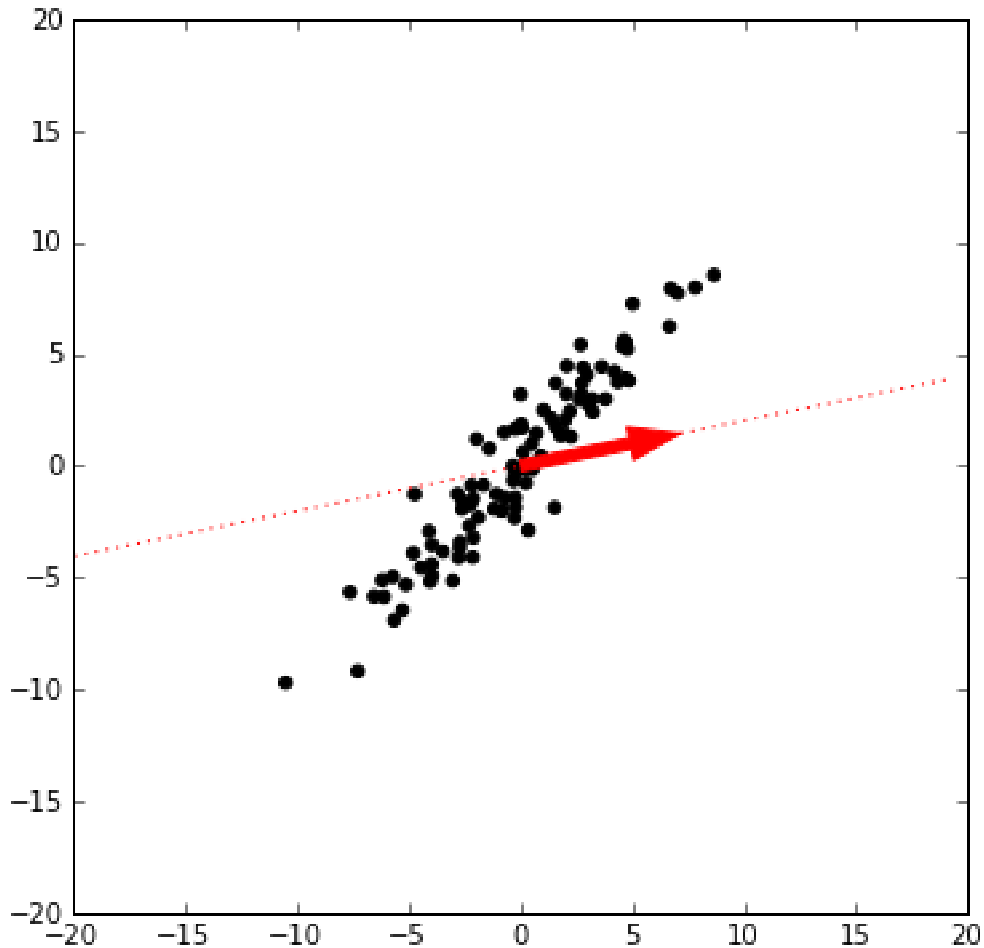
$$\boldsymbol{x} = \boldsymbol{V\alpha}$$

$$\boldsymbol{\alpha} = \boldsymbol{V}^+ \boldsymbol{x}$$

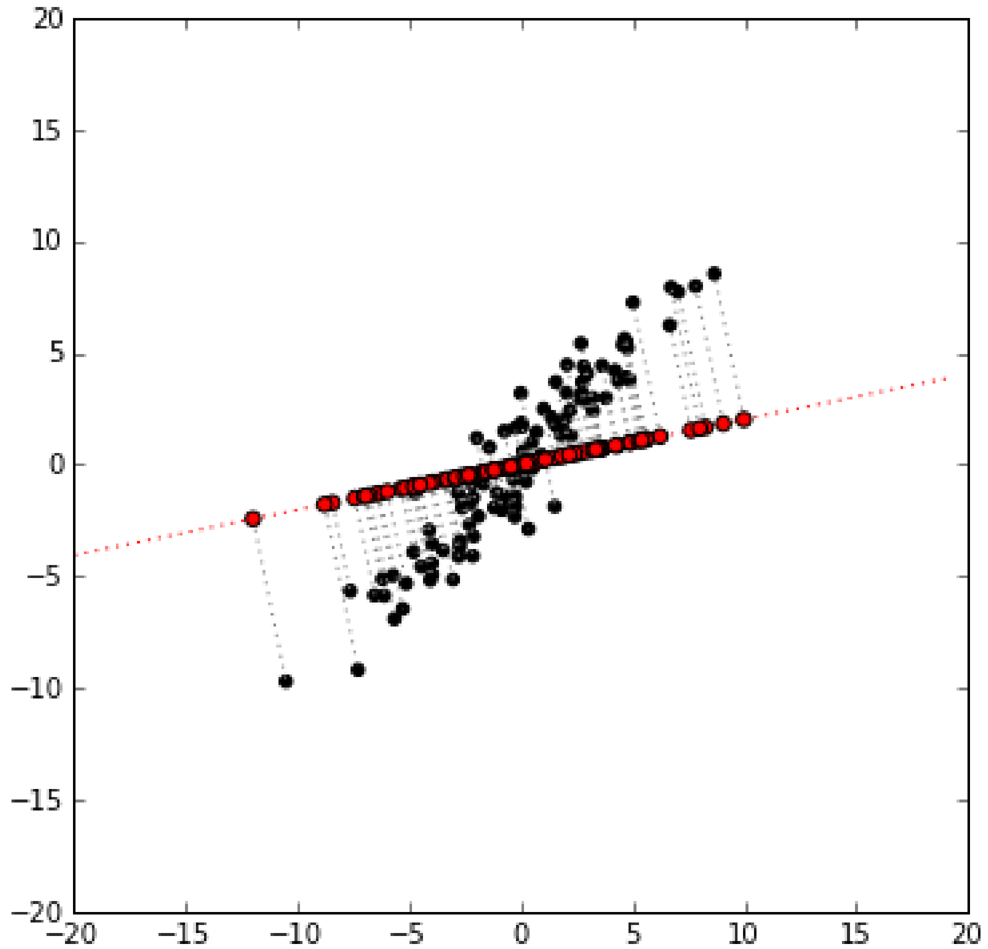
How do we find a good basis?



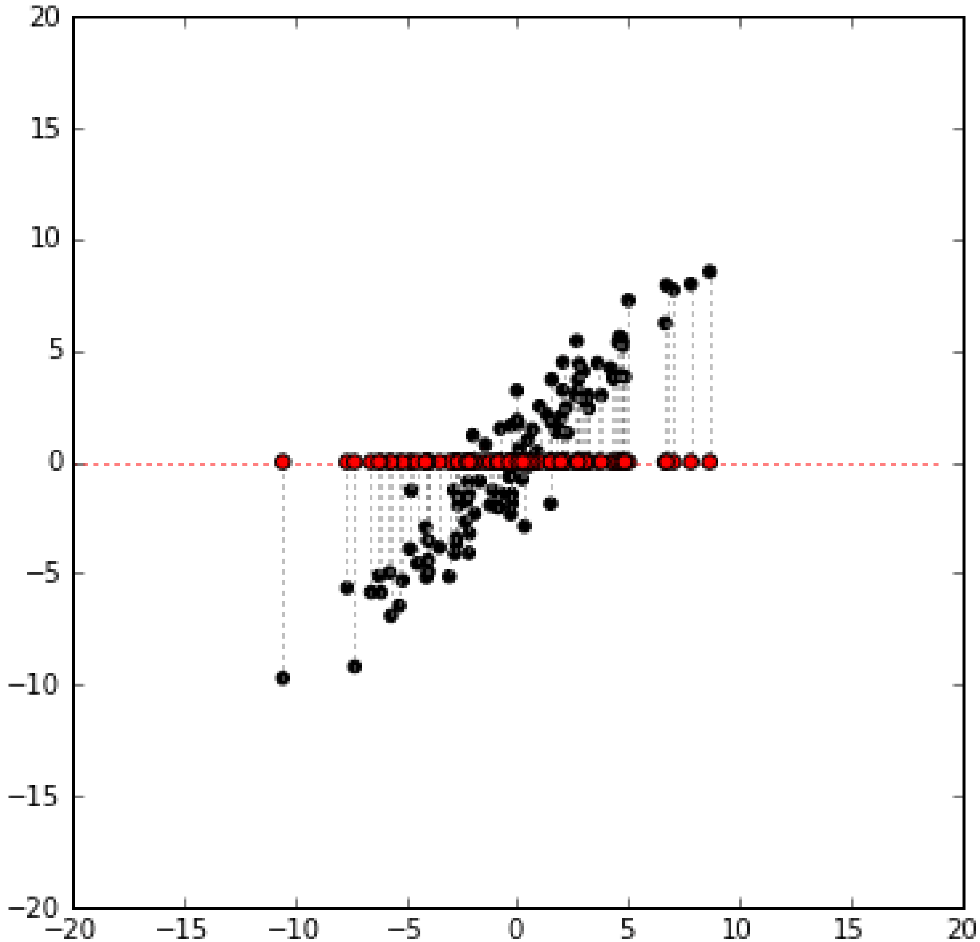
Linear projection



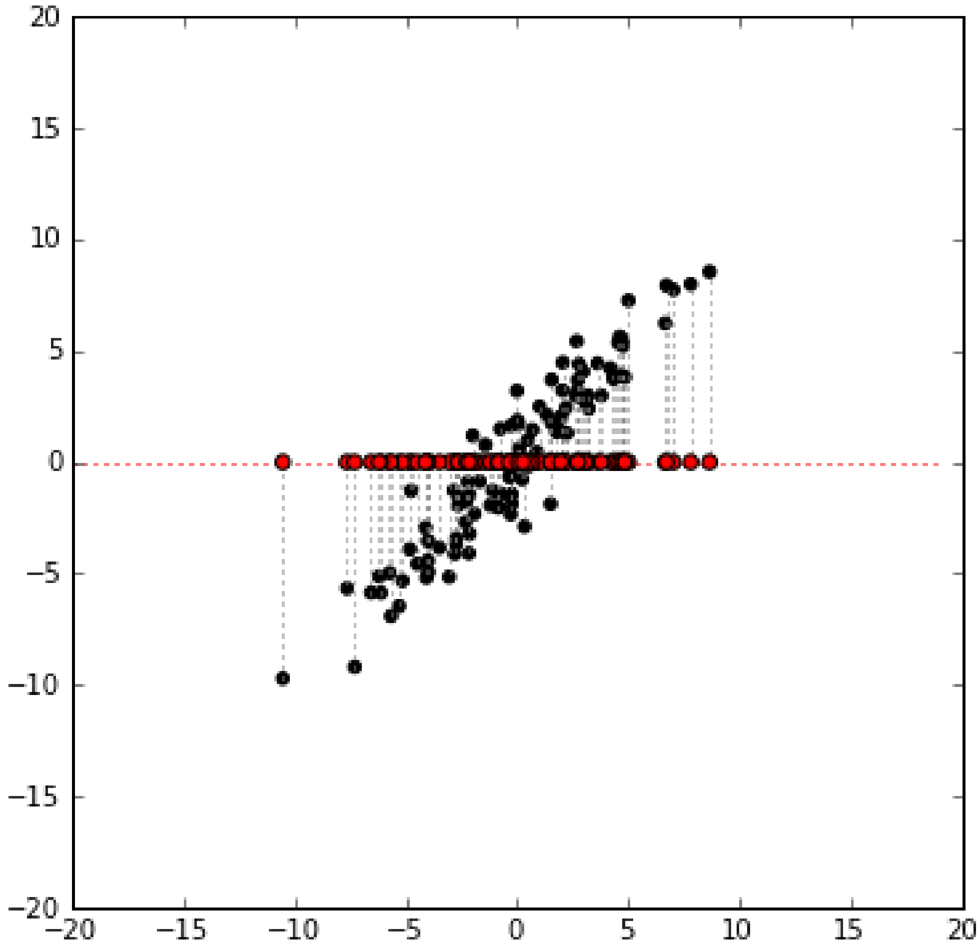
Linear projection



Linear projection



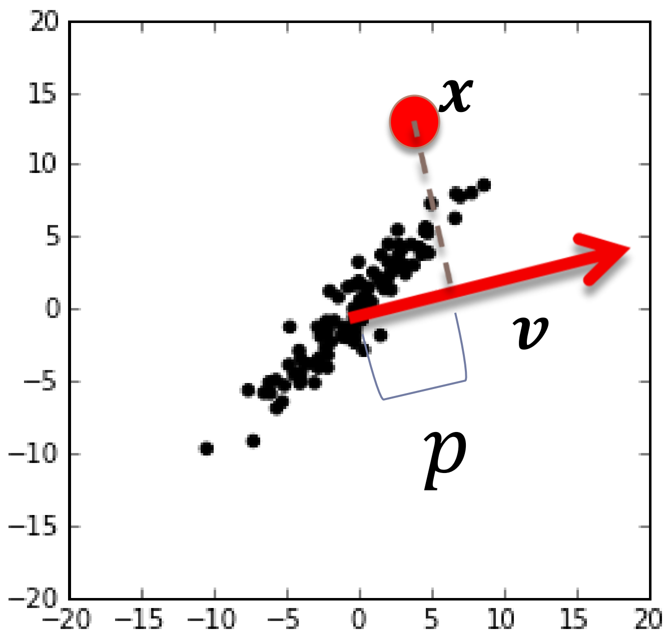
Linear projection



Idea: Maximize projection variance

- For a point $\boldsymbol{x}_i$ and a unit basis vector $\boldsymbol{v}$ the length of projection of $\boldsymbol{x}_i$ onto $\boldsymbol{v}$ is given by

$$p = \langle \boldsymbol{v}, \boldsymbol{x}_i \rangle = \boldsymbol{v}^T \boldsymbol{x}_i$$



Projection variance

$$p_i = \boldsymbol{v}^T \boldsymbol{x}_i$$

Projection variance

$$p_i = \boldsymbol{v}^T \boldsymbol{x}_i$$

$$\sigma_{\boldsymbol{v}}^2 = \frac{1}{n} \sum_i (p_i - \overline{p})^2$$

Projection variance

$$p_i = \boldsymbol{v}^T \boldsymbol{x}_i$$

$$\sigma_{\boldsymbol{v}}^2 = \frac{1}{n} \sum_i (p_i - \overline{p})^2$$

$$\boldsymbol{v} = \operatorname{argmax}_{\boldsymbol{v}} \sigma_{\boldsymbol{v}}^2$$

Projection variance

$$p_i = \boldsymbol{v}^T \boldsymbol{x}_i$$

$$\sigma_{\boldsymbol{v}}^2 = \frac{1}{n} \sum_i (p_i - \overline{p})^2$$

Pre-center data, so that $\overline{p} = \boldsymbol{v}^T \overline{\boldsymbol{x}} = 0$

Projection variance

$$p_i = \boldsymbol{v}^T \boldsymbol{x}_i$$

$$\sigma_{\boldsymbol{v}}^2 = \frac{1}{n} \sum_i (p_i)^2$$

Pre-center data, so that $\overline{p} = \boldsymbol{v}^T \overline{\boldsymbol{x}} = 0$

Projection variance

$$p_i = \boldsymbol{v}^T \boldsymbol{x}_i$$

$$\sigma_{\boldsymbol{v}}^2 = \frac{1}{n} \sum_i (p_i)^2 = \frac{1}{n} \|\boldsymbol{p}\|^2$$

Projection variance

$$p_i = \boldsymbol{v}^T \boldsymbol{x}_i$$

$$\sigma_{\boldsymbol{v}}^2 = \frac{1}{n} \sum_i (p_i)^2 = \frac{1}{n} \|\boldsymbol{p}\|^2$$

$$= \frac{1}{n} \|\boldsymbol{X}\boldsymbol{v}\|^2$$

Projection variance

$$p_i = \boldsymbol{v}^T \boldsymbol{x}_i$$

$$\sigma_{\boldsymbol{v}}^2 = \cdots$$
$$= \frac{1}{n} \|\boldsymbol{X}\boldsymbol{v}\|^2 = \frac{1}{n} (\boldsymbol{X}\boldsymbol{v})^T (\boldsymbol{X}\boldsymbol{v})$$
$$= \frac{1}{n} \boldsymbol{v}^T \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{v} = \boldsymbol{v}^T \boldsymbol{\Sigma} \boldsymbol{v}$$

Projection variance

$$p_i = \boldsymbol{v}^T \boldsymbol{x}_i$$

$$\sigma_{\boldsymbol{v}}^2 = \boldsymbol{v}^T \boldsymbol{\Sigma} \boldsymbol{v}$$

Projection variance

$$p_i = \boldsymbol{v}^T \boldsymbol{x}_i$$

$$\sigma_{\boldsymbol{v}}^2 = \boldsymbol{v}^T \boldsymbol{\Sigma} \boldsymbol{v}$$

Data covariance matrix

$X^T X$

Objective function

$$\operatorname{argmax}_v \boldsymbol{v}^T \boldsymbol{\Sigma} \boldsymbol{v}$$

$$s.t. \|\boldsymbol{v}\|^2 = 1$$

Optimization

$$\operatorname{argmax}_{v}\ \boldsymbol{v}^T \boldsymbol{\Sigma} \boldsymbol{v}$$

$$s.t.\ \|\boldsymbol{v}\|^2 = 1$$

Method of Lagrange multipliers…

$$\boldsymbol{\Sigma v} = \lambda \boldsymbol{v}$$

Optimization

$$\operatorname{argmax}_{v}\ \boldsymbol{v}^T \boldsymbol{\Sigma} \boldsymbol{v}$$

$$s.t.\ ||\boldsymbol{v}||^2 = 1$$

Method of Lagrange multipliers…

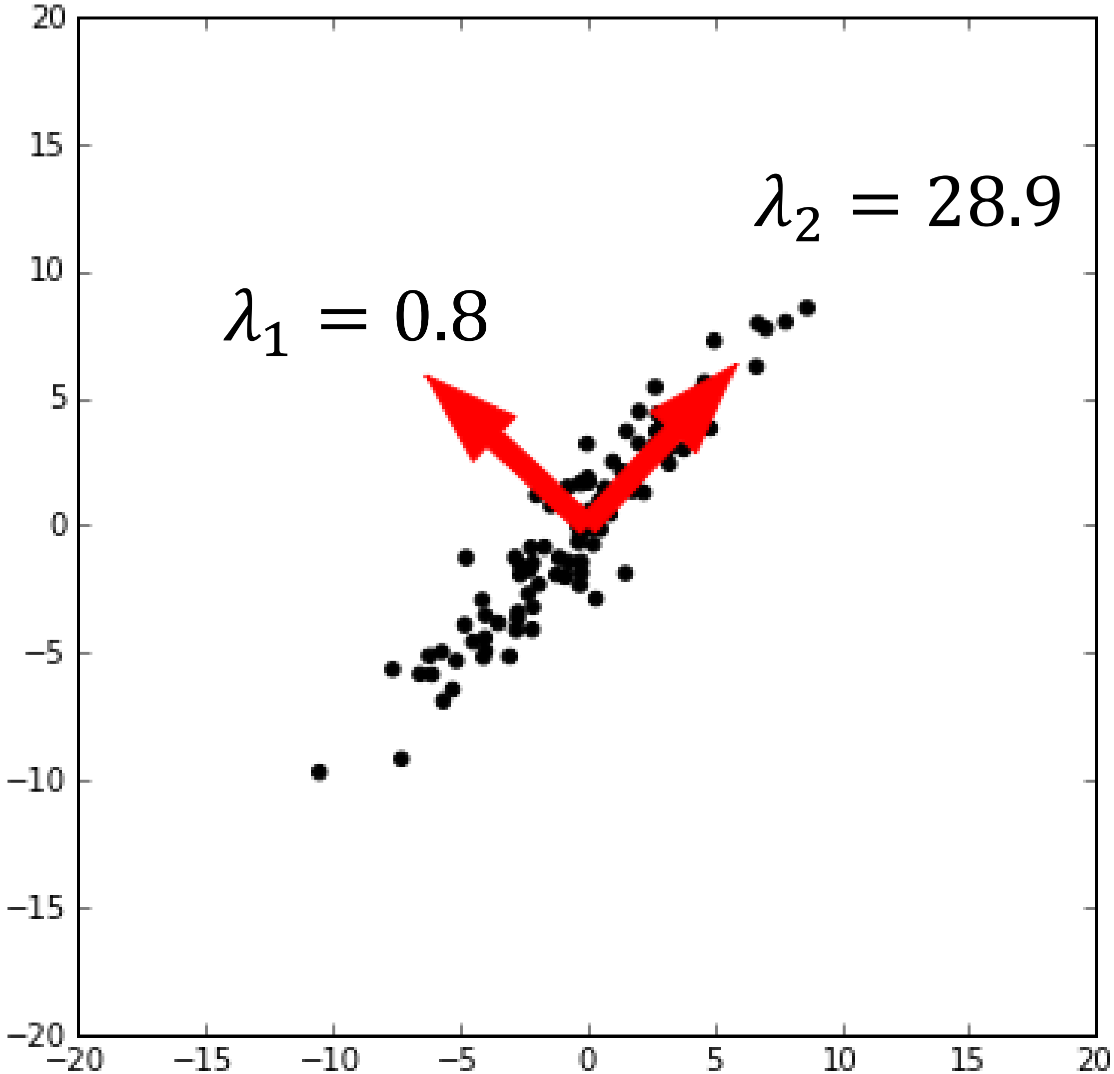
$$\boldsymbol{\Sigma}\boldsymbol{v} = \lambda \boldsymbol{v}$$

Eigenvector of Σ **Eigenvalue**

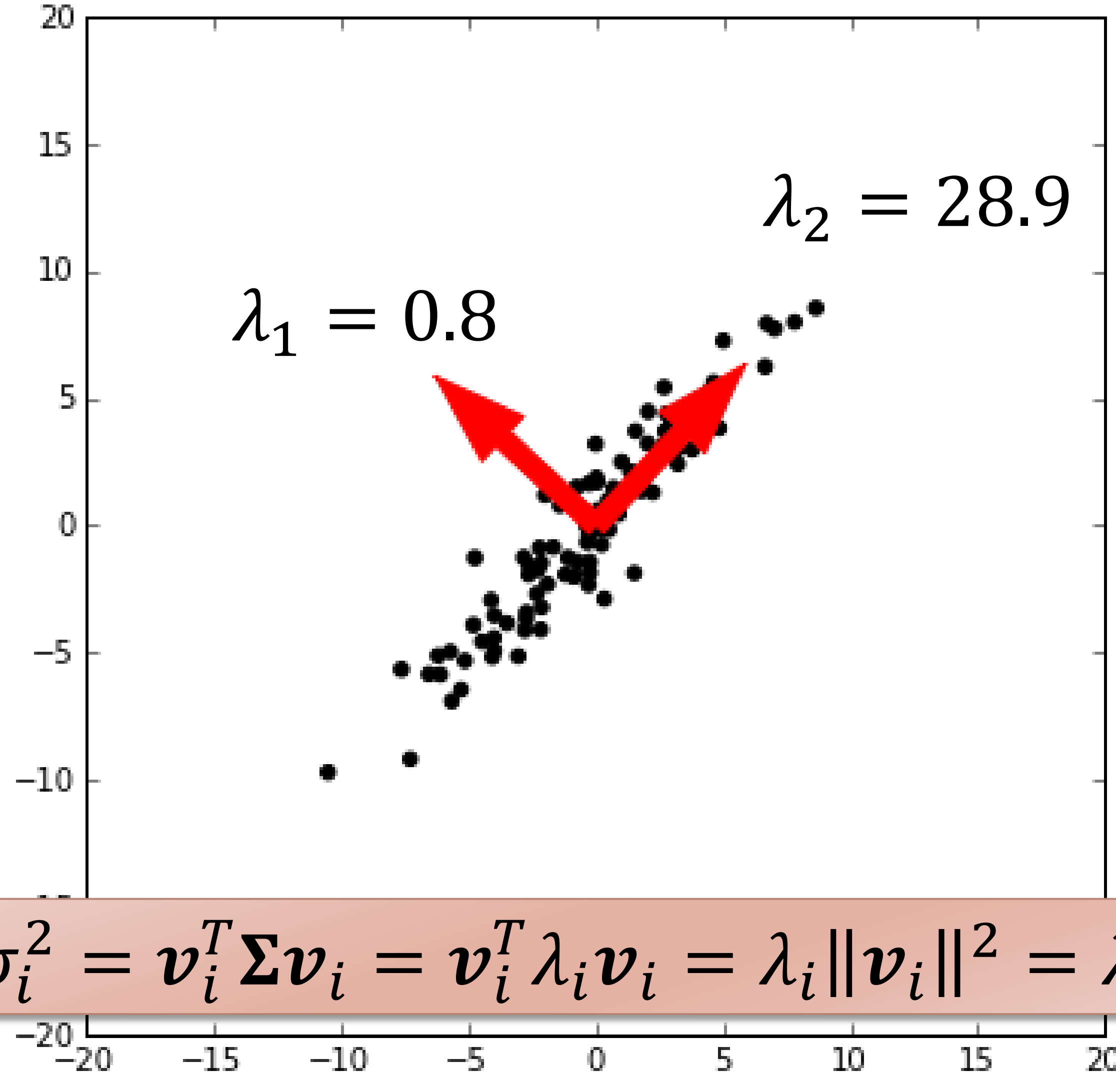
Example

```
Xc = X - mean(X, axis=0)

Sigma = Xc.T * Xc / n

                (= cov(Xc, rowvar=0))

lambdas, vs = eigh(Sigma)
```

Example

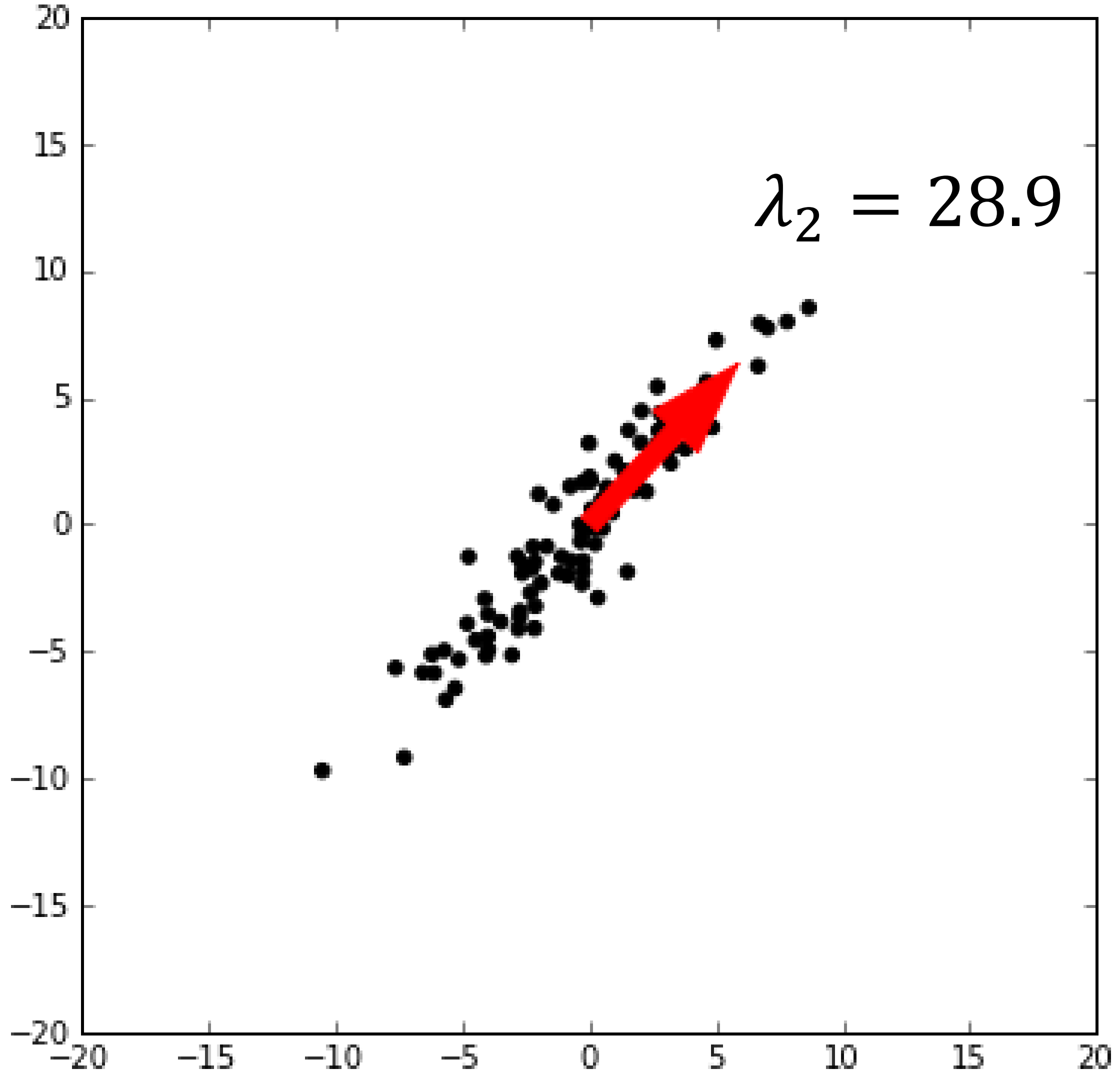


Example

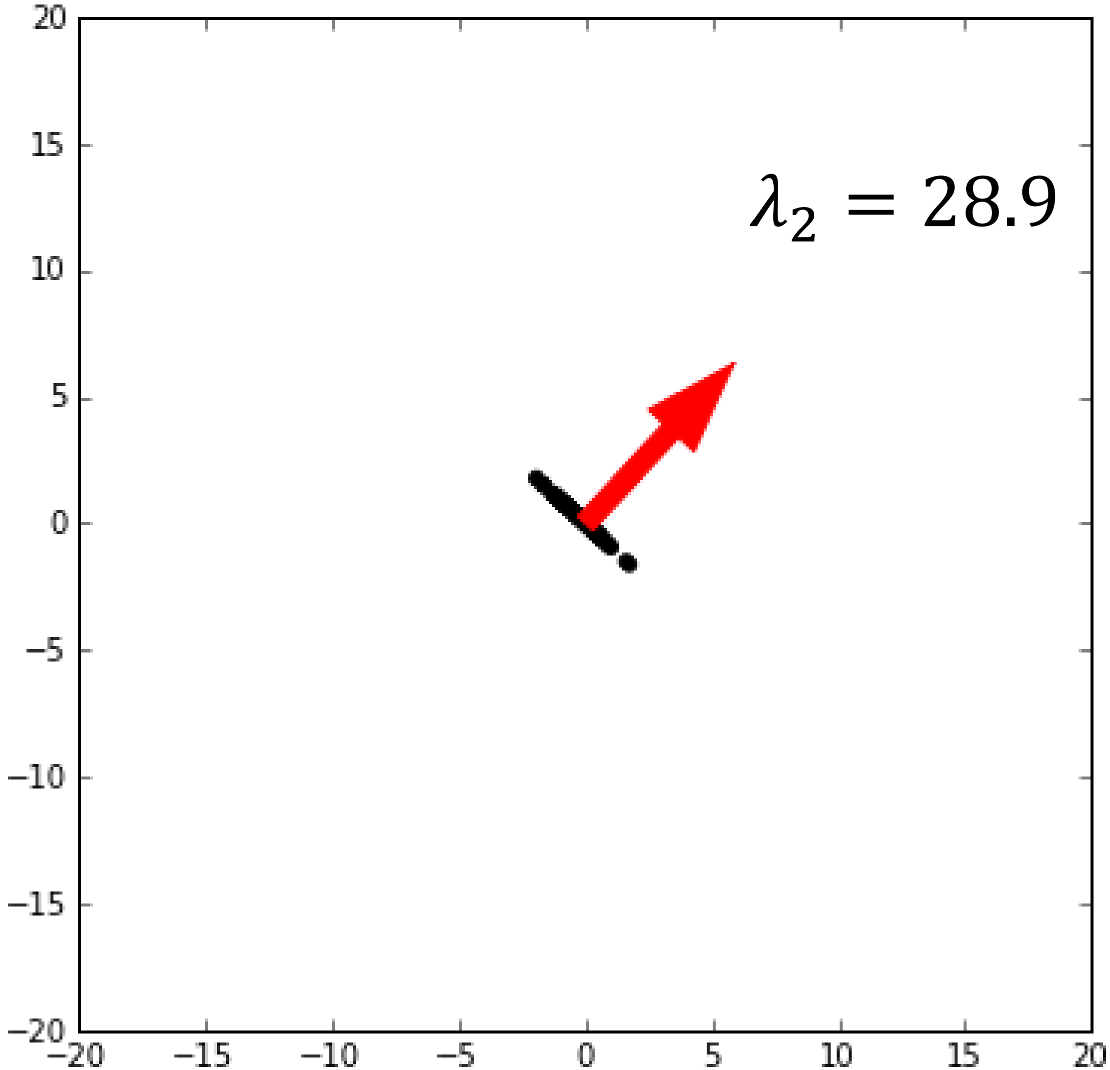


$$\sigma_i^2 = \boldsymbol{v}_i^T \boldsymbol{\Sigma} \boldsymbol{v}_i = \boldsymbol{v}_i^T \lambda_i \boldsymbol{v}_i = \lambda_i \|\boldsymbol{v}_i\|^2 = \lambda_i$$

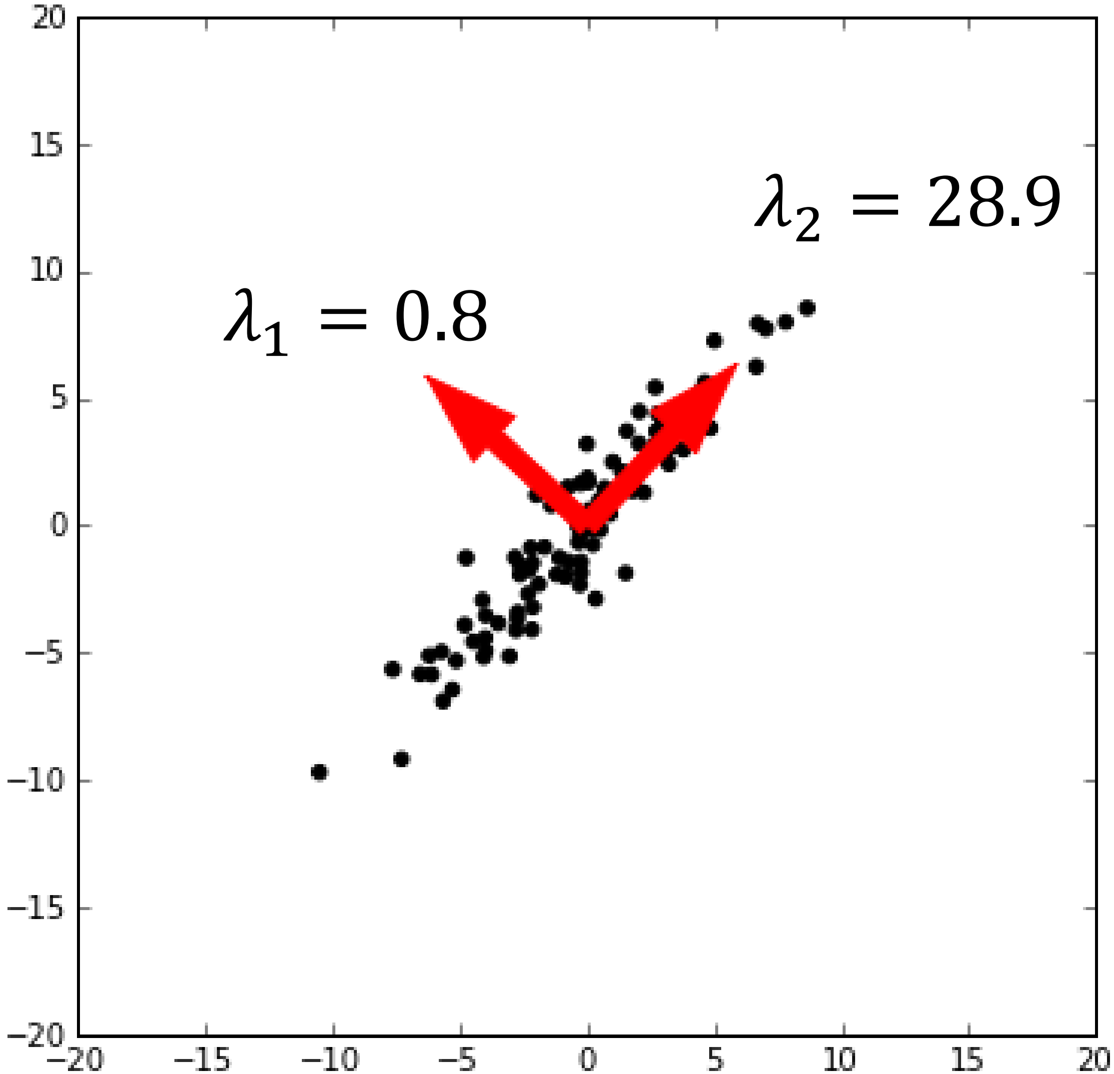
Example



Example



Example



Principal Components Analysis

Principal components are the **eigenvectors** of the **covariance matrix**.

$$V, \lambda = \mathbf{eig}(\boldsymbol{\Sigma})$$

Principal Components Analysis

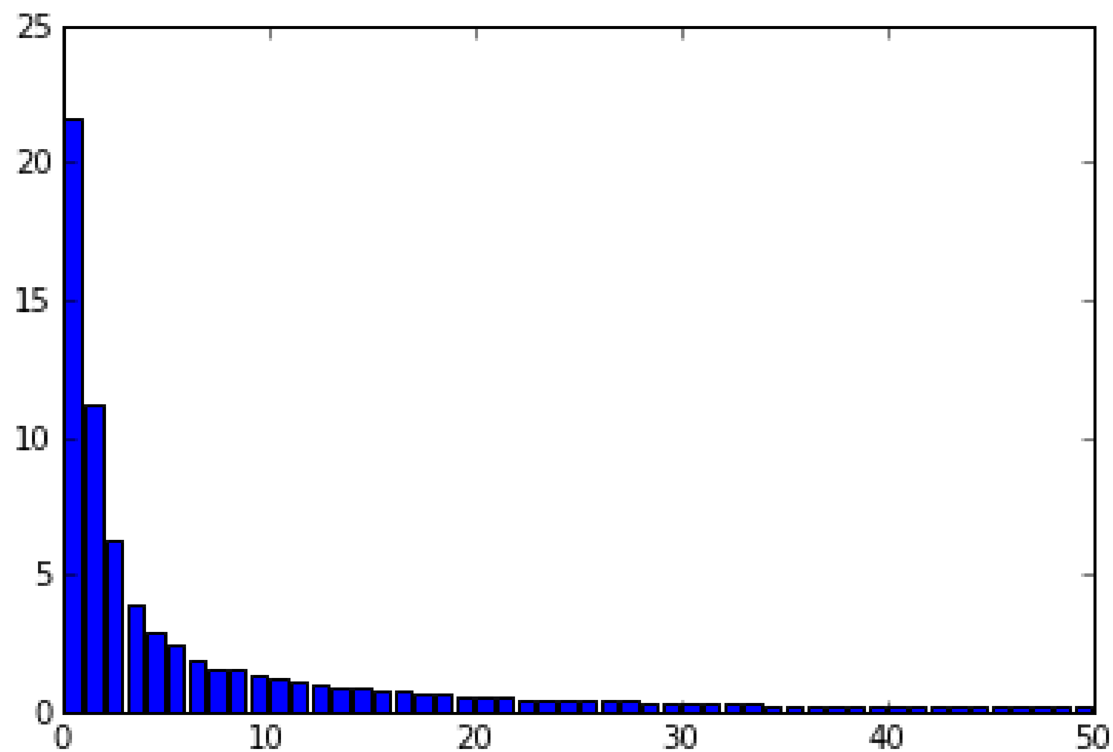
Principal components are the **eigenvectors** of the **covariance matrix**.

$$\boldsymbol{V}, \boldsymbol{\lambda} = \mathbf{eig}(\boldsymbol{\Sigma})$$

For each PC, the corresponding eigenvalue λ_i shows the **amount of variance explained** by the component.

Principal Components Analysis

Eigenvalue spectrum of Σ



Principal Components Analysis

Data projection onto PC i:

$$\boldsymbol{p} = \boldsymbol{X}\boldsymbol{v}_i$$

Data projection onto multiple PCs:

$$\boldsymbol{X}_{\text{proj}} = \boldsymbol{X}\boldsymbol{V}_*$$

Data reconstruction from PC coordinates:

$$\boldsymbol{X}_{\text{proj}}\boldsymbol{V}_*^T = \boldsymbol{X}$$

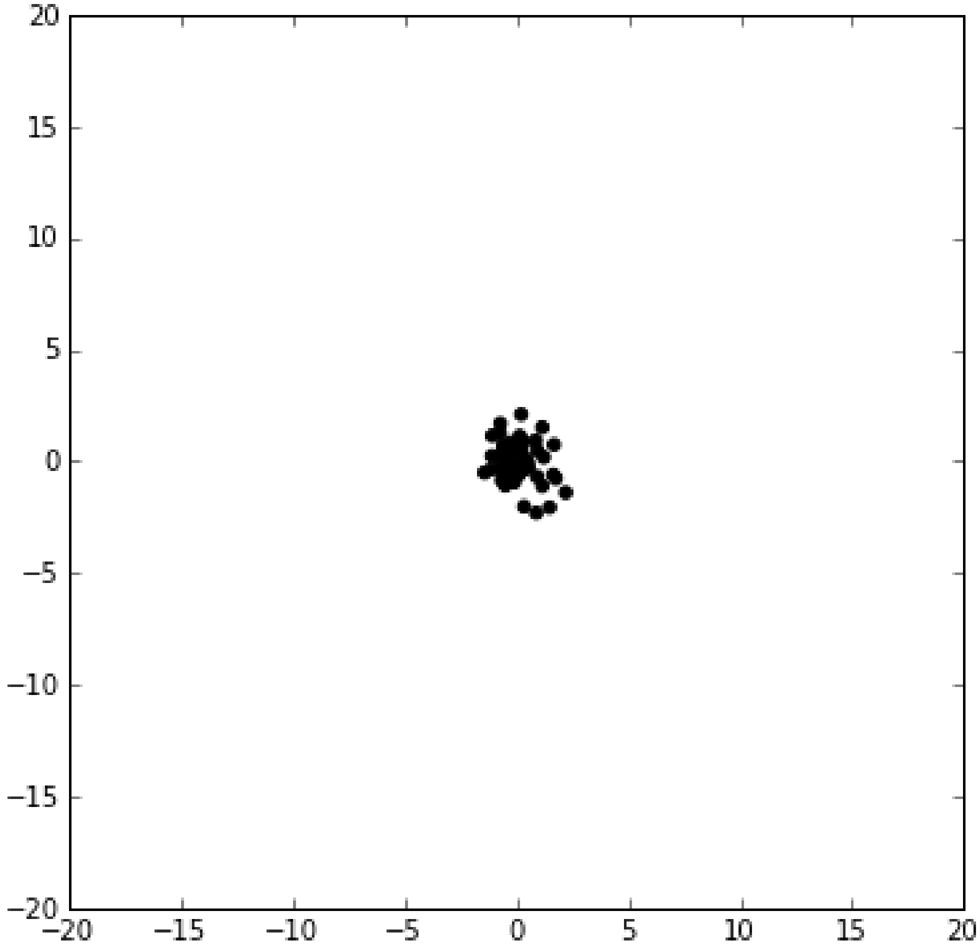
SKLearn's PCA

```
from sklearn.decomposition
                              import PCA

model = PCA(n_components=2)
model.fit(X)
X_t = model.transform(X)

model.components_[1,:]
```

SKLearn's PCA

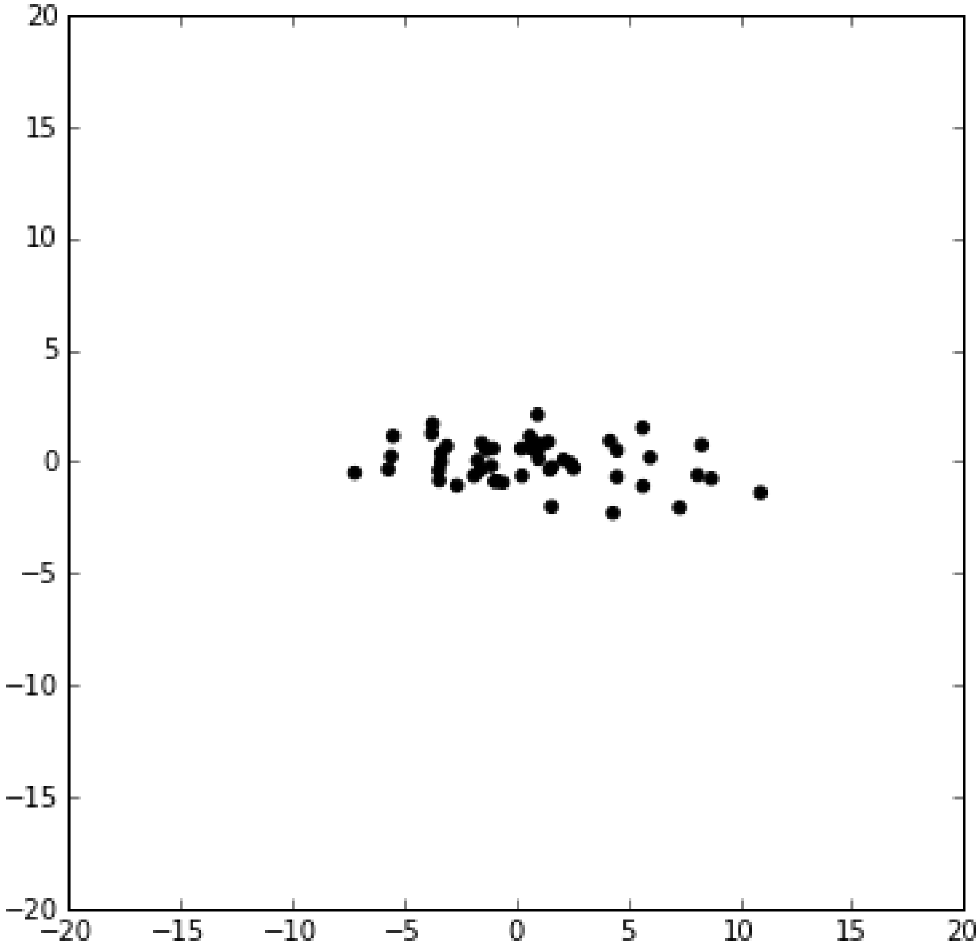
```
from sklearn.decomposition
import
    PCA,
    SparsePCA,
    ProbabilisticPCA,
    KernelPCA,
    FastICA,
    NMF,
    DictionaryLearning,
    ...
```

PCA: Geometric intuition



$$\boldsymbol{X} \sim N(0,1)$$

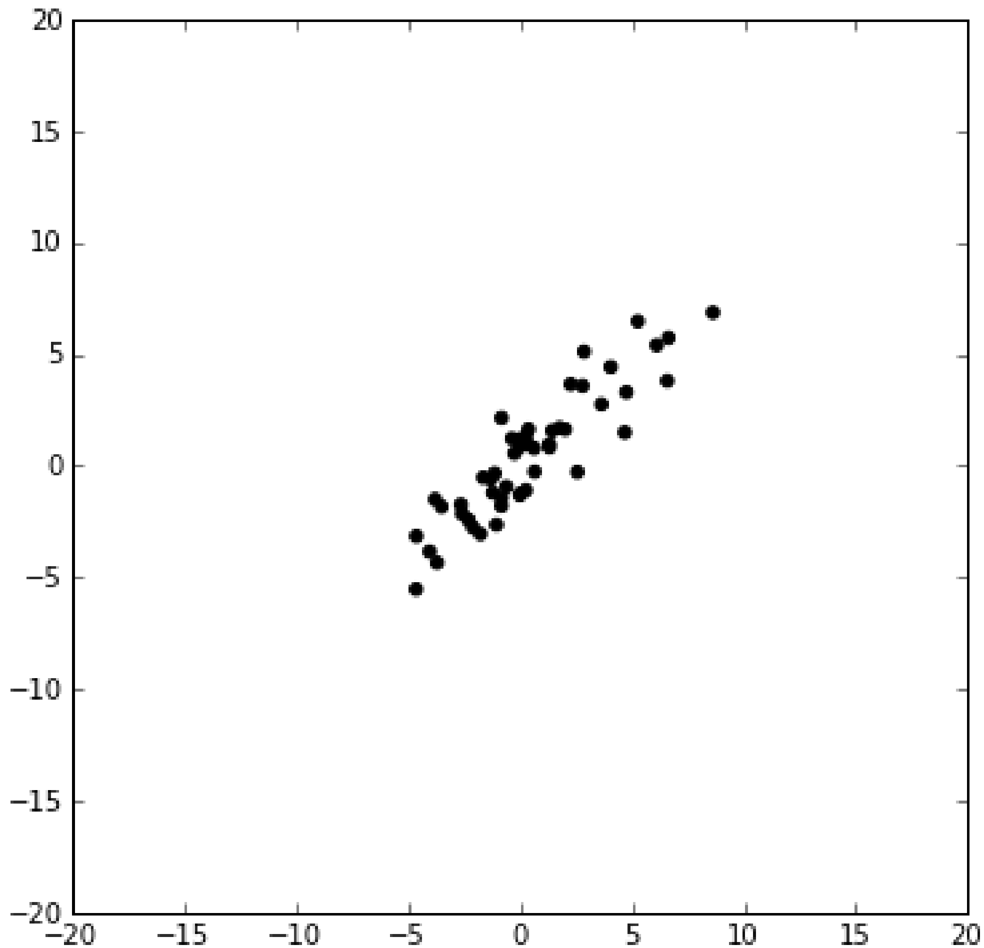
PCA: Geometric intuition



$$\boldsymbol{X} \sim N(0,1)$$

$$\boldsymbol{X}' = \boldsymbol{X}\begin{pmatrix} 5 & 0 \\ 0 & 0.9 \end{pmatrix}$$

PCA: Geometric intuition

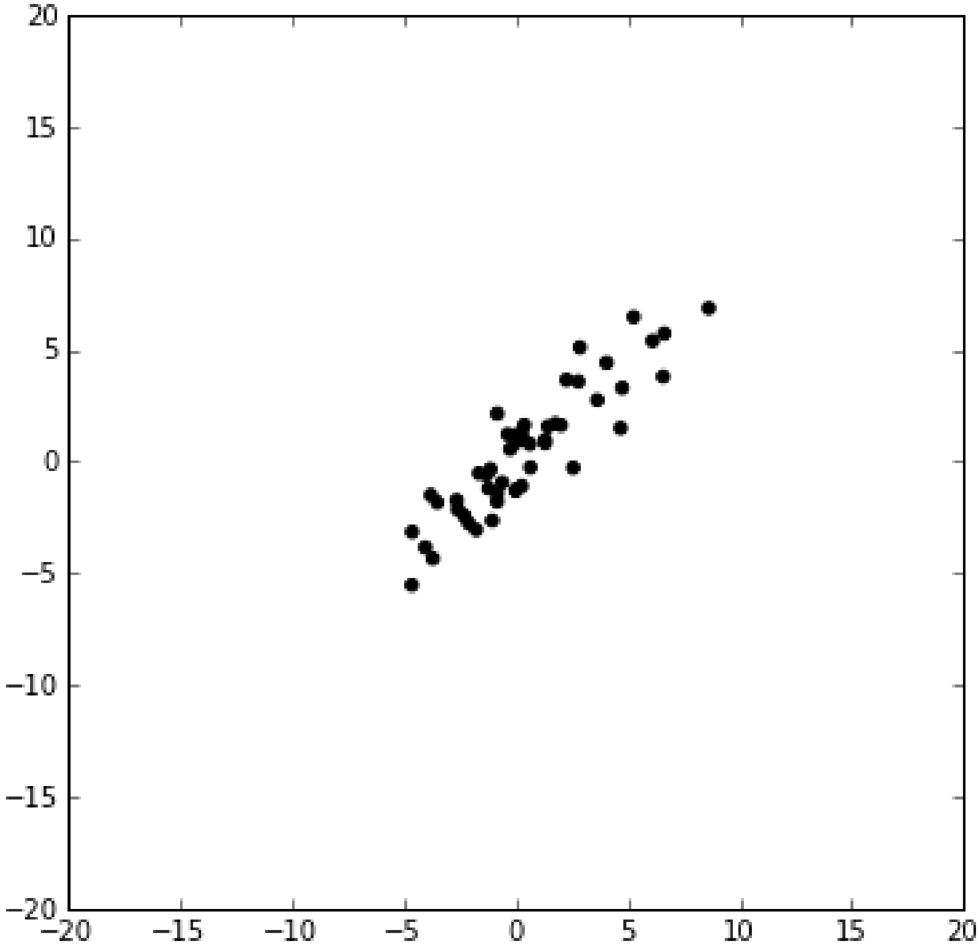


$$\boldsymbol{X} \sim N(0,1)$$

$$\boldsymbol{X}' = \boldsymbol{X}\begin{pmatrix} 5 & 0 \\ 0 & 0.9 \end{pmatrix}$$

$$\boldsymbol{X}'' = \boldsymbol{X}'\begin{pmatrix} \cos 0.8 & -\sin 0.8 \\ \sin 0.8 & \cos 0.8 \end{pmatrix}$$

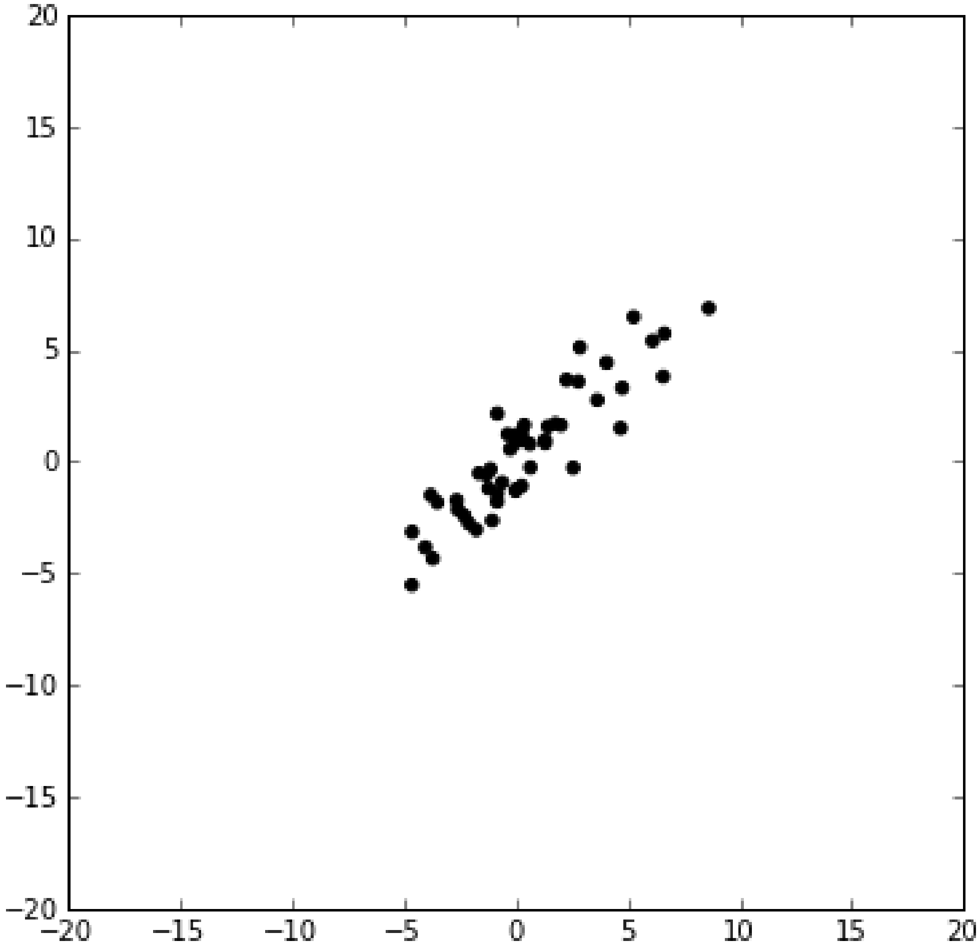
PCA: Geometric intuition



$$X \sim N(0,1)$$

$$X'' = X \cdot D \cdot R$$

PCA: Geometric intuition

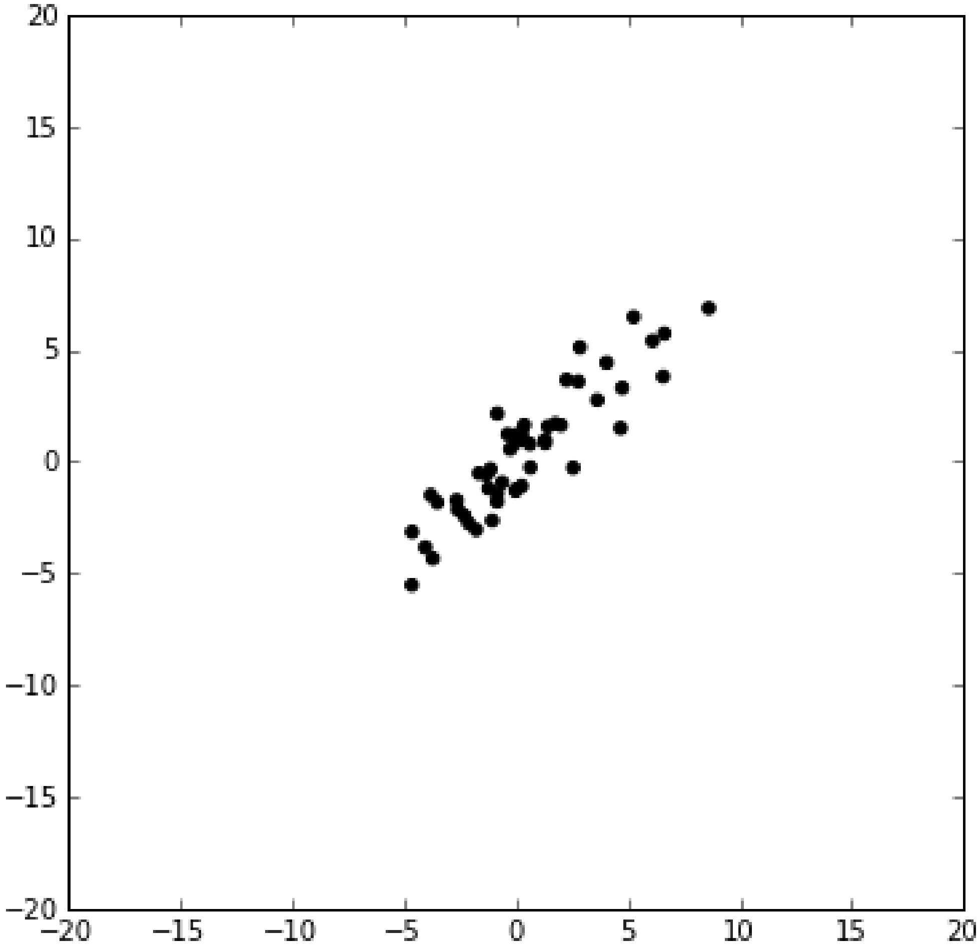


$$\boldsymbol{X} \sim N(0,1)$$

$$\boldsymbol{X''} = \boldsymbol{X} \cdot \boldsymbol{D} \cdot \boldsymbol{R}$$

$$(\boldsymbol{X''})^T(\boldsymbol{X''}) = (\boldsymbol{XDR})^T(\boldsymbol{XDR})$$

PCA: Geometric intuition

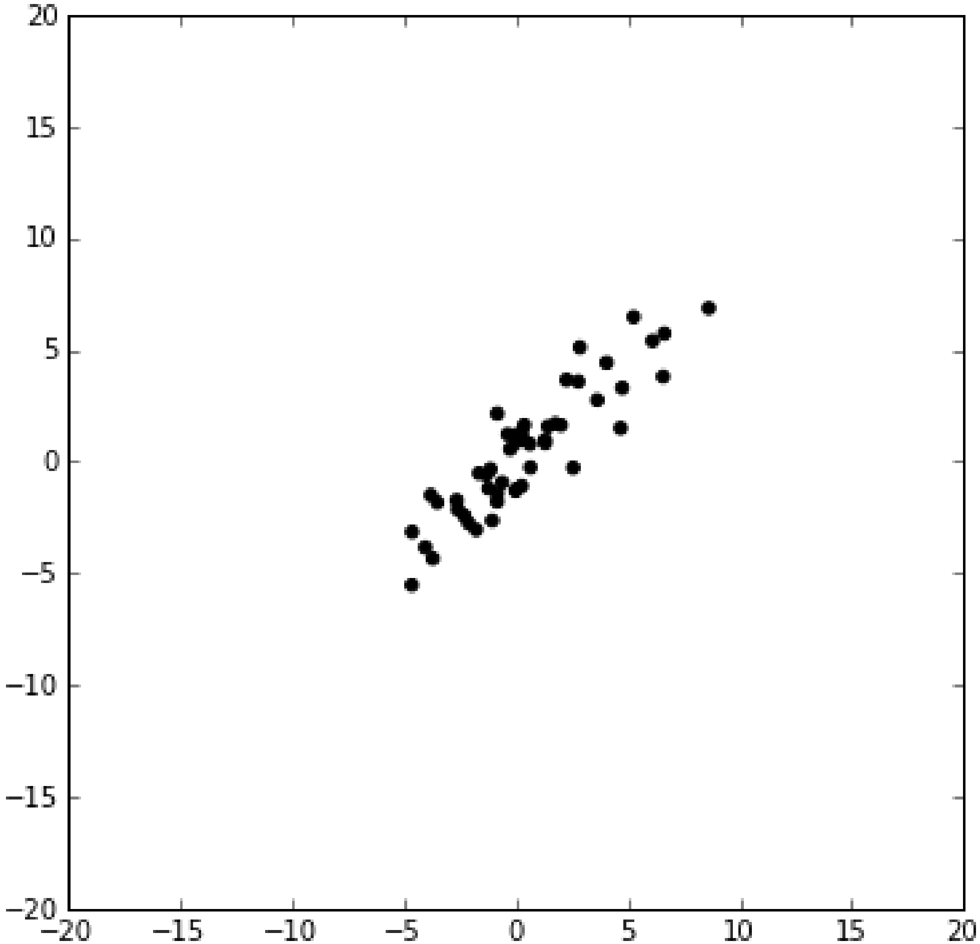


$$X \sim N(0,1)$$

$$X'' = X \cdot D \cdot R$$

$$(X'')^T (X'') = (XDR)^T (XDR) = R^T D^T X^T X D R$$

PCA: Geometric intuition

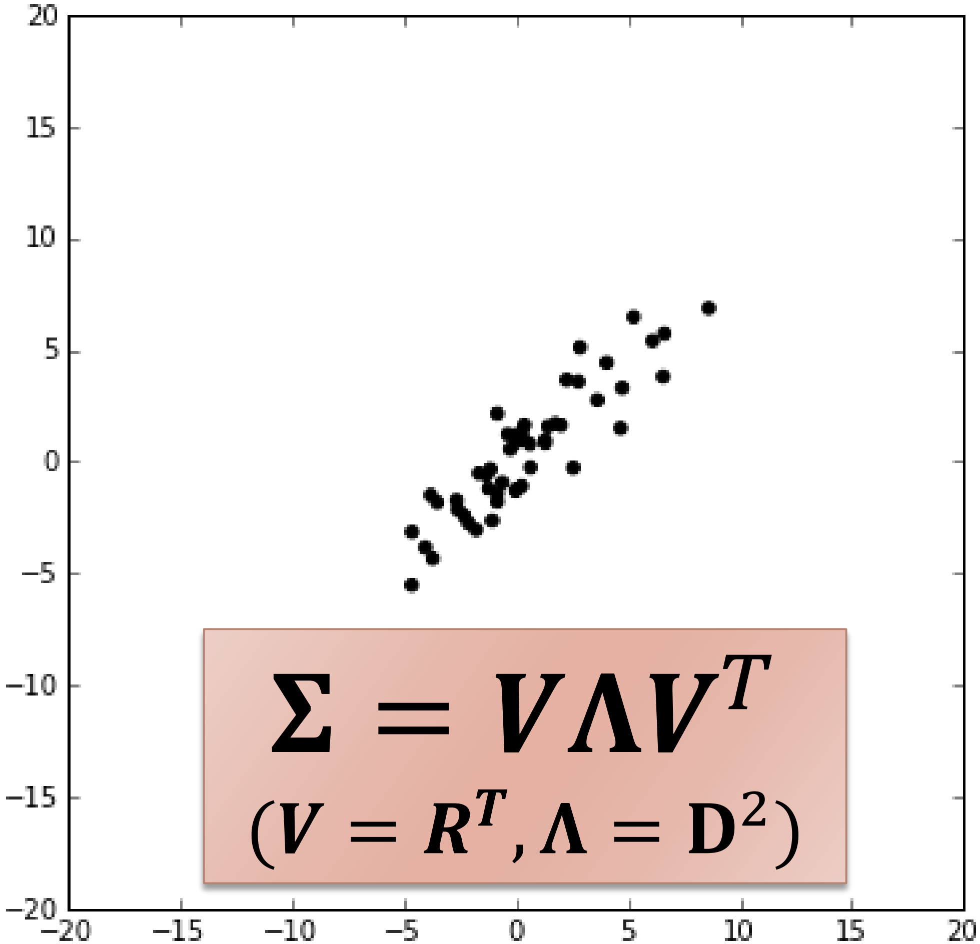


$$\boldsymbol{X} \sim N(0,1)$$

$$\boldsymbol{X''} = \boldsymbol{X} \cdot \boldsymbol{D} \cdot \boldsymbol{R}$$

$$(\boldsymbol{X''})^T(\boldsymbol{X''})$$
$$= (\boldsymbol{XDR})^T(\boldsymbol{XDR})$$
$$= \boldsymbol{R}^T\boldsymbol{D}^T\boldsymbol{X}^T\boldsymbol{XDR}$$
$$= \boldsymbol{R}^T\boldsymbol{D}^2\boldsymbol{R}$$

PCA: Geometric intuition



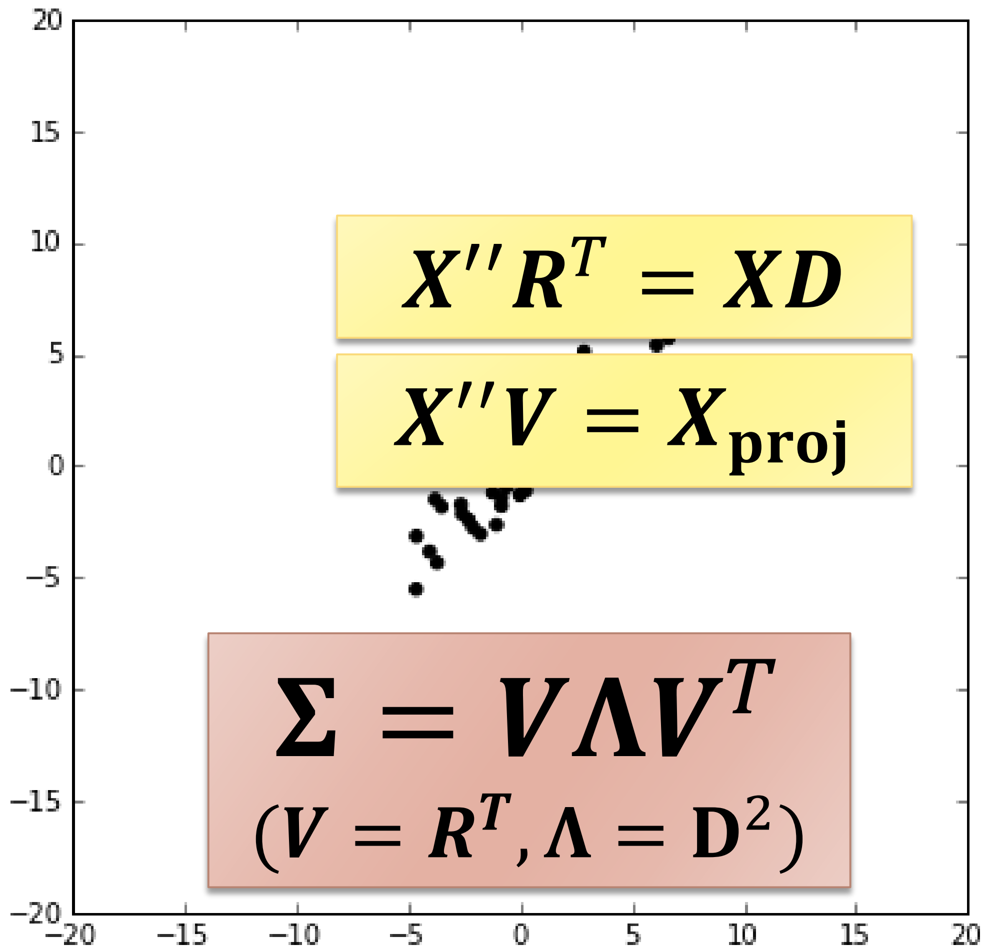
$$\Sigma = V \Lambda V^T$$
$$(V = R^T, \Lambda = \mathrm{D}^2)$$

$$X \sim N(0,1)$$

$$X'' = X \cdot D \cdot R$$

$$(X'')^T (X'') = (XDR)^T (XDR) = R^T D^T X^T X D R = R^T D^2 R$$

PCA: Geometric intuition



$$X''R^T = XD$$

$$X''V = X_{\mathrm{proj}}$$

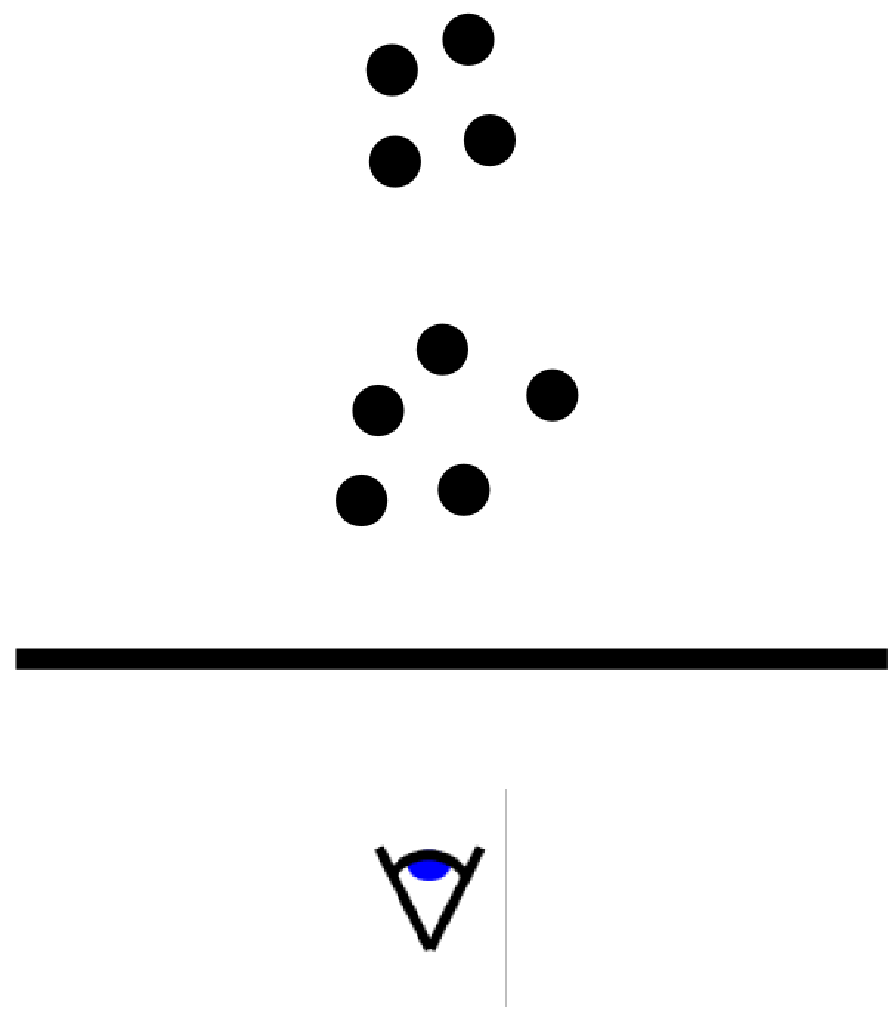
$$\Sigma = V\Lambda V^T$$

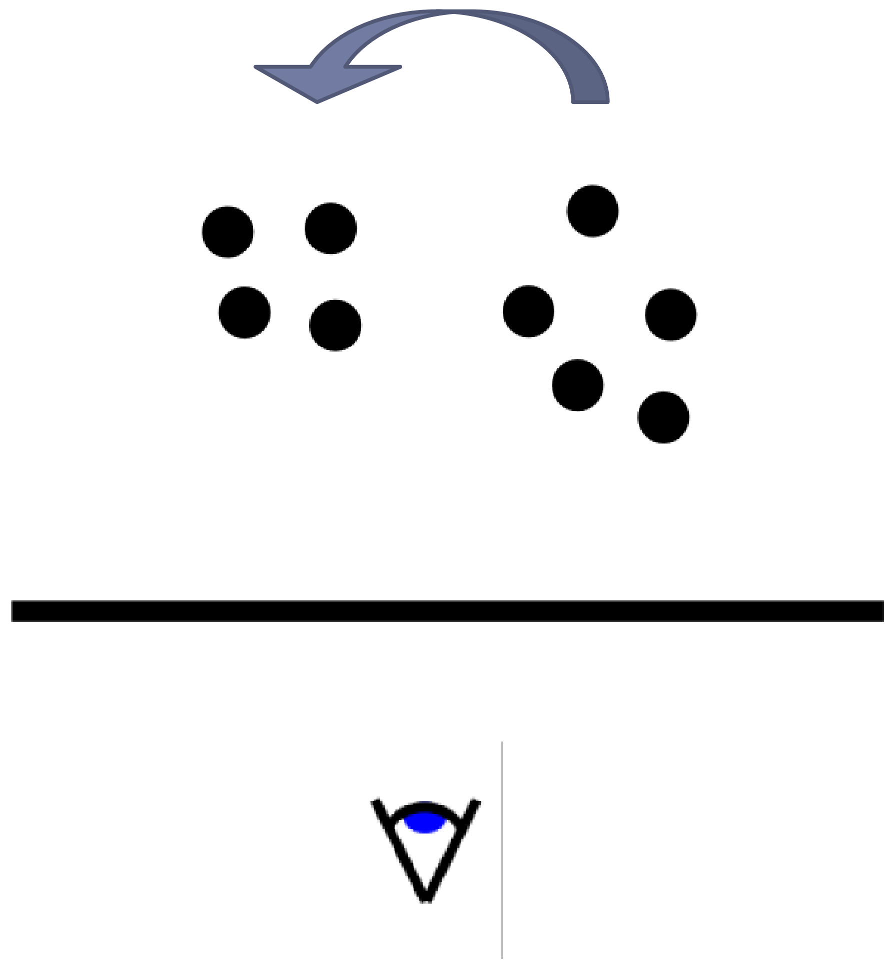
$$(V = R^T, \Lambda = \mathrm{D}^2)$$

$$X \sim N(0,1)$$

$$X'' = X \cdot D \cdot R$$

$$(X'')^T(X'') = (XDR)^T(XDR) = R^T D^T X^T X D R = R^T D^2 R$$





Quiz

- Principal components are ______________ of the ______________ matrix.

- Eigenvalue spectrum shows how much ______________ is explained by each _______.

- If $\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$, then

$$\boldsymbol{X}_{\mathbf{proj}} = \_\_\_\_\_\_$$