SVD Proofs:

- Let's find this V, Σ_1, U assuming they always exist.

  Assuming $AV = U\Sigma_1$, let's calculate what U, Σ_1, V are.

- Prove that V is the eigenbasis of A^T (row space of A)

$$A = U\Sigma_1 V^T$$
$$A^T = V\Sigma_1^T U^T = V\Sigma_1 U^T$$
$$\therefore \quad A^TA = (V\Sigma_1 U^T)(U\Sigma_1 V^T)$$
$$= V\Sigma_1 U^T U\Sigma_1 V^T$$
$$= V\Sigma_1^2 V^T = V\Sigma_1^2 V^{-1}$$
$$\text{or} \quad A^TA.V = V\Sigma_1^2$$

$\therefore$ V is the eigenvector matrix of AA^T

and $[\sigma_1, \sigma_2 \dots]^T$ are the $\sqrt{\lambda_1}\ \sqrt{\lambda_2} \dots$ of matrix AA^T.

$A_{m\times n} \Rightarrow A^TA \in n\times n \quad \therefore \quad V \equiv n\times n$

- Prove that U is the eigenbasis of A (col. space of A).

Now, how to find U?

$$AA^T = (U\Sigma_1 V^T)(V\Sigma_1 U^T) = U\Sigma_1 V^T V\Sigma_1 U^T$$
$$= U\Sigma_1^2 U^T = U\Sigma_1^2 U^{-1}$$
$$\Rightarrow \quad AA^T U = U\Sigma_1^2.$$

↳ Eigenvector of AA^T.

$U = m\times m$

- Prove that U and V are both orthogonal.

Prove that matrix A always has the SVD decomposition

$A^T A . V = \lambda . V$ → always true, $\lambda \geq 0$ and V is $\perp$ since $A^T A$ is PSD.

$A^T \left(\frac{AV}{\sqrt{\lambda}} \right) = \frac{\lambda}{\sqrt{\lambda}} . V$

Now $AA^T \left(\frac{AV}{\sqrt{\lambda}} \right) = A \left(\frac{\lambda}{\sqrt{\lambda}} V \right) = \lambda \left(\frac{AV}{\sqrt{\lambda}} \right)$ → This is the eigenvector eqn for AA^T.

∴ The matrix $\left(\frac{AV}{\sqrt{\lambda}} \right)$ must be orthonormal, since AA^T is PSD.

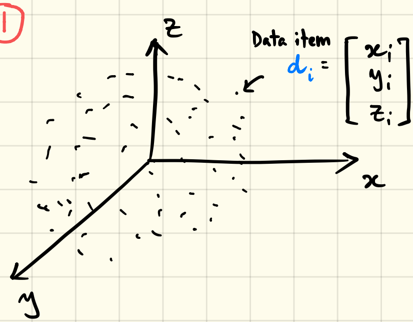
Let $U = \frac{AV}{\sqrt{\lambda}}$ where U is orthonormal. ∴ $AV = U\sqrt{\lambda}$

$$\therefore \quad A = U\sqrt{\lambda}V^{-1} = U \Sigma V^T \quad //$$

PRINCIPAL COMPONENT ANALYSIS (PCA)

PCA : Principal Component Analysis

①



② Prerequisite: Matrix $A \equiv \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \\ z_1 & z_2 & & z_n \end{bmatrix}$ (columns $d_1, d_2, \ldots, d_n$)

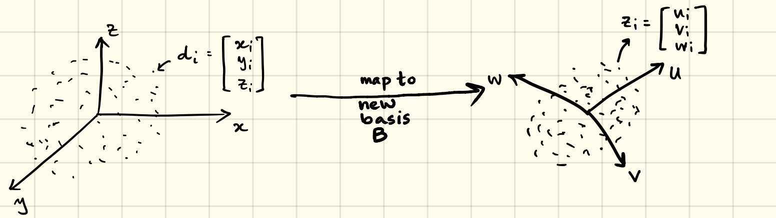
$$\text{Covariance}(A) = AA^T = \begin{bmatrix} x_1 & x_2 & & x_n \\ y_1 & y_2 & \cdots & y_n \\ z_1 & z_2 & & z_n \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & & \\ x_n & y_n & z_n \end{bmatrix}$$

$$= \begin{bmatrix} Var(X) & Cov(XY) & Cov(XZ) \\ Cov(YX) & Var(Y) & Cov(YZ) \\ Cov(ZX) & Cov(ZY) & Var(Z) \end{bmatrix}$$

PCA's Goal: Which basis B will make the data uncorrelated?

Ans: Let's represent data in another orthogonal basis B.

③



Note, when B is orthogonal it can be easily made orthonormal.

④ When data d_i is represented in this new basis B, it becomes, say, z_i.

Note: If B is a Fourier basis, then z_i is the fourier transform.

$$D = \begin{bmatrix} x_1 & x_2 & & x_n \\ y_1 & y_2 & \cdots & y_n \\ z_1 & z_2 & & z_n \end{bmatrix} \equiv \begin{bmatrix} | & | & & | \\ d_1 & d_2 & \cdots & d_n \\ | & | & & | \end{bmatrix}$$

So, $\begin{bmatrix} | & | & | \\ u & v & w \\ | & | & | \end{bmatrix} \begin{bmatrix} u_1 & u_2 & & u_n \\ v_1 & v_2 & \cdots & v_n \\ w_1 & w_2 & & w_n \end{bmatrix} = \begin{bmatrix} | & | & & | \\ d_1 & d_2 & \cdots & d_n \\ | & | & & | \end{bmatrix}$

$\therefore \quad B \cdot Z = D$

Now, to be uncorrelated, covariance of data (in new basis) should be a diagonal matrix (because uncorrelated means $cov(x,y) = 0$)

⑤ Now, data covariance (in new basis) $= ZZ^T$

$$ZZ^T = \begin{bmatrix} u_1 & u_2 & & u_n \\ v_1 & v_2 & \cdots & v_n \\ w_1 & w_2 & & w_n \end{bmatrix} \begin{bmatrix} u_1 & v_1 & w_1 \\ u_2 & v_2 & w_2 \\ \vdots & & \\ u_n & v_n & w_n \end{bmatrix} = \Lambda$$

$$(B^{-1}D)(B^{-1}D)^T = \Lambda$$

$$B^{-1} D . D^T (B^{-1})^T = \Lambda$$

$$D . D^T (B^{-1})^T = B\Lambda$$

$$D . D^T (B^T)^T = B\Lambda \quad \ldots \because B^{-1} = B^T$$

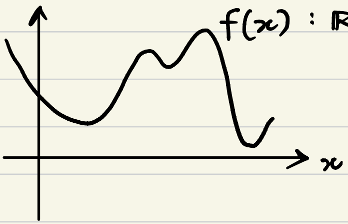
$$DD^T B = B\Lambda$$

$\therefore$ B is eigenvector

Thus, the eigen vectors of the data covariance matrix gives us the desired basis vectors to decorrelate the data.

Now, to compress data D, basically remove the last K columns of B and last K rows of Z, then take the product of the matrices $B'Z' = D'$.
This D' is the compressed matrix.

434 : Optimization Basics

➔ 

$f(x) : \mathbb{R} \rightarrow \mathbb{R}$

$\frac{\partial f(x)}{\partial x} = 0$ gives us local extremum

How do you know maxima or minima ?

$$\left.\frac{\partial^2 f(x)}{\partial x^2}\right|_{x=x^*} > 0 \quad \text{i.e.} \quad \frac{\partial^2 f(x^*)}{\partial x^2} > 0$$

➔ Functions in higher dimensions (i.e. when $\bar{x}$ is vector) ⇒ $f : \mathbb{R}^n \rightarrow \mathbb{R}$

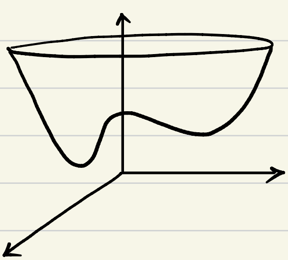
$$\nabla f(x) \equiv \nabla f_x = \begin{bmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \\ \vdots \\ \partial f / \partial x_n \end{bmatrix}$$

∇ → "nabla" or "del"

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & & & \\ \frac{\partial^2 f}{\partial x_n x_1} & \frac{\partial^2 f}{\partial x_n x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

called the "Hessian" matrix

➔ How do we find local maxima / minima of such functions of vectors ?



$\nabla f_x = 0$ ⇒ gives extremums

$\nabla^2 f_{x^*} > 0$ ⇒ indicates minima

(Hessian is a positive definite matrix.)

Matrix A is P.D. when all $\lambda_i(A) > 0$ or $x^T A x > 0, \forall x$

Positive semi definite (PSD) when $\lambda_i(A) \geq 0$, $x^T A x \geq 0$, $\forall x$

→ Note: $\nabla^2 f_{x^*} \geq 0$ is a necessary but not sufficient condition

Example: $f(x) = x^3$

$\nabla f_x = 3x^2 = 0 \Rightarrow x^* = 0$

But is x^* a minima or maxima or neither ?

$\nabla^2 f(x^*) = 6x \big|_{x=0} = 0$

But observe that $x^* = 0$ is neither a minima or maxima.

$x^* = 0$ is NOT maxima or minima.

called " stationary " points

$\nabla^2 f_x > 0$ is sufficient condition

→ $\nabla f_x = 0$ and $\nabla^2 f_x > 0$ gives us local minima.

But how can I get global minima ?

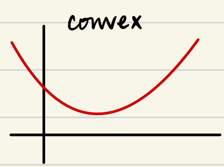
↓

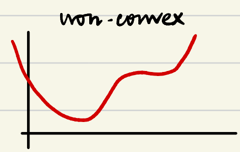
Well, if f_x is a convex fn , then local minima is global minima.

↓

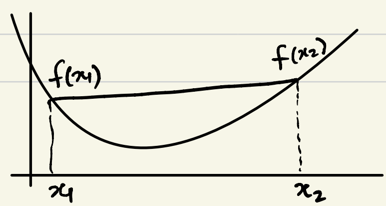
→ What's a convex fn ?

↳ Functions that have an upward curvature everywhere.





Intuitively : The straight line joining any two points $f(x_1)$ and $f(x_2)$ always lies above $f(y)$, where $y \in [x_1, x_2]$



Mathematically : $\alpha f(x_1) + (1-\alpha) f(x_2) \geq f(\alpha x_1 + (1-\alpha) x_2)$, $\alpha \in [0,1]$

↓

How to test for convexity ? $\nabla^2 f_x \geq 0 \iff$ convex fns.

⊛ Summary : Given $f(x)$,
if $\nabla^2 f(x) \geq 0$ (i.e., Positive semi-def Hessian)
then $f(x)$ is convex fn.
Thus $\nabla f(x) = 0$ gives GLOBAL MINIMA.

⊛ But here is the bad news :
↳ Even if $f(x)$ is convex, in many cases, its difficult to solve for $\nabla f(x) = 0$.
Example : $f(x) = e^x + x^2$
↳ Closed form solution difficult

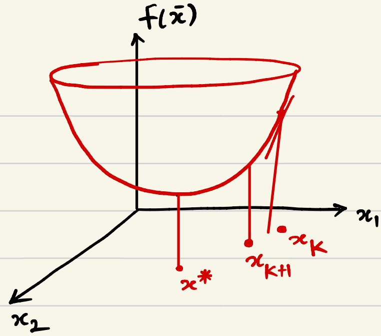
↓

We need to solve such functions iteratively
↳ Motivates gradient descent

⊛ Main idea : We want to start at some $x = x_0$
Move $x_0 \to x_1 \to x_2 \cdots\cdots \to x^*$
s.t. x^* is local/global minima of $f(x)$

This implies : $f(x_{k+1}) < f(x_k)$
So from x_k, we should go along a direction that decreases the value of $f(x_k)$.
↳ Say this direction is $\vec{v}_k$

$\therefore \quad \vec{x}_{K+1} = \vec{x}_K + \vec{v}_K$



(→) What $\vec{v}$ direction will take us most downward?

Answer: The direction of $-\nabla f(x_K)$.

Proof: Taylor's 1st order expansion says

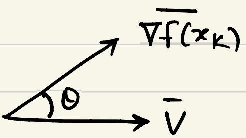
$$f(y) = f(x) + \nabla f(x)^T (y - x) + o(|y - x|)$$

$$\therefore \quad f(x_K + \varepsilon \bar{v}) = f(x_K) + \varepsilon \cdot \nabla f(x_K)^T v_K + o(\varepsilon)$$

$$\underbrace{\lim_{\varepsilon \to 0} \frac{f(x_K + \varepsilon \bar{v}) - f(x_K)}{\varepsilon}} = \nabla f(x_K)^T v_K$$

Rate of change of $f(x)$ along direction v_K

So what is the max and min value of $\nabla f(x_K)^T v_K$?



max when aligned

min when opposite

Recall $\bar{a} \cdot \bar{b} = \|a\| \, \|b\| \cdot \cos\theta$

By Cauchy-Schwarz inequality

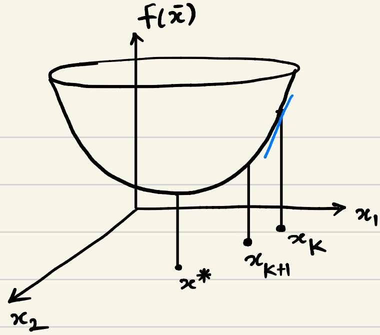
$$-\|\nabla f(x_K)\| \, \|v\| \leq \nabla f(x_K)^T v_K \leq \|\nabla f(x_K)\| \, \|v\|$$

$\therefore$ Maximal downward direction $= -\nabla f(x_K)$ ∎

(→) Thus: $x_{K+1} = x_K + v_K = x_K - \alpha \nabla f(x_K)$

↑ α: step size.

This is called "steepest gradient descent (SGD)"



⊛ Steepest Grad. Descent Algorithm :

1. $K = 0$; α = small positive value ;
   ε = very small value
2. $x[K]$ = random vector
3. Calculate $\nabla f(x[K])$
4. $x[K+1] = x[K] - \alpha \nabla f(x[K])$
5. if $f(x[K+1]) - f(x[K]) < \varepsilon$ then terminate
6. K++
7. Goto ③

Questions :

(a) Why does step size α need to be small ?

(b) Can you draw a case where SGD may not converge if α is not small enough?

(c) Does SGD take the shortest path from x_0 to x^* ?