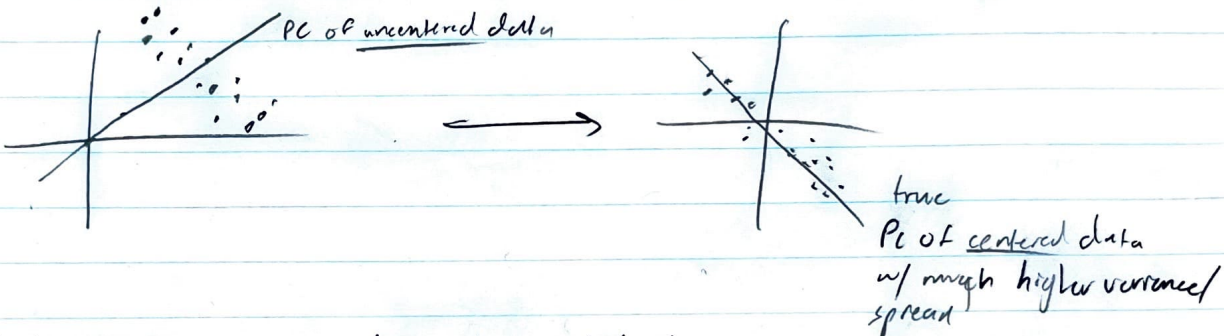
Robert Heeter
ELEC 378
3/10/23

Midterm Exam

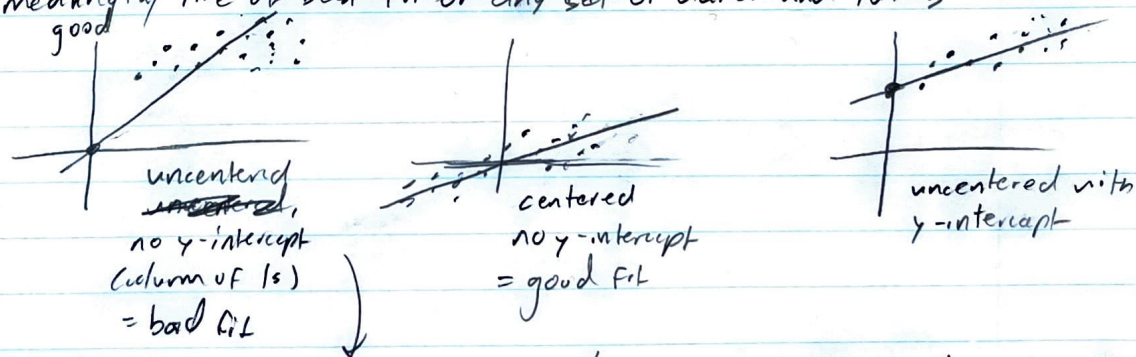
Start time: 12:50 PM CST 3/10/2023
End time: 3:50 PM CST

1a. Sparse vector $\in \mathbb{R}^5$: $\underline{x} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3 \\ 0 \end{bmatrix}$ most elements $= 0$

b. Data centering is important for accurately identifying the real principal components of a data set since the 1D subspace must pass through the origin:



c. Data centering or appending column of 1s to a data matrix to account for a y-intercept is important to produce a meaningful line of best fit for any set of data and labels



without centering or a column of 1s / y-intercept, the trend within the data set cannot be captured as well.

d. Ridge or lasso regression are forms of regularization to account for singular or ill-conditioned data sets where a unique set of coefficients, (w), either cannot be found or is very "unstable" (in the case of very small but non-zero singular values). The penalty terms add a constraint to the optimization/regression to account for this and produce a stable/unique set of coefficients w.

Continued on next page

1). (d). Ridge regression has analytical Equation that can be solved $X^T y = (X^T X + \lambda I) w_{ridge}$ since it uses the 2-norm, while lasso reg. must be solved with a more complex optimization algorithm, which can be more complex/less intuitive

Lasso regression produces a more sparse set of coefficients w, effectively helping with variable selection (selecting the most impactful variables and "ignoring" [0-weight] the others) while ridge does not which may be good for simplifying the number of vars. in the problem

e. "Linear" in lin. reg. refers to the fact that the terms in the model depend linearly on the weighting coefficients w — i.e. in the line of best fit case in 2D:

$$\hat{y} = w_0 + w_1 x$$

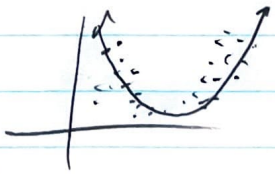
w's are multiplied to x's (linear combination)

The actual model can be nonlinear however (i.e. the dependence of y prediction on the data, x) — for quadratics:

$$\hat{y} = w_0 + w_1 x + w_2 x^2$$

x^2 term is nonlinear

This nonlinearity can be useful for data sets that follow a clearly non-linear relationship (i.e. exponentials or quadratics:



where a line of best fit is not ideal

f. Overfitting refers to the case in linear regression where there are more parameters w_j than data points (x_i, y_i), which causes the model to be fit to noise in the data. This occurs when $n < p$ for the data matrix — i.e. the matrix has more columns than rows and is short & wide, causing some columns of X to be linearly independent vectors

g. Next page

(1). g. If a function is convex for its entire domain, then it must have a single / unique global minimum. ~~that the gradient~~ The gradient in this case always points "downhill" towards the global minimum from any point in the domain of the function.

h. Gradient descent uses the entire set of objective functions each iteration/step (= epoch) ~~is~~; the updated step is $w^{t+1} = w^t - \mu^t \sum_{r=1}^{T} \nabla \mathcal{L}_r(w)$ using all T objective functions $\mathcal{L}_r(w)$ in the step.

Stochastic gradient descent uses a random subset of the objective functions each iteration / step until all objectives have been used (1 epoch); the step now is $w^{t+1} = w^t - \mu^t \sum_{r=1}^{T'} \nabla \mathcal{L}_r(w)$ where T' is a set of randomly selected gradient components that is chosen for the iteration.

SGD is ~~more~~ computationally less expensive and converges faster than standard GD, but is not a smooth descent due to the ~~set~~ arbitrary switching of objective functions, while standard GD is more computationally intensive since it must calculate an average of all of the gradient funcs each iteration, but produces a smoother descent.

2. Next page

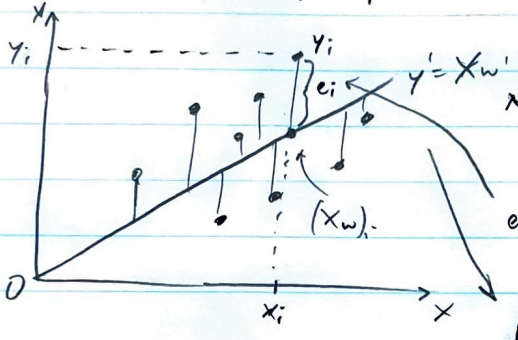
2. a. Linear regression finds the "best" line (through the origin) for the data x_i, y_i for $i = 1 \dots n$ by minimizing the sum of the squared residuals between each data point and the line of best fit:

residuals = "error" = difference between label and prediction

$$\min_{w \in \mathbb{R}^2} \|y - Xw\|^2 \leftarrow \text{squared}$$

$y - Xw = e$

e = error = label (y) − prediction (Xw), where w are the coefficients / weights on the data matrix X.



$e_i = y_i - (Xw')_i$

Minimize $\sum_{i=1}^{n} |e_i|^2$ ← squared residuals / error strength

The normal equations give w^*: ← optimal w

$X^T X w^* = X^T y$

$\therefore w^* = (X^T X)^{-1} X^T y$ (Moore Penrose Pseudoinverse)

where $\hat{y} = Xw^*$

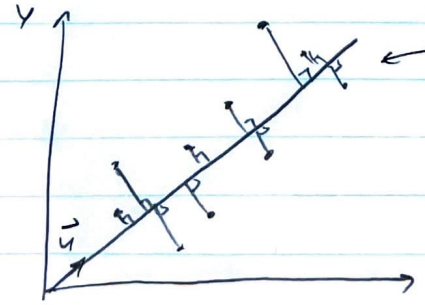
b. $X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$

i.e. "principal component"

Principal components analysis aims to find the optimal 1-dimensional subspace that maximizes the energy / spread / variance of the data after orthogonal projection onto the subspace

$$\max_{\|u\|_2 = 1} \sum_{i=1}^{n} (x_i \cdot u)^2 = \max_{\|u\|_2 = 1} u^T X^T X u$$

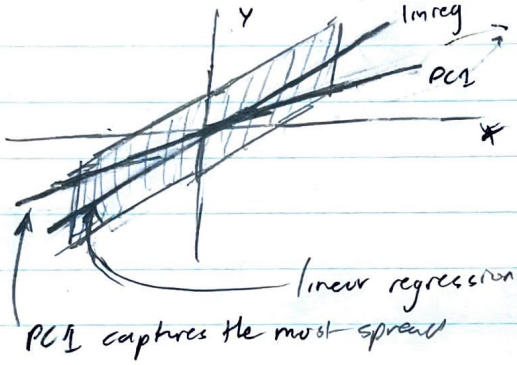
← X centered data matrix; u is direction of 1D subspace / principal component



← orthogonal projection of data on to 1D subspace

Find direction $\vec{u}$ that maximizes spread / variance of projected data

(2). c. The lines found from linear regression and PCA are not the same since they are optimizing different quantities — the sum of the errors squared for linear regression is minimized, while PCA maximizes the variance of the projected data onto a 1D PC/subspace. The line that captures the most spread/variance in the data is not necessarily the same one that minimizes the error in the least squares sense. For example:



→ data set is "parallelogram" in shape

linear regression minimizes error strength

PC1 captures the most spread

d. Linear regression objective function: $\mathcal{L}(w) = \|y - Xw\|^2$

PCA objective function: $\mathcal{E} = \sum_{i=1}^{n} (x_i \cdot u)^2$

Linear regression aims to reduce the error (squared) between the data labels and their "predicted" value by optimizing the prediction curve/line of best fit. The error is defined as the difference between the actual values/labels y and the predicted values Xw.

$$e = y - Xw$$

$$\mathcal{L}(w) = \|e\|^2 \rightarrow \text{minimize objective fxn } \mathcal{L}(w)$$

PCA aims to maximize the objective function $\mathcal{E}$, meaning it aims to maximize the variance/spread of the orthogonally-projected data by optimizing the PC line.

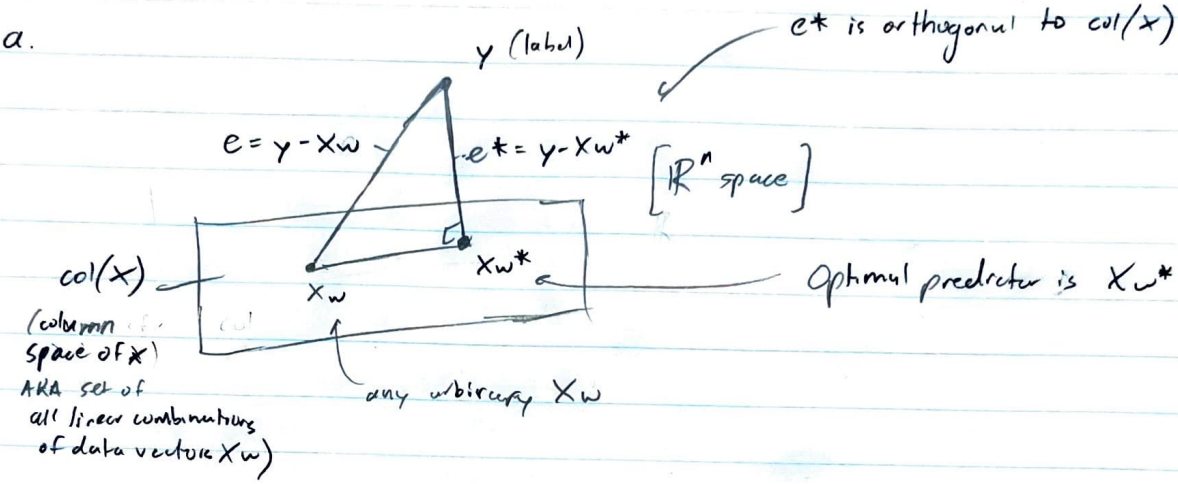
$(x_i \cdot u)^2$ (squared) represents the distance of the projected point x_i on $\vec{u}$ from the origin.

3. Next page

3. $e^* = y - Xw^* \in \mathbb{R}^n$

X: $n \times p$ data matrix; w: arbitrary vector in $\mathbb{R}^p$

a.



b. Orthogonality principle states that optimal error vector $e^* = y - Xw^*$ is orthogonal to each row of X^T (each column of X)

For orthogonality: $x \cdot y = 0 = y^T x \rightarrow e^* \cdot X = 0 = X^T e^*$

$X^T e^* = 0$

$X^T (y - Xw^*) = 0$

$X^T y - X^T X w^* = 0 \rightarrow X^T y = X^T X w^*$

$(X^T X)^{-1} X^T X w^* = (X^T X)^{-1} X^T y$ (I)

$$\boxed{w^* = (X^T X)^{-1} X^T y}$$

where X is data matrix, y are labels

expression for optimal least-squares coefficients → same as expression from objective fxn and gradient in lecture

4. Next page

4. $x_i, y_i \in \mathbb{R} \rightarrow (p = 1)$

Not centered data

Centroid: $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$, ~~$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$~~ $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad w = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$$

← append 1's to center (account for y-intercept)

data matrix

$$y = Xw + e$$

← error

$$\hat{y} = Xw^*$$

← prediction $\hat{y}$

← optimal w parameters

$\hat{y} = Xw^*$

$w^* = (X^T X)^{-1} X^T y$

$$\begin{bmatrix} w_0^* \\ w_1^* \end{bmatrix} = \left(\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

w^* $\quad (X^T X)^{-1}$ $\quad X^T$ $\quad y$

$$X^T X = \begin{bmatrix} (1 \times n) & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix}$$

$$X^T y = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \end{bmatrix}$$

~~Xw~~

$e = y - Xw = 0$?

$\bar{e} = \bar{y} -$

$\bar{e} = \frac{1}{n} \sum_{i=1}^{n} e_i = 0$?

Rearrange

Rearrange?

Rearrange to normal equations from $w^* = (X^T X)^{-1} X^T y$

$X^T X w^* = X^T y$

See next page

(4). Normal equations: $X^T X w^* = X^T y$

Expand:

$$\underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \end{bmatrix}\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}}_{① X^T X}\underbrace{\begin{bmatrix} w_0^* \\ w_1^* \end{bmatrix}}_{w^*} = \underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{② X^T y}$$

$$① X^T X = \begin{bmatrix} (1 \times n) & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix}$$

$$② X^T y = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \end{bmatrix}$$

Now:

$$\underbrace{\begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix}}_{① X^T X}\underbrace{\begin{bmatrix} w_0^* \\ w_1^* \end{bmatrix}}_{w^*} = \underbrace{\begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \end{bmatrix}}_{② X^T y}$$

From first equation:

$$n w_0^* + \left(\sum_{i=1}^{n} x_i\right) w_1^* = \sum_{i=1}^{n} y_i \quad \leftarrow \text{Divide terms by } n$$

$$w_0^* + \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)}_{=\bar{x}} w_1^* = \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} y_i\right)}_{=\bar{y}}$$

$$w_0^* + \bar{x} w_1^* = \bar{y} \longrightarrow \therefore \text{optimal least squares line passes through the data centroid } (\bar{x}, \bar{y})$$

Honor code: On my honor I have neither given nor received any unauthorized aid on this exam.

- [signature]

Robert Heeter

03/10/2023

ELEC 378

Midterm Cheat Sheet

1. Linear algebra, optimization

Unsupervised: PCA, k-means clustering, hier. clust.

Supervised: lin reg, least squares, ridge/lasso reg.

2. Vector space V: α is scalar, $x, y \in V$ $\Rightarrow \alpha x \in V,\ x+y \in V$

Linear combination: $y = \sum_{m=1}^{M} \alpha_m x_m = X a$ (y: vector; X: vector; a: ie. mixing board)

Euclidean 2 norm: $\|x\|_2 = \sqrt{\sum_{i=1}^{p} |x[i]|^2}$ → energy $\|x\|_2^2$

q-norm: $\|x\|_q = \left(\sum_{i=1}^{p} |x[i]|^q\right)^{1/q}$

∞-norm $\|x\|_\infty = \max_i |x[i]|$ ← peak value of x

Normalize vector $\|x\|_2 = 1$ → scale by $\frac{1}{\|x\|_2}$

Inner product: $x \cdot y = \langle x, y \rangle = y^T x = \sum_{i=1}^{p} x[i] y[i]$

Cosine similarity: $\cos(\theta)_{x,y} = \frac{x}{\|x\|_2} \cdot \frac{y}{\|y\|_2}$ $\quad x \cdot x = \|x\|_2^2$

Orthogonality: $x \cdot y = 0 = \langle x, y \rangle$

3. Cauchy-Schwarz inequality: $0 \le |x \cdot y| \le \|x\|_2 \|y\|_2$

x & y are: most diff when orthogonal; or most similar when collinear/lin. dependent

Basis: $\{b_k\}$ vectors are lin. ind. & span V

dimension of V is p

$b_k \cdot b_\ell = 0$ $\quad k \neq \ell$

Orthogonal basis $\{b_k\}_{k=1}$ → elements orthogonal

Orthonormal basis → orthogonal basis & normalized $\|b_k\|_2 = 1$

Inverse of matrix $AA^{-1} = A^{-1}A = I$

↳ B contains orthonormal basis, then $B^{-1} = B^T$ and B is orthogonal matrix

$\underline{x} = \underline{\underline{B}}\,\underline{a} = \sum_{k=1}^{p} a_k \underline{b}_k$ synthesis; $\underline{a} = \underline{\underline{B}}^T \underline{x}$, $a_k = \underline{x} \cdot \underline{b}_k$ analysis

Data matrix $n \times p$ $[X]_{ij}$ $\quad \underline{x}_i$: data point (vector) $\quad \underline{x}_j \in \mathbb{R}^n$ $p \times 1$ column vec.

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{bmatrix}$$ ← vectors are rows

Correlation matrix $X^T X$ → symmetric

$$X^T X = \sum_{k=1}^{n} x_k x_k^T$$

Centering & normalizing: $\mu_i = \frac{1}{n} \sum_{k=1}^{n} x_k[i]$ $\quad \sigma_i = \frac{1}{n} \sum_{k=1}^{n} |x_k[i] - \mu_i|^2$

$x_j[i] \leftarrow \frac{x_j[i] - \mu_i}{\sigma_i}$

3. $Av = \lambda v$ (λ: eigenvalues; v: eigenvectors) $\quad A = V \Lambda V^{-1}$

A must be square

→ real symmetric matrix (correlation/covariance, for example) have real-valued eigenvals & e-vecs

Hermitian transpose A^H

$A^H A = A A^H = I$ → unitary matrix A (extension of orthogonal)

SVD: $A = U \Sigma V^H$ (columns of U, U^H, V, V^H are orthonormal bases)

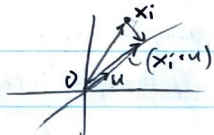
$n \times p$ $=$ $n \times n$ unitary w/ eigenvectors of AA^H $\times$ $n \times p$ diagonal w/ singular values $\times$ $p \times p$ unitary w/ eigenvals of $A^H A$ (non-neg) real #s

Singular value σ: $Av = \sigma u$, $A^H u = \sigma u$

Frobenius norm $\|A\|_F^2 = \sum_{i=1}^{p} \sum_{j=1}^{n} |a_{i,j}|^2$

Rank = # of nonzero singular vals.

4. PCA → Data centered at 0 + normalized



$\|u\|_2 = 1$

$\varepsilon = \sum_{i=1}^{n} (x_i \cdot u)^2$

PCA → $\max_{\|u\|_2 = 1} \sum_{i=1}^{n} (x_i \cdot u)^2 = \|X_c u\|_2^2 = u^T X_c^T X_c u$

maximize energy/spread/variance of data after orthogonal projection onto lower subspace

$X^T X = V \Lambda V^T \rightarrow \max_{\|u\|_2 = 1} u^T V \Lambda V^T u$

covariance matrix; max is first eigenvector v_1 of $X^T X$

energy is eigenvalue → $\lambda_1 = \sum_{i=1}^{n} (x_i \cdot v_1)^2$

2nd PC is orthogonal to first & is 2nd leading eigenvector

PCA via SVD: $X = U \Sigma V^T$ (right singular vectors/columns of V are PCs)

σ_k^2 is energy in PCs

$P = XV = U\Sigma$

5. Objective (loss) fxn $\mathcal{L}$, w_0 is minimizer/max...

Convex fxn: for any x_1, x_2 in f and $0 < t < 1$

$f(tx_1 + (1-t)x_2) \le t f(x_1) + (1-t) f(x_2)$

Convex set: $tx_1 + (1-t)x_2 \in S$

Local min $\|w - w^*\| < \delta$

constraint ↓

Lagrange multiplier λ: $\min_{w \in \mathbb{R}^p, \lambda \in \mathbb{R}} \mathcal{L}(w) + \lambda g(w)$

Gradient: $\nabla \mathcal{L}(w) = \left[\frac{\partial \mathcal{L}}{\partial w_1}, \frac{\partial \mathcal{L}}{\partial w_2}, \ldots \frac{\partial \mathcal{L}}{\partial w_p}\right]^T$

Quadratic

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

$= w^t - \mu^t \sum_{t=1}^{T} \nabla \mathcal{L}_t(w)$ ← step size/learn rule

(5). Gradient descent: $w^{t+1} = w^t - \mu^t \nabla \mathcal{L}(w^t)$

↳ Issues: w sodule and not smooth

Stochastic GD: $\mathcal{L}(w) = \sum_{t=1}^{T} \mathcal{L}_t(w)$ ← sum of smaller obj. functions

↳ computationally **less** expensive; slow convergent & non monotonically ↓

$w^{t+1} = w^t - \mu^t \sum_{t=1}^{T'} \nabla \mathcal{L}_t(w)$

↳ Epoch = all t_n used grads

↖ set of gradients picked each iteration (sequence or random)

6. Cluster → homogeneous groups; unsupervised

K-means: $\min_{C_1 \dots C_K} \sum_{k=1}^{K} W(C_k)$ ← within cluster variation minimize

↳ $W(C_k)$ w/ 2 norm) $= \frac{1}{|C_k|} \sum_{i,i' \in C_k} \|x_i - x_{i'}\|^2$

↳ computation ineffficient

finds local min ← { 1) Random cluster assignment, 2) Centroid $\frac{1}{|C_k|} \sum_{i \in C_k} x_i$

3) Assign to closest centroid

Different "distance" metrics i.e. Euclidean ↗

1) $\| \ \|_2$ 2-norm

2) $\| \ \|_2^2$ squared 2 norm

3) $\| \ \|_1$ 1norm

4) $\| \ \|_\infty$ pnorm (max dist)

5) Cosine distance

Hierarchical clustering → tree-based taxonomy

↳ Agglomerative (bottom-up) ↳ Divisive (top-down)

↳ Similarity function → dendrogram tree

8. x_i → data point/vector y_i → label

$y_i = 1 w_1 + x_i w_2 + e_i$ ← minimize error strength $\sum_{i=1}^{n} |e_i|^2$ (least squares)

$y = Xw + e$

$\min_{w \in \mathbb{R}^2} \underbrace{\|y - Xw\|^2}_{e} \rightarrow \mathcal{L}(w) = \|y - Xw\|^2$ objective fxn

$\nabla \mathcal{L}(w^*) = 0$ @ minimizer w^*

$\nabla \mathcal{L}(w) = \left[\frac{\partial \mathcal{L}(w)}{\partial w_1}, \frac{\partial \mathcal{L}(w)}{\partial w_2} \right]^T$

$\mathcal{L}(w) = \|y - Xw\|^2 = (y - Xw)^T (y - Xw)$

$y^T Xw = w^T X^T y$ ← $= y^T y - y^T Xw - w^T X^T y + w^T X^T Xw$

$= y^T y - 2w^T X^T y + w^T X^T Xw$

$\nabla_z z^T a = a, \quad \nabla_z z^T A z = zA_z$

$\nabla \mathcal{L}(w) = -2X^T y + 2X^T Xw = 0 \rightarrow X^T y = X^T Xw$

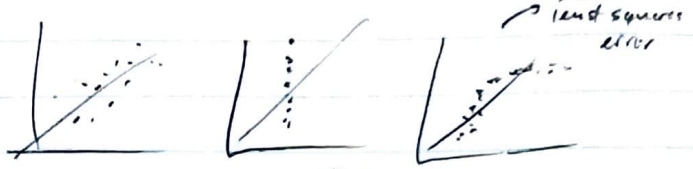
Normal eqns: $X^T y = X^T X w^*$

$w^* = \underbrace{(X^T X)^{-1} X^T}_{= X^+ \text{ Moore Penrose}} y$

$X = U \Sigma V^T, \quad X^+ = V \Sigma^+ U^T$

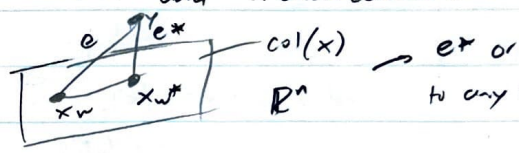
↑ reciprocal of nonzero entries of Σ

8. $w^* = (X^T X)^{-1} X^T y \quad \hat{y} = X w^*$



Orthogonality principle $X^T e^* = 0$

↳ optimal error vector e^* is orthogonal to each row of X^T and ∴ each column of X



→ e^* orthogonal to any Xw

Include column of 1s for y-intercept term w_0

Center data by subtracting mean of y and each column of X

$1^T y = 0, \ 1^T X = 0^T$

→ w_1 is unidentifiable and ∞ w's solve normal eqns

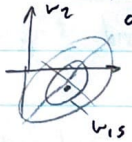
$(X^T X)^{-1}$ does not exist → $X^T X$ is singular or ill conditioned

1) $n \geq p$ (tall, narrow X) but $\text{rank}(X) < p$

↳ 2+ columns of X (and $X^T X$) are lin. dependent

2) $n < p$ (short, wide X) → overfitting; more params w_j than data points (x_i, y_i) model fit to noise

LS Geometry



axes are eigenvectors

eccentricity ↑ as from conditioning #

tradeoff λ

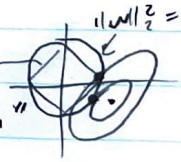
variance ↓ bias ↓

Regularization w/ constraint: $\min_{w \in \mathbb{R}^p, \lambda \in \mathbb{R}} \|y - Xw\|_2^2 + \lambda \|w\|_2^2$

$X^T y = \underbrace{(X^T X + \lambda I)}_{\text{always well-conditioned}} w_{ridge} \rightarrow w_{ridge} = V(\Sigma + \lambda I)^{-1} U^T$

Ridge

Lasso → 1 norm penalty $\|w\|_1 = c$



w_{lasso} is sparse, "variable selection"

Elastic net → combine 2 penalties

Need optimization alg. for lasso