Lecture 7/01 : Features (recap) NO CLASS Friday July 4th

- PCA new feat = linear_combination(original feat)
- Feature selection using L1-reg
- Boosting margins, margin per feature

BINARY $y \in \{\pm 1\}$

Margin(x) = $y \cdot$ classifier score $= y \cdot \underbrace{F(x)}_{\text{score}}$

if $y=+1$: want large score $>0 \iff$ large margin

if $y=-1$: want large neg score $\iff$ large margin <0

---

if $F(x) \simeq 0 \iff$ low margin $\iff$ small confidence

---

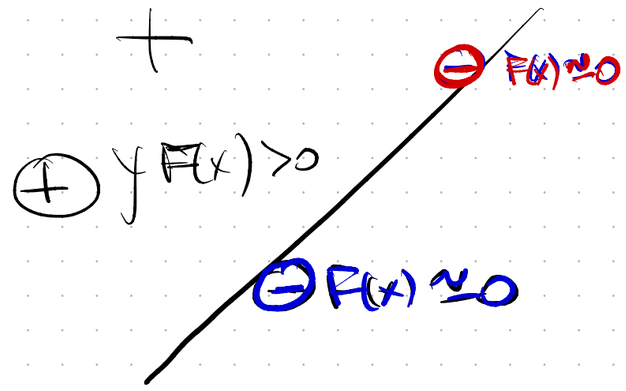
if $yF(x) < 0 \iff$ mistake classification

if margin $\ll 0 \iff$ mistake with high confidence

Quality of classifier $\Leftrightarrow$ large (pos) margins

<u>margin</u> better measurement than <u>accuracy</u>

"how ~~far~~ we predict from 50-50" confidence



correct side.

⊖ yF(x) large pos margin

increasing margin $\Leftrightarrow$ increase confidence

GB, AdaBoost: increase margins yF(x) during training

<u>even</u> for pnts classfied correctly already yF(x)>0.

classif boosting Alg: margin maximized $= e^{yF(x)}$ or $\frac{1}{1+e^{-yF(x)}}$

Margin Analysis

GB : $F(x) = \sum_{t=1}^{T} h_t(x)$ $\quad h_t$ = dec. stump

Adaboost : $F(x) = \sum_{t=1}^{T} \alpha_t \cdot h_t(x)$

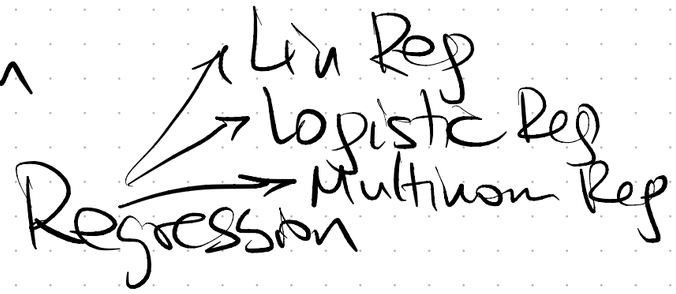
Reorder the summation per feature

$$F(x) = \underbrace{\sum_{f_t=f_1} \alpha_t h_t(x) \cdot y}_{\text{margin}_{f_1}(x)} + \underbrace{\sum_{f_t=f_2} \alpha_t h_t(x) \cdot y}_{\text{margin}_{f_2}(x)} + \underbrace{\sum_{f_t=f_3} \alpha_t h_t(x) \cdot y}_{\text{margin}_{f_3}(x)} + \dots + \underbrace{\sum_{f_t=f_D} \alpha_t h_t(x) \cdot y}_{\text{margin}_{f_D}(x)}$$

$\Rightarrow$ contribution of each feature to the classifier

Note online, optional PB HW4.

HW4 Feat selection using L1-reg Regression



- use very large L1 penalty train
  ⇒ many Feat. end up with 0 or close-to-0 coef.
  L1 penalty ⇒ sparsity. (acc not that good, irrelevant)
- Select $k = \begin{matrix}100\\200\end{matrix}$ features by |coef| largest
- Train (real) classifier without L1 penalty.
  ex L2-reg Regression
  ex. NNet
  ex. NaiveBayes, GDA,
  <u>not good</u>: DecTree, because D.T. have "natural" sparsity be #splits

PCA recap ① rotates data to align dim with axis

Σ = sigma = covar — dim are sorted by var[dim]

$\sum$ = summation — each dim = projection = linear comb (original feat)

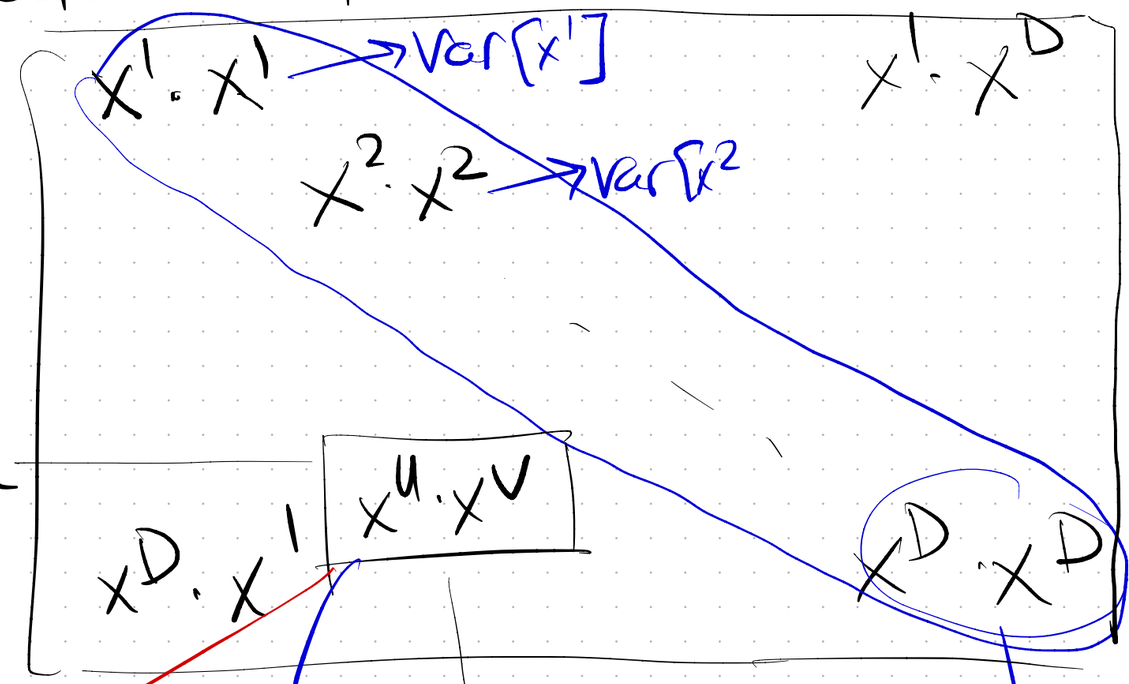
② selects first R dim ⇒ loosing var[] information

$$\mu = \sum_{i=1}^{N} x_i \quad 1 \times D$$

data centered $X = X - \mu$. Each column = feat has 0-mean.

$$\sum_{D \times D} = \text{covar}(X) = \frac{1}{N} \underset{D \times N}{X^T} \underset{N \times D}{X}$$

=



orthonormal vectors

$e_1, e_2 \ldots e_D$

- $\|e_i\| = 1$
- $\|e_i \cdot e_j\| = 0$ (perpendicular)

$\mu \neq 0$ then:

$$\sum_{i=1}^{N} (x_i^u - \mu^u)(x_i^v - \mu^v)$$

Lin correlation (feat u, feat v)

Σ = nice math matrix $\Rightarrow$ <u>orthonormal decomposition</u>

- symm
- pos semidef
- form A^TA

$(e_1,\lambda_1)(e_2\lambda_2)\dots(e_D\lambda_D)$ eigen vector-values

$\circ\ e\Sigma = \lambda e$ "e does not change direction multp by Σ"

E $\qquad$ E^T

$$\underset{\text{covar}}{\Sigma} = \begin{bmatrix} | & | & & | \\ e_1 & e_2 & \cdots & e_D \\ | & | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D \end{bmatrix} \begin{bmatrix} — & e_1 & — \\ — & e_2 & — \\ & \vdots & \\ — & e_D & — \end{bmatrix}$$

eigenvectrs = cols $\qquad$ eigenvalues $\qquad$ eigenvect = rows

$\lambda_1 > \lambda_2 > \dots > \lambda_D \geq 0$ $\quad$ >0

Notes: Why $[X \cdot e]$

① <u>Rotation</u> PCA dim are obtained by projecting X on e_1, e_2, e_D

first dim $\quad X \cdot e_1 = X^1_{new}$ $\quad$ (N×D, D×1, N×1)

$\Rightarrow X^1_{new}$ = lin combination of orig feat with coef given by $e_1 = [e_1^1 e_1^2 \dots e_1^D]$

2nd dim $\quad X \cdot e_2 = X^2_{new}$

② R dim $R \ll D$ (for ex $D=1000$ $R=20$)

$$\underset{N\times D}{X} \cdot \underbrace{\begin{bmatrix} | & | & & | \\ e_1' & e_2' & \cdots & e_{20}' \\ | & | & & | \end{bmatrix}}_{D\times 20} = \underbrace{X_{new}}_{N\times 20}$$

Thery: $R=20$ first PCA-dim $\Longleftrightarrow$ approx Σ with first 20 (eigered, eigen vals)

$$\underbrace{\Sigma}_{\substack{\text{approx} \\ D\times D}} = \underbrace{\begin{bmatrix} | & | & & | \\ e_1 & e_2 & \cdots & e_R \\ | & | & & | \end{bmatrix}}_{D\times R} \underbrace{\begin{bmatrix} | & & | \\ e_{R+1} & \cdots & e_D \\ | & & | \end{bmatrix}}_{} \underbrace{\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_R \end{bmatrix}}_{R\times R} \underbrace{\begin{bmatrix} \lambda_{R+1} & & \\ & \ddots & \\ & & \lambda_D \end{bmatrix}}_{} \underbrace{\begin{bmatrix} — & e_1 & — \\ — & e_2 & — \\ & \vdots & \\ — & e_R & — \end{bmatrix}}_{R\times D} \underbrace{\begin{bmatrix} & e_{R+1} & \\ & \vdots & \\ — & e_D & — \end{bmatrix}}_{}$$

rank deficient at most R

Information lost: $\Sigma_{approx} \simeq \Sigma$ depending on magnitude of ignored eigenvalues.

- $\lambda_{R+1}, \lambda_{R+2}, \ldots \lambda_D$ close to 0 $\Rightarrow \Sigma_{approx}$ very good

$R = ?$ ← don't want miss on large eigenvalues.

- if $\lambda_{R+1} = \lambda_{R+2} = \ldots = \lambda_D = 0$ (actually 0) $\Rightarrow \Sigma_{approx} = \Sigma$