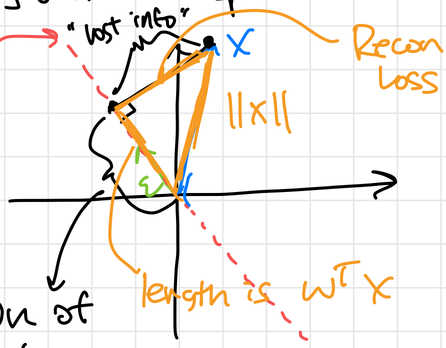
"Reconstruction Error": How well can we reconstruct $\{x^{(1)}, \dots, x^{(n)}\}$ based on only the projections of $\{x^{(1)}, \dots, x^{(n)}\}$ onto subspace defined by $w$

minimize

$$\sum_{i=1}^{n} \| x^{(i)} - (w^T x^{(i)}) w \|^2$$

goal: make this "lost info" as little as possible




Projection of $x$ onto subspace spanned by $w$

$$= \underbrace{(w^T x)}_{\text{How far in direction of } w \text{ is } x\text{?}} \cdot \underbrace{w}_{\text{direction}}$$

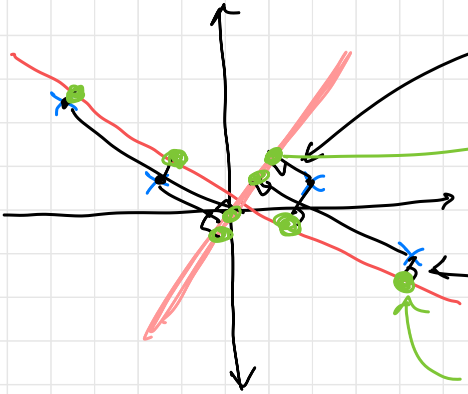
Pythagorean theorem: $\underbrace{(w^T x)^2}_{\text{equivalent to maximize this!}} + \underbrace{(\text{Recon loss})^2}_{\text{minimize this}} = \underbrace{\|x\|^2}_{\text{fixed}}$

Equivalent goal: maximize $\displaystyle\sum_{i=1}^{n} (w^T x^{(i)})^2$

Note: $\displaystyle\frac{1}{n}\sum_{i=1}^{n} \left(w^T x^{(i)} - \cancel{\mathbb{E}[w^T x^{(i)}]}_{=0}\right)^2$

is variance of $w^T x^{(i)}$

Same as maximizing variance of $w^T x^{(i)}$'s




Find $w$ to maximize $\frac{1}{n}\sum_{i=1}^{n}\left(w^T x^{(i)}\right)^2$

$$= \frac{1}{n}\sum_{i=1}^{n} \underbrace{(w^T x^{(i)})}_{1\times 1} \underbrace{(x^{(i)T} w)}_{1\times 1}$$

by associativity

$$= \frac{1}{n}\sum_{i=1}^{n} \underbrace{w^T}_{1\times d} \underbrace{(x^{(i)} x^{(i)T})}_{d\times d} \underbrace{w}_{d\times 1}$$

$$= w^T \left( \underbrace{\frac{1}{n}\sum_{i=1}^{n} x^{(i)} x^{(i)T}}_{} \right) w$$

in general, covariance matrix

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n} (x^{(i)} - \cancel{\mu}^{\,0})(x^{(i)} - \cancel{\mu}^{\,0})^T$$

$$= w^T \underbrace{\Sigma}_{} w$$

symmetric (because covariance between A & B = covariance between B & A)

Every symmetric matrix $\Sigma$ can be written as

$$\Sigma = UDU^T \quad \text{where} \quad D = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}$$

is diagonal

eigenvalues

and $U$ is orthonormal matrix

$$\begin{array}{cccc} \begin{bmatrix} \, & \, & \cdots & \, \end{bmatrix} \\ u_1 \; u_2 & & & u_d \end{array}$$

eigenvectors

Each column $u_i$ has $\|u_i\| = 1$

and $u_i^T u_j = 0$ for all $i \neq j$

(i.e. orthogonal)

Now: maximize $w^T \Sigma w \iff \max \underbrace{w^T U}_{a^T} D \underbrace{U^T w}_{a}$

define $a = U^T w$.

$\|a\| = 1$ because $\|w\| = 1$ and $U$ is orthonormal

$$\Rightarrow \underset{a}{\text{maximize}} \quad a^T D a$$

$$= \sum_{j=1}^{d} \lambda_j a_j^2 \qquad \text{subject to} \quad \sum_{j=1}^{d} a_j^2 = 1$$

Optimal solution: Choose $a_j = 1$ for $\lambda_j$ that is the largest eigenvalue

$a_j = 0$ else

Alternatively: Sort all the eigenvalues so that

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$$

then best choice of $a = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

Find $w$: Solve for

$$a = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = U^T w$$

$$\begin{array}{c} u_1 \\ u_2 \\ \vdots \\ u_d \end{array} \begin{bmatrix} \longrightarrow \\ \longrightarrow \\ \vdots \\ \longrightarrow \end{bmatrix} \begin{bmatrix} w \end{bmatrix} = \begin{bmatrix} u_1^T w \\ u_2^T w \\ \vdots \\ u_d^T w \end{bmatrix}$$

Solution is:

$w = u_1$

Overall PCA algorithm:

Given $\{x^{(1)}, \dots, x^{(n)}\}$

1. Mean-center data
2. Compute $\Sigma = \frac{1}{n} \sum_{i=1}^{n} x^{(i)} x^{(i)T}$
3. Decompose $\Sigma$ into $UDU^T$
4. Choose $w$ to be eigenvector corresponding to largest eigenvalue

What if we want $>1$ dimension?
e.g. 2-D plots

Solution: If you want $K$ dimensions, use the eigenvectors corresponding to largest $K$ eigenvalues

## Supervised Learning

$\text{Data} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$

## Unsupervised Learning

$\text{Data} = \{x^{(i)}\}_{i=1}^{n}$

Data was handed to us

Learning algorithm did not influence what data was collected
"passive"

## Reinforcement Learning

Learning algorithm is an agent that can take actions

(1) Influence what data you observe — Bandits ✓

(2) Influence state of the world or agent — Bandits ✗

---

## Bandit Problems:

- Has actions & observations, but no state

**Casino** "one-armed bandit" = slot machine

Action: Choose 1 slot machine to try

Observation: $$ win or lose

**Medicine** K different candidate medicines w/ unknown success rates

Action: Prescribe 1 of K medicines to each new patient

Observation: Did they get better?

**TikTok** A bunch of videos to recommend, new user joins with unknown preferences

Action: Recommend a video

Observation: Did user watch / like / etc ...

## Formalization of Bandits:

- Set of actions $\{1, \dots, K\}$
- Each action has $\boxed{\text{reward}}$ distribution $P_a(r)$
  - ↳ unknown to learner
  - ↳ part of the environment

What thing do we want to maximize?
- Casino: $$
- Medicine: Patient welfare

Agent plays $T$ rounds of a game
At each time $t = 1, \dots, T$:
- Player $\boxed{\text{chooses}}$ action $A_t \in \{1, \dots, K\}$
- Player receives reward $R_t \sim P_{A_t}(r)$

learning algorithm chooses based on previous rewards $R_1, \dots, R_{t-1}$

Goal: Maximize total reward $\sum_{t=1}^{T} R_t$

To evaluate a bandit algorithm, measure its $\boxed{\text{regret}}$
= How well you did relative to best possible strategy

Define $\mu(a) = \mathbb{E}_{R \sim P_a(R)}[R]$ (expected reward when choosing action $a$)

Optimal action $a^* = \underset{a \in \{1, \dots, K\}}{\operatorname{argmax}} \; \mu(a)$

$$\text{Regret} = \underbrace{N(a^*)\cdot T}_{\text{expected reward of optimal strategy}} - \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} R_t\right]}_{\text{Expected reward when using our learning algorithm}}$$

Ideally, want small regret

Any good algorithm will still have non-zero regret b/c has to try all the actions first