

An aerial photograph of a city, likely Madison, Wisconsin, taken from a high vantage point. The city is situated on a peninsula, with a large body of water (Monona Lake) in the foreground and background. The sun is setting behind a hill in the distance, creating a warm, golden glow over the entire scene. Several sailboats are visible on the water. A large, semi-transparent rectangular box is overlaid on the center of the image, containing the course title in black text.

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: $c = \begin{bmatrix} a \\ b \end{bmatrix} \rightarrow \|c\|_2^2 = \|a\|_2^2 + \|b\|_2^2$



SVD for regularizing least squares: supervised learning

$$\min_w \|Aw - y\|^2 \rightarrow w^* = \underbrace{(A^T A)^{-1} A^T y}_{\text{pseudo-inverse } A^\dagger}$$

What if $(A^T A)$ is not invertible?

Regularize!

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

→ truncate the SVD.

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.

$$w^* = V \Sigma^{-1} U^T y \rightarrow w^* = V \Sigma_r^{-1} U^T y \quad r \text{ is a regularization parameter!}$$

$$\begin{bmatrix} \frac{1}{\sigma_1} & & & 0 \\ & \frac{1}{\sigma_2} & & \\ & & \ddots & \\ 0 & & & \frac{1}{\sigma_p} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{1}{\sigma_1} & & & 0 \\ & \frac{1}{\sigma_2} & & \\ & & \ddots & \\ 0 & & & 0 \end{bmatrix}_{r \times r}$$

pseudo-inverse in Python

`numpy.linalg.pinv`

`numpy.linalg.pinv(a, rcond=1e-15, hermitian=False)`
Compute the (Moore-Penrose) pseudoinverse

Set other entries to zero

Principal Component Analysis (PCA): (Unsupervised Learning)

unsupervised learning

no y vector/ label

want to find pattern

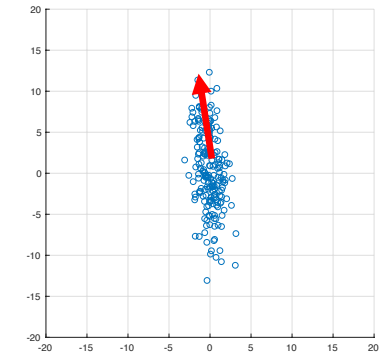
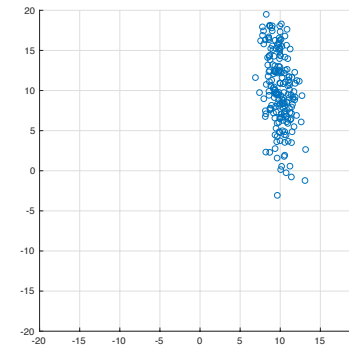
input: data $x_1, x_2, \dots \in \mathbb{R}^2$

step 1: center data by removing mean

step 2: stack data as columns of matrix $X \in \mathbb{R}^{2 \times n}$

step 3: compute SVD of $X = U \Sigma V^T$

1st principal component is first column of 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)

