Unsupervised Learning

- recap: supervised learning : - we possess a training set of features and desired response

  $TS = \{ (X_i, Y_i) \}_{i=1}^{N}$ => learn the mapping $Y = f(X)$

  - basic properties of the mapping ("structure") ~~of~~ is already given in TS, i.e. we know that we are looking for a regression plane (least squares regression) or a decision boundary (classification)

- unsupervised learning: - no pre-defined problem structure, TS without response

  $TS \{ X_i \}_{i=1}^{N}$

  - why? * labeled data (with Y's) is very expensive, unlabelled data is cheap

    => what can we do without labels?

    - in the initial state of a research project, we may not even know what to look for $\hat{=}$ "data mining"

  - ultimately, unsupervised learning is equivalent to artificial general intelligence (AGI)

    => "AI scientist": can define research goals, data collection procedures/experiments, models, optimazation/learning alg. all on it's own

  - for now: search for certain types of useful "default" structures in the data

    $\hat{=}$ pattern discovery, where types of patterns are implicitly defined by the model & learning alg.

## Unsupervised Learning

· recap: supervised learning : - we possess a training set of features and desired response

$$TS = \{ (X_i, Y_i) \}_{i=1}^{N} \Rightarrow \text{learn the mapping } Y = f(X)$$

- basic properties of the mapping ("structure") ~~of~~ is already given in TS, i.e. we know that we are looking for a regression plane (least squares regression) or a decision boundary (classification)

· unsupervised learning: - no pre-defined problem structure, TS without response

$$TS \; \{ X_i \}_{i=1}^{N}$$

- why? * labeled data (with Y's) is very expensive, unlabelled data is cheap
  => what can we do without labels?
  · in the initial state of a research project, we may not even know what to look for ≙ "data mining"
- ultimately, unsupervised learning is equivalent to artificial general intelligence (AGI)
  => "AI scientist" : can define research goals, data collection procedures/experiments, models, optimazation/learning alg. all on it's own
- for now: search for certain types of useful "default" structures in the data ≙ pattern discovery, where types of patterns are implicitly defined by the model & learning alg.

## Principal Component Analysis (PCA)

- find linear variable transform, such that the dimension is reduced without loosing information: $z = \varphi(x) = X \cdot V^T$ ~~$\in \mathbb{R}^{D'}$ $x \in$~~

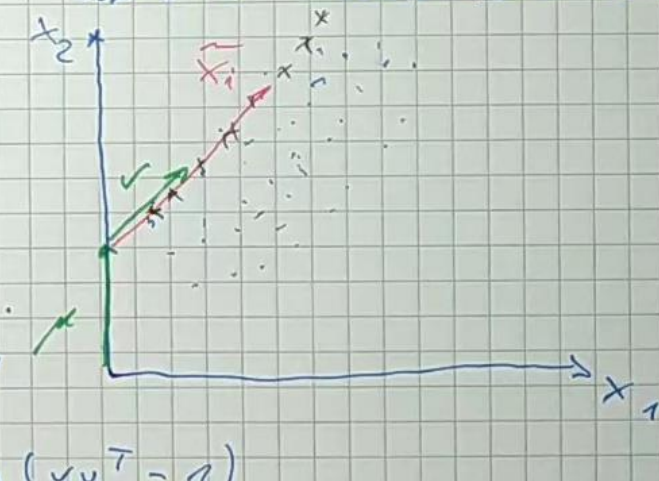
  $x_i \in \mathbb{R}^D$, $z_i \in \mathbb{R}^{D'}$ $\quad D' < D$ (ideally $D' \ll D$)

  all the essential properties of $x$ are still in $z$, but the "noise" is gone
- what's the optimal projection matrix $V$?
- extreme case: $D' = 1$, $z_i$ is a single scalar feature

  embed new feature into original feature space: define a 1-dimensional line in $\mathbb{R}^D$

row vectors

$$\tilde{x}_i = \underset{\text{mean}}{\mu} + \underset{\text{direction}}{v} \cdot z_i$$



the $\tilde{x}_i$ should be approximations of $x_i$

$$\Rightarrow \hat{\mu}, \hat{v}, \{\hat{z}_i\} = \arg\min_{\mu, v, \{z_i\}} \sum_i (x_i - \tilde{x}_i)^2 \quad \text{s.t. } v v^T = 1$$

$$= \arg\min_{\mu, v, \{z_i\}, \lambda} \sum_i (x_i - \mu - v z_i)^2 + \lambda (v v^T - 1)$$

$$\frac{\partial \text{Loss}}{\partial \mu} = -2 \sum_i (x_i - \mu - v z_i) \overset{!}{=} 0 \Rightarrow \mu = \underbrace{\frac{1}{N} \sum_i x_i}_{\bar{x}} - v \cdot \underbrace{\frac{1}{N} \sum_i z_i}_{\bar{z}}$$

$\Rightarrow \mu$ is not unique, can choose $\bar{z}$ freely $\Rightarrow \bar{z} = 0$ (new feature is centered)
if the original are also centered, $\bar{x} = 0 \Rightarrow \mu = 0$

=> assume data is centered, $\bar{x} = 0$, set $\mu = 0$ => $\bar{z} = 0$ is also centered

simplified opt. problem

$$\hat{v}, \{\hat{z}_i\} = \arg\min_{v, \{z_i\}, \lambda} \sum_i (x_i - v z_i)^2 + \lambda (v v^T - 1)$$

$$\frac{\partial loss}{\partial z_i} = -2 \cancel{\sum_i} (x_i - v z_i) \cdot v^T \overset{!}{=} 0 \qquad x_i v^T = \underbrace{v v^T}_{=1} z_i \qquad z_i = \frac{x_i v^T}{v v^T}$$

$$\boxed{z_i = x_i v^T}$$

simplify again:

$$\hat{v}, \hat{\lambda} = \arg\min_{v, \lambda} \sum_i (x_i - v \cdot x_i v^T)^2 + \lambda (v v^T - 1)$$

expand the square:

$$
\begin{aligned}
(x_i - (x_i v^T) \cdot v)^2 &= (x_i - (x_i v^T) v)(x_i - (x_i v^T) v)^T \\
&= x_i x_i^T - (x_i v^T)(v x_i^T) - (x_i v^T)(x_i v^T) + (x_i v^T) \underbrace{v v^T}_{=1} (x_i v^T) \\
&= x_i x_i^T - (x_i v^T)(v x_i^T) \\
& \underbrace{\cancel{= x_i x_i^T}}_{\text{independent of the solution } v \Rightarrow \text{drops}}
\end{aligned}
$$

$$\hat{v}, \hat{\lambda} = \arg\max_{v, \lambda} \sum_i v x_i^T x_i v^T + \lambda (v v^T - 1)$$

$$\boxed{\hat{v}, \hat{\lambda} = \arg\max_{v, \lambda} v S v^T + \lambda (v v^T - 1)} \qquad S = \sum_i x_i^T x_i = X^T X$$

scatter matrix

⇒ assume data is centred, $\bar{x}=0$, set $\mu=0$ ⇒ $\bar{z}=0$ is also centred

simplified opt. problem

$$\hat{v},\{\hat{z}_i\} = \arg\min_{v,\{z_i\},\lambda} \sum_i (x_i - v z_i)^2 + \lambda(vv^T - 1)$$

$$\frac{\partial loss}{\partial z_i} = -2 \cancel{\sum_i} (x_i - v z_i)\cdot v^T \overset{!}{=} 0 \qquad x_i v^T = \underbrace{vv^T}_{=1} z_i \qquad z_i = \frac{x_i v^T}{vv^T}$$

$$\boxed{z_i = x_i v^T}$$

simplify again:

$$\hat{v},\hat{\lambda} = \arg\min_{v,\lambda} \sum_i (x_i - v\cdot x_i v^T)^2 + \lambda(vv^T-1)$$

expand the square:

$$(x_i - (x_i v^T)\cdot v)^2 = (x_i - (x_i v^T)v)(x_i - (x_i v^T)v)^T$$

$$= x_i x_i^T - (x_i v^T)(v x_i^T) - (x_i v^T)(x_i v^T) + (x_i v^T)\underbrace{v v^T}_{=1}(x_i v^T)$$

$$= x_i x_i^T - (x_i v^T)(v x_i^T)$$

$$\underbrace{\cancel{= x_i x_i^T}}$$

independent of the solution $v$ ⇒ drops

$$\hat{v},\hat{\lambda} = \arg\max_{v,\lambda} \sum_i v x_i^T x_i v^T - \lambda(vv^T-1)$$

$$\boxed{\hat{v},\hat{\lambda} = \arg\max_{v,\lambda} v S v^T + \lambda \underset{(1-vv^T)}{\cancel{(vv^T-1)}}}$$

$S = \sum_i x_i^T x_i = X^T X$

scatter matrix

$$\frac{\partial loss}{\partial v} = 2 S v^T - 2 \lambda v^T \overset{!}{=} 0 \qquad \boxed{S v^T = \lambda v^T}$$

$\Rightarrow$ $v^T$ must be an eigenvector of the scatter matrix, which one?

$$\hat{\lambda} = \arg\max_{\lambda} \; v \underbrace{S v^T}_{\lambda v^T} + \lambda (1 - v v^T)$$

$$= \arg\max_{\lambda} \; \lambda v v^T + \lambda - \lambda v v^T = \arg\max_{\lambda} \; \lambda$$

$\Rightarrow$ $\boxed{\lambda \text{ must be the biggest eigenvalue of } S}$ $\boxed{v \text{ the corresponding eigenvector}}$

can reapeat the same procedure with residuals $x_i' = x_i - \hat{x}_i = x_i - v \cdot z_{i1}$

$\Rightarrow$ get the second-best feature $z_{i2}$, and so on until we have $D'$ features

PCA algorithm

① center $X$ and compute scatter matrix $S = X^T X$

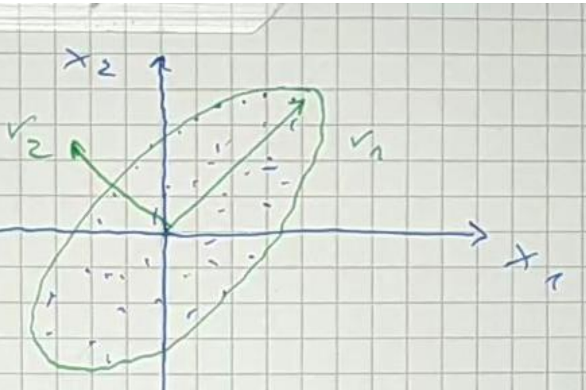
② compute eigen decomposition $S = V \Lambda V^T$ and sort by decreasing eigenvalue: $\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_D \end{pmatrix}$ such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_D \geq 0$

③ choose new feature dimension $D' < D$

④ compute new features: $z_{ij} = x_i v_j^T$ ← eigenvectors for $j = 1 \ldots D'$

intuitive interpretation :

- eigen-decomposition of S approximates the data by an ellipse (see QDA)



- eigenvectors are the ellipse axes ,

=> the most important new feature $z_{i1}$ is along the longest axis of ellipse (because it is largest eigenvalue)

=> PCA selects the coordinate system, where the data vary most

≙ the variance of the new features $\mathrm{Var}_i(z_{ij})$ is maximized

≙ "most informative" means that the values of the features vary a lot

(=> choosing good units for the original features $x_i$ is important, otherwise you can make the data vary arbitrarily => PCA meaningless)

e.g. scale original features such that $\mathrm{Var}_i(x_{ij}) = 1$

- important property of PCA: the new features $z_j$ are pair-wise uncorrelated

Variants of PCA

- keep the linear variable transform, but change the optimality criterion and/or constraints
  - Independent Component Analysis (ICA) : $z_j$ are pair-wise independent } later
  - Non-negative matrix factorization (NMF): $z_j, v \geq 0$ } later
- define non-linear PCA via the kernel trick
  - kernel-PCA