Machine learning

Dimensionality

Mohammad-Reza A. Dehaqani

dehaqani@ut.ac.ir

Dimensionality Reduction

$$x \in \mathbb{R}^d \longrightarrow z \in \mathbb{R}^k \qquad k < d$$

$$\boxed{g(x) = w^T x}$$

$x \in \mathbb{R}^d \qquad w \in \mathbb{R}^d$

$$w = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$

$\text{Error} = \text{Bias} + \underline{\text{Variance}} + \epsilon$

Bayes error

Curse of Dimensionality

$N(\mu, \Sigma)$ $\quad x \in \mathbb{R}^d$ $\quad N$

$\mu \to d$ $\quad \Sigma \to d \times d$ $\quad O(d^2 + d)$

$d = 100 \longrightarrow 10000$ $\quad O(100)$

$$x = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad P = \begin{bmatrix} \epsilon \\ \epsilon \\ \vdots \\ \epsilon \end{bmatrix} \qquad \epsilon\sqrt{d}$$

$$g(x) = \omega^T x$$

$$\underset{n \times d}{X} =$$

$$\underset{d \times 1}{\omega^*} = (X^T X)^{-1} X^T y$$

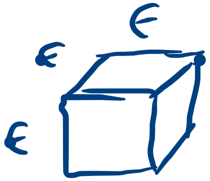
$$\underset{d \times n}{X^T}$$

$$X \quad (X^T X)_{d \times d}$$

$$\boxed{d = 10000}$$

$$O(d^3)$$

$$\boxed{n < d}$$



$V = \epsilon^d$

$0 < \epsilon < 1$

$d \to \infty$

$$\boxed{V \to 0}$$

Filter ₹

Dimensionality Reduction

wrapper

Feature selection

Feature Reduction

Supervised

unsupervised

$I(x^i ; y)$

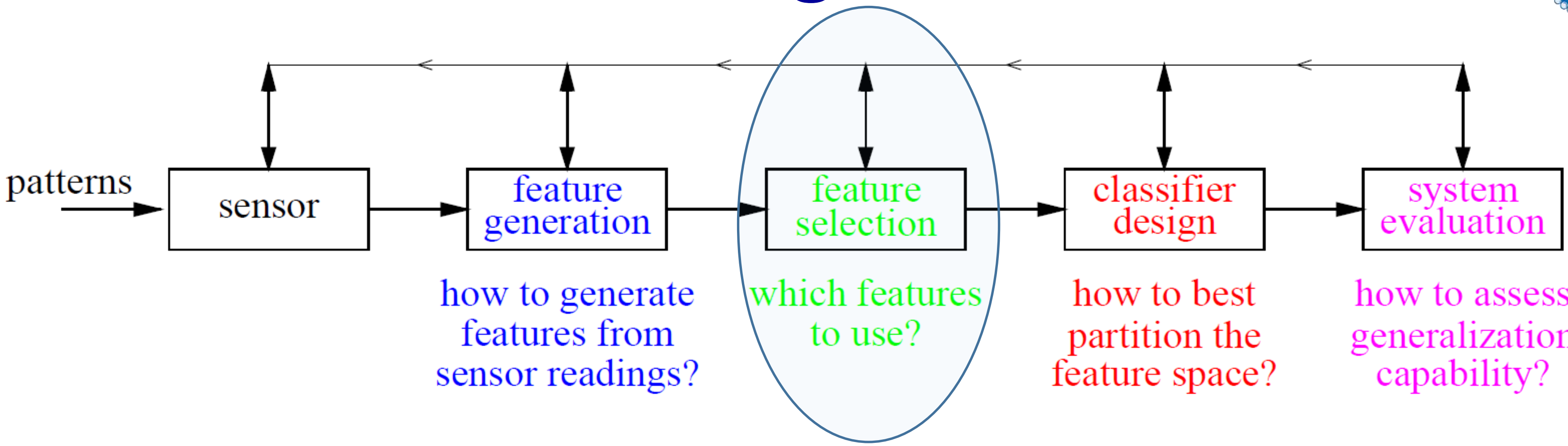
Correlation

$x_1 \quad x_2 \quad x_3$

$z_1 = x_1 + x_2$

$z_2 = x_1 + x_3 - 2x_2$

Feature conditioning



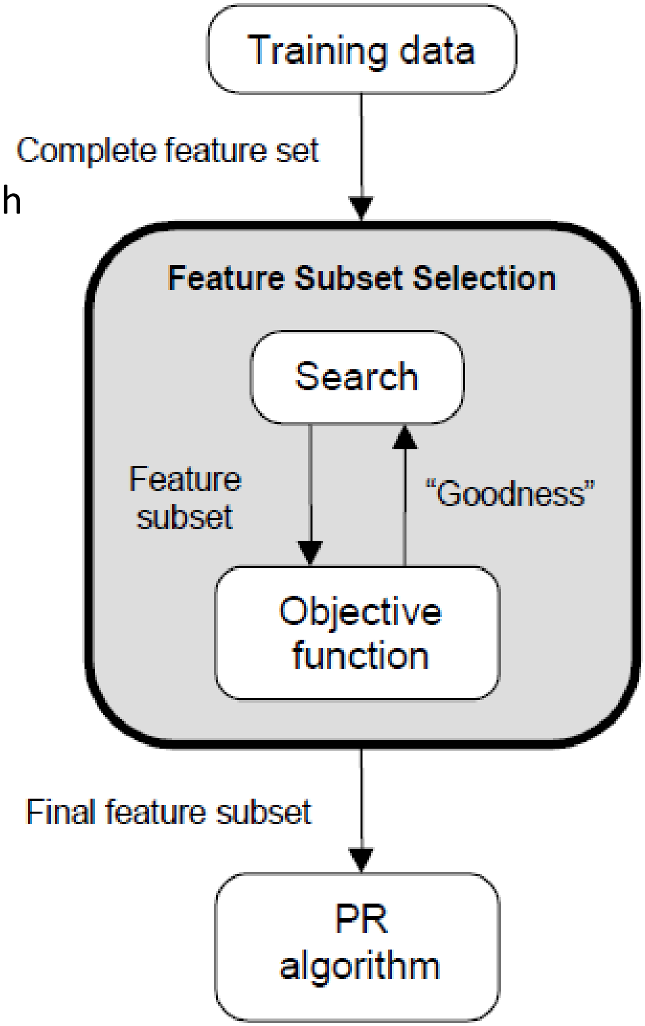
- In practical multicategory applications, it **is not unusual** to encounter problems involving **hundreds of features**.
- Feature selection:
  - Using a **criterion function** that is often a function of the classification error for feature selection (to select **discriminative** and **invariant** features)
- Feature reduction:
  - Using linear or non-linear **combinations of features** is feature selection that reduces dimensionality by selecting subsets of existing features.

Exhaustive search

- Examining all $\binom{d}{m}$ possible **subsets** of size m, and selecting the subset that **performs the best** according to the criterion function.
- The number of subsets grows **combinatorially**, making the exhaustive search **impractical**.
- Iterative procedures are **often used** but they **cannot guarantee** the selection of the **optimal** subset.

Feature Selection Steps

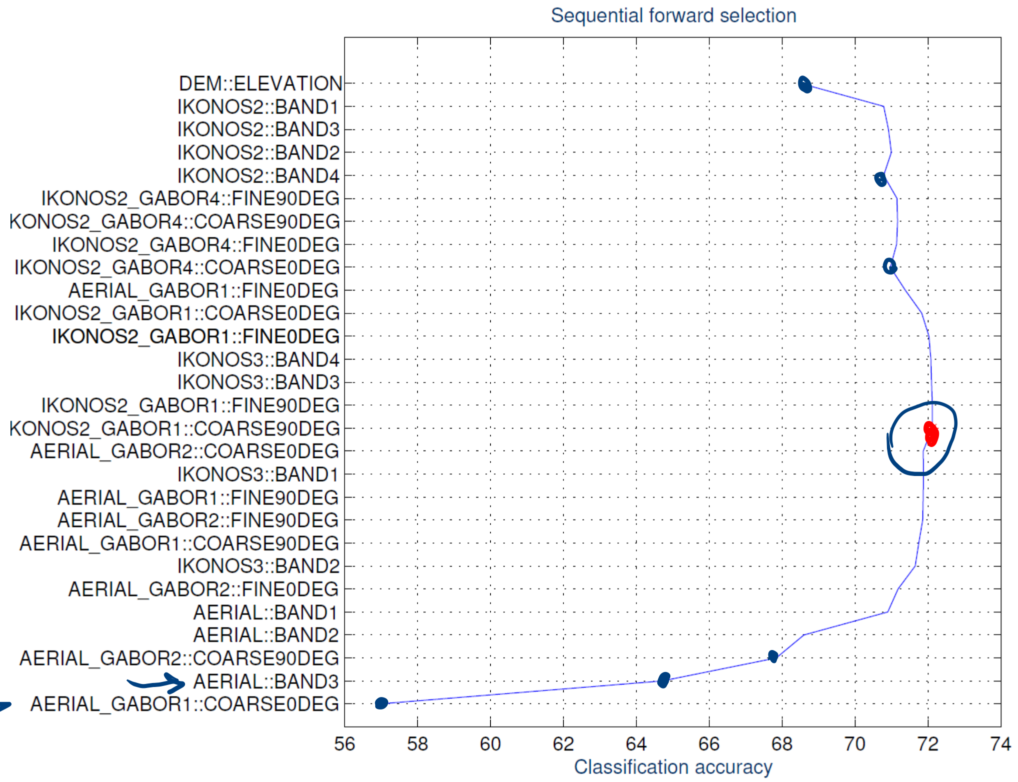
- Feature selection is an **optimization** problem.
  - **Step 1:** Search the **space of possible feature** subsets.
  - **Step 2:** **Pick** the subset that is optimal or near-optimal with respect to some **objective** function.



- **Search strategies**
  - Exhaustive
  - Heuristic
  - Randomized
- **Evaluation** strategies
  - Filter methods
  - Wrapper methods

Sequential forward selection: SFS

features



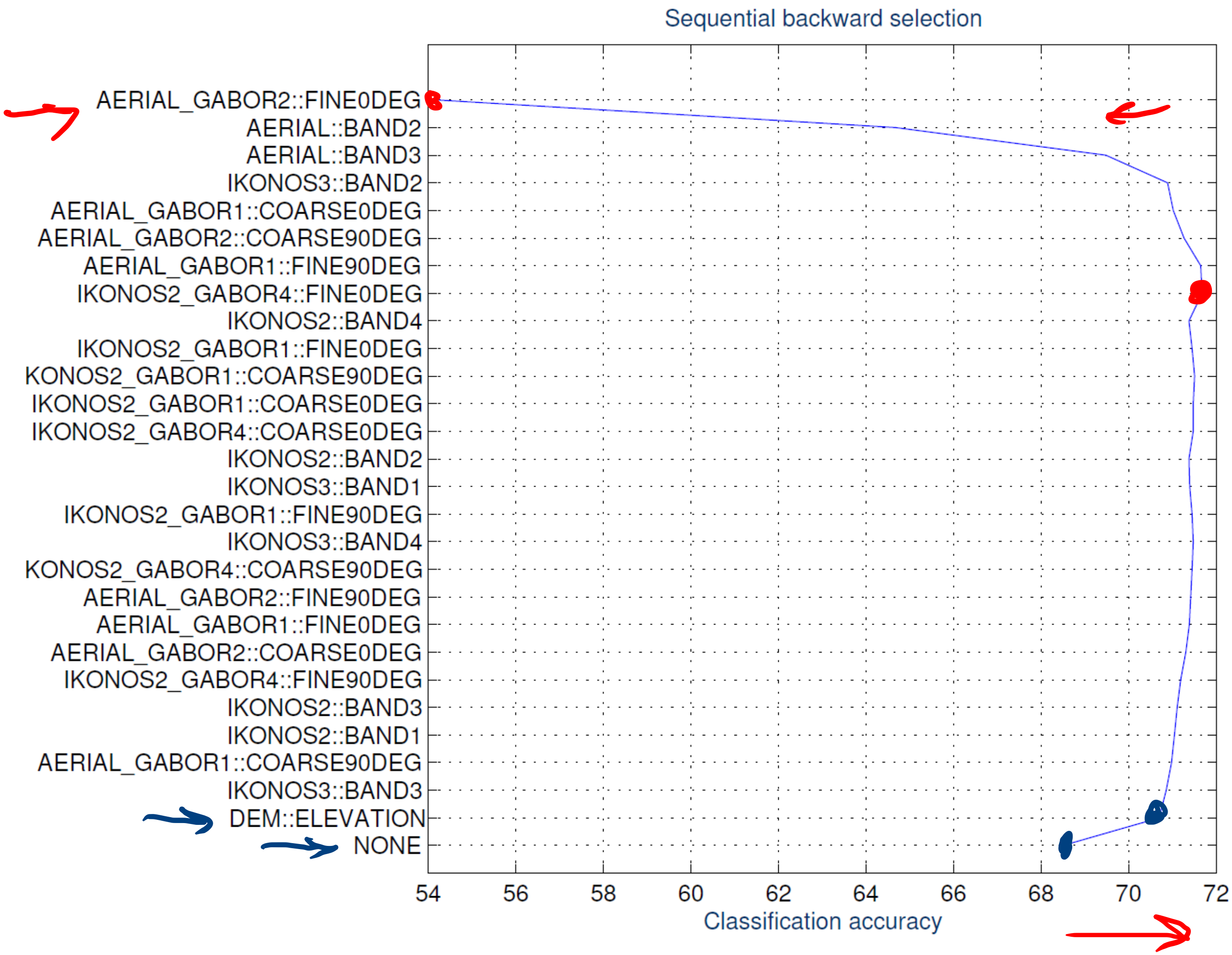
- classification of a satellite image using **28 features**.
- x-axis shows the classification accuracy (%) and y-axis shows the features added at **each iteration**
- (the first iteration is at the **bottom**).
- The **highest** accuracy value is shown with a **star**.

$d \rightarrow k$

$\binom{d}{k}$

$O(d^k)$

Sequential backward selection



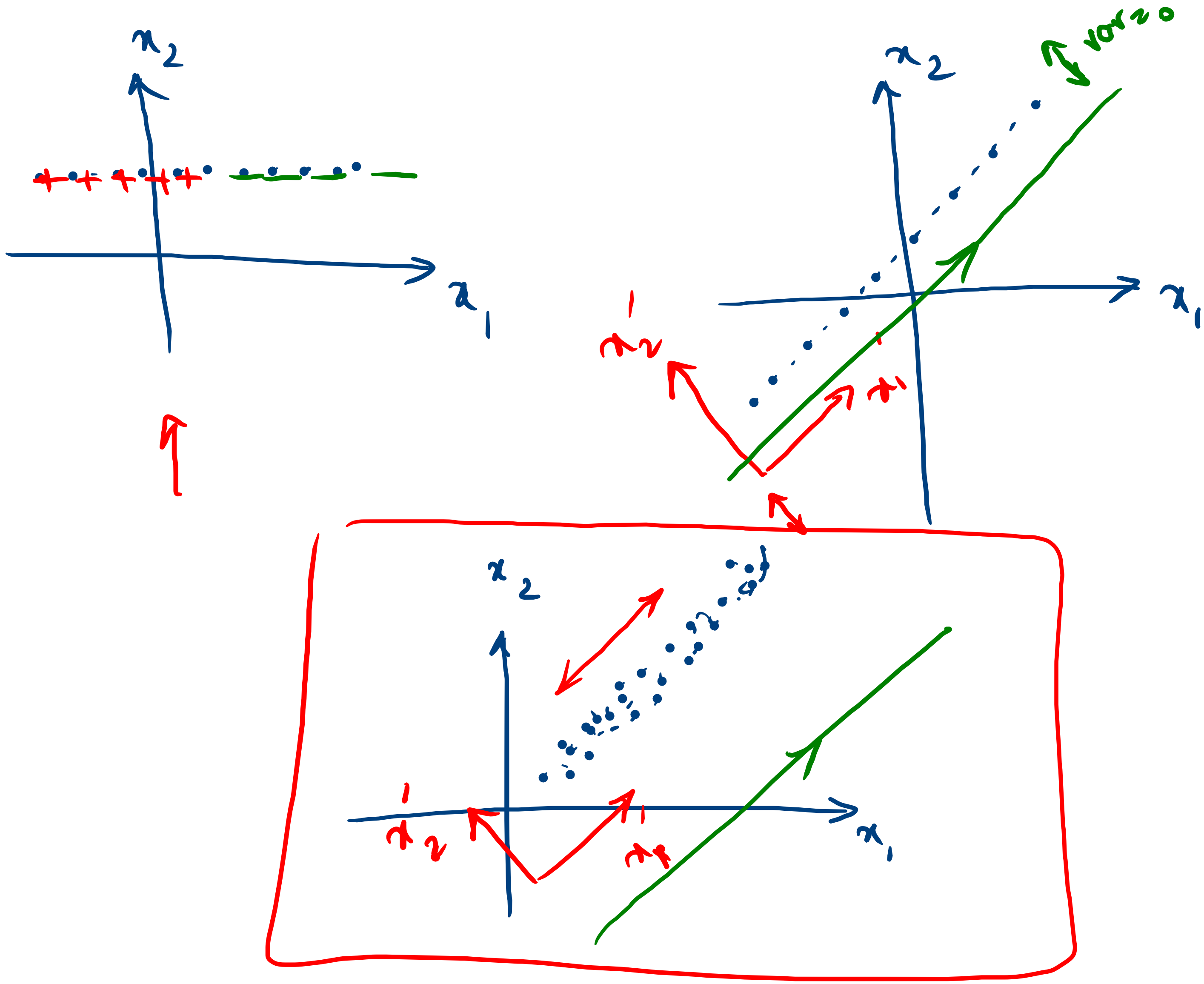
- y-axis shows the features **removed** at each iteration (the first iteration is at the bottom). The highest accuracy value is shown with a **star**.

Bidirectional Search (BDS)

- BDS applies SFS and SBS **simultaneously**:
  - SFS is performed from the empty set.
  - SBS is performed from the full set.
- To guarantee that SFS and SBS converge to the same solution:
  - Features already selected by SFS are **not removed** by SBS.
  - Features already removed by SBS are **not added** by SFS.

Floating techniques

- The main limitation of:
  - **SFS** is that it is unable to remove features that become non useful **after the addition** of other features.
  - **SBS** is its inability to reevaluate the usefulness of a feature after it has been **discarded**.
- **Sequential floating forward selection (SFFS):**
  - Sequential floating forward selection (SFFS) starts from the **empty** set.
  - After each forward step, SFFS performs backward steps as long as the objective function increases.
- **Sequential floating backward selection (SFBS)**
  - Sequential floating backward selection (SFBS) starts from **the full set**.
  - After each backward step, SFBS performs forward steps as long as the objective function increases

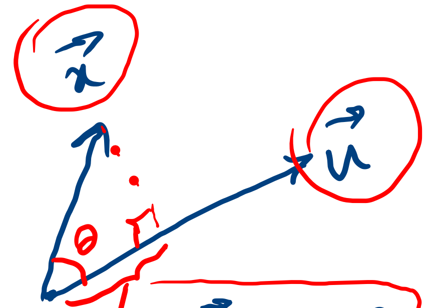


x_2
x_1
x_2
x_1
x'_2
x'_1
vorzo
x_2
x_1
x'_2
x'_1

Principal Component Analysis (PCA)

$$x \in \mathbb{R}^d \longrightarrow \boxed{z \in \mathbb{R}}$$

$\vec{u}$

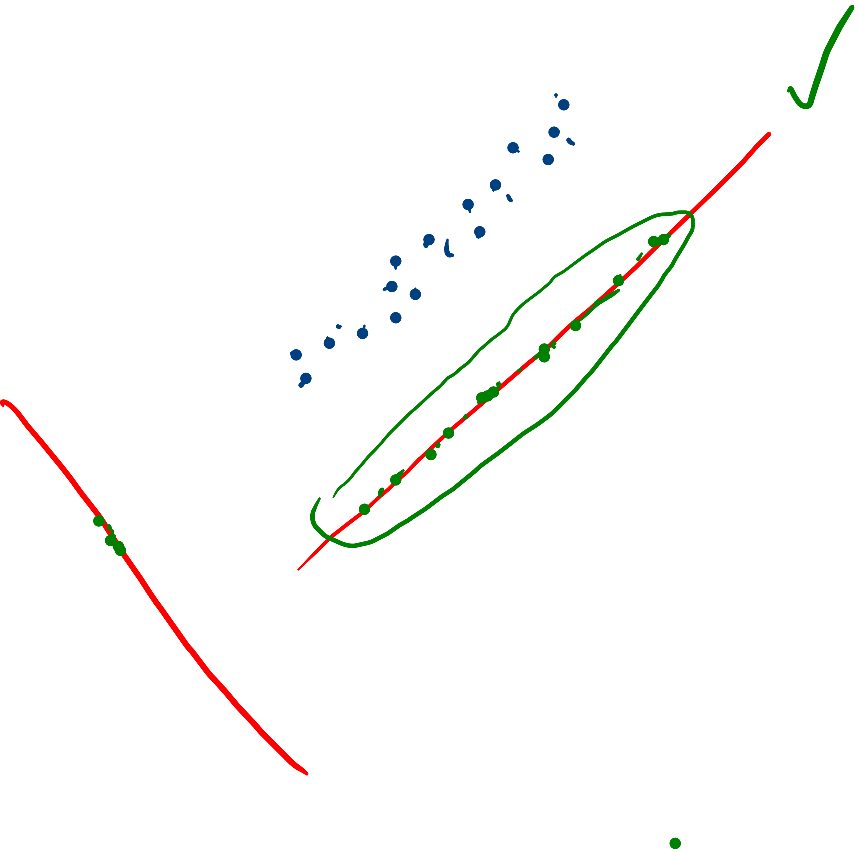


$\|u\|_2 = 1$

$$z = x^T u = \boxed{\|x\| \, \|u\| \cos\theta}$$

$$\boxed{\|\vec{x}\|_2 \cos\theta} = \langle x, u \rangle = x^T u = u^T x$$

$$\boxed{\max_u \ \mathrm{var}(z) \quad \text{s.t.} \quad \|u\|_2^2 = 1}$$



$$\max_{u} \ \mathrm{var}(z) \quad \text{s.t.} \quad \|u\|_2^2 = 1$$

$$\boxed{E[x] = 0} \ \checkmark$$

$$z = u^T x \Rightarrow E[z] = E[u^T x] = u^T \underbrace{E[x]}_{0} = 0$$

$$\mathrm{var}(z) = E[z^2] - \underbrace{E^2(z)}_{0} \simeq \frac{1}{n}\sum_{i=1}^{n} z_i^2$$

$$\max_{u} \ \frac{1}{n}\sum_{i=1}^{n} z_i^2 = \frac{1}{n}\sum_{i=1}^{n} \underbrace{(x_i^T u)^2}_{(u^T x_i)(x_i^T u)} = \frac{1}{n}\sum_{i=1}^{n} u^T x_i x_i^T u$$

s.t. $\|u\|_2^2 = 1$

$$= u^T \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} x_i x_i^T\right)}_{S:\ \text{Sample cov. Matrix}} u$$

$$= u^T S u$$

$$\boxed{\max_{u} \; u^T S u \quad \text{s.t.} \quad \|u\|_2^2 = 1}$$

$u^T u$

$$\longrightarrow \underbrace{L(u) = u^T S u - \lambda\left(\|u\|_2^2 - 1\right)}$$

$x^T A x$

$$\nabla_u L = 0 \implies 2Su - \lambda 2u = 0$$

$$\implies \boxed{Su = \lambda u}$$

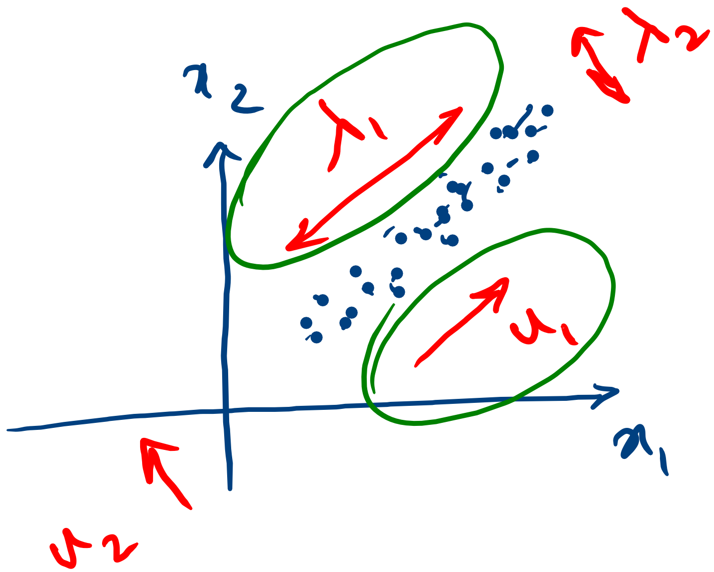
eigenvector

eigenvalue

$x \in \mathbb{R}^d$

$S_{d \times d}$ → d, d

$$\max \underbrace{u^T S u} = \max u^T \lambda u = \max \lambda \underbrace{u^T u}_{1} = \boxed{\max \lambda}$$



x_2
x_1
λ_1
λ_2
v_1
v_2