CS/ECE/ME 532
Matrix Methods in Machine Learning

Welcome!

Seeing your assessment

- Today in class/after class
- During in-person office hours
- Tomorrow (Friday, March 8):
  - 9am – 1pm: sign up for a slot to see your assessment in EH 3419
  - *There will be no Zoom office hours tomorrow*
- Make an appointment by email

Activity 12

Hint for 2b: $\boldsymbol{c} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} \longrightarrow ||\boldsymbol{c}||_2^2 = ||\boldsymbol{a}||_2^2 + ||\boldsymbol{b}||_2^2$

SVD for regularizing least squares:

supervised learning

$$\min_{\boldsymbol{w}} ||\boldsymbol{A}\boldsymbol{w} - \boldsymbol{y}||^2 \longrightarrow \boldsymbol{w}^* = \underbrace{(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T}_{\text{pseudo-inverse } \boldsymbol{A}^\dagger}\boldsymbol{y}$$

What if $(\boldsymbol{A}^T\boldsymbol{A})$ is not invertible?

```
np.linalg.inv(X.T@X)
---------------------------------
--------------------------
LinAlgError: Singular matrix
```

Regularize!

$\longrightarrow$ ridge regression: $(\boldsymbol{A}^T\boldsymbol{A} + \lambda\boldsymbol{I})^{-1}\boldsymbol{A}^T\boldsymbol{y}$

$\longrightarrow$ truncate the SVD.

$$\boxed{\boldsymbol{A}^\dagger = \lim_{\lambda \downarrow 0} (\boldsymbol{A}^T\boldsymbol{A} + \lambda\boldsymbol{I})^{-1}\boldsymbol{A}^T}$$

problem: $\frac{1}{\sigma_p}$ huge if columns of A are close to linearly dependent

idea: set $\frac{1}{\sigma_p}, \dots$ to zero, i.e, truncate the SVD.

r is a regularization parameter!

$$\boldsymbol{w}^* = \boldsymbol{V}\Sigma^{-1}\boldsymbol{U}^T\boldsymbol{y} \longrightarrow \boldsymbol{w}^* = \boldsymbol{V}\Sigma_r^{-1}\boldsymbol{U}^T\boldsymbol{y}$$

$$\begin{bmatrix} \frac{1}{\sigma_1} & & & 0 \\ & \frac{1}{\sigma_2} & & \\ & & \ddots & \\ 0 & & & \frac{1}{\sigma_p} \end{bmatrix} \longrightarrow \begin{array}{c} r \times r \\ \left[\begin{array}{cc} \boxed{\begin{matrix} \frac{1}{\sigma_1} & & \\ & \frac{1}{\sigma_2} & \\ & & \ddots \end{matrix}} & 0 \\ 0 & 0 \end{array}\right] \end{array}$$

Set other entries to zero

pseudo-inverse in Python

numpy.linalg.pinv

numpy.linalg.pinv(*a, rcond=1e-15, hern*

Compute the (Moore-Penrose) pseu

Principal Component Analysis (PCA):

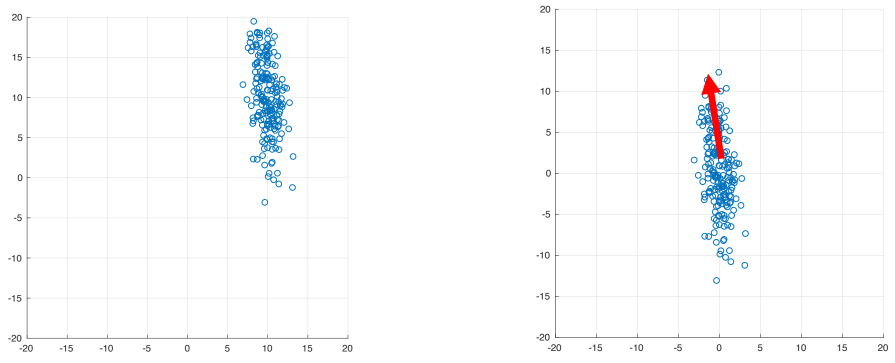
(Unsupervised Learning)

unsupervised learning
no y vector/ label
want to find pattern

input: data $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots \in \mathbb{R}^2$

step 1: center data by removing mean
step 2: stack data as columns of matrix $\boldsymbol{X} \in \mathbb{R}^{2\times n}$
step 3: compute SVD of $\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^T$

1st principal component is first column of $\boldsymbol{U}$



PCA can be used to fit a line (or subspace) to data.

- PCA minimizes perpendicular distance to line
- Regression minimizes vertical distance (along the “label” dimension)