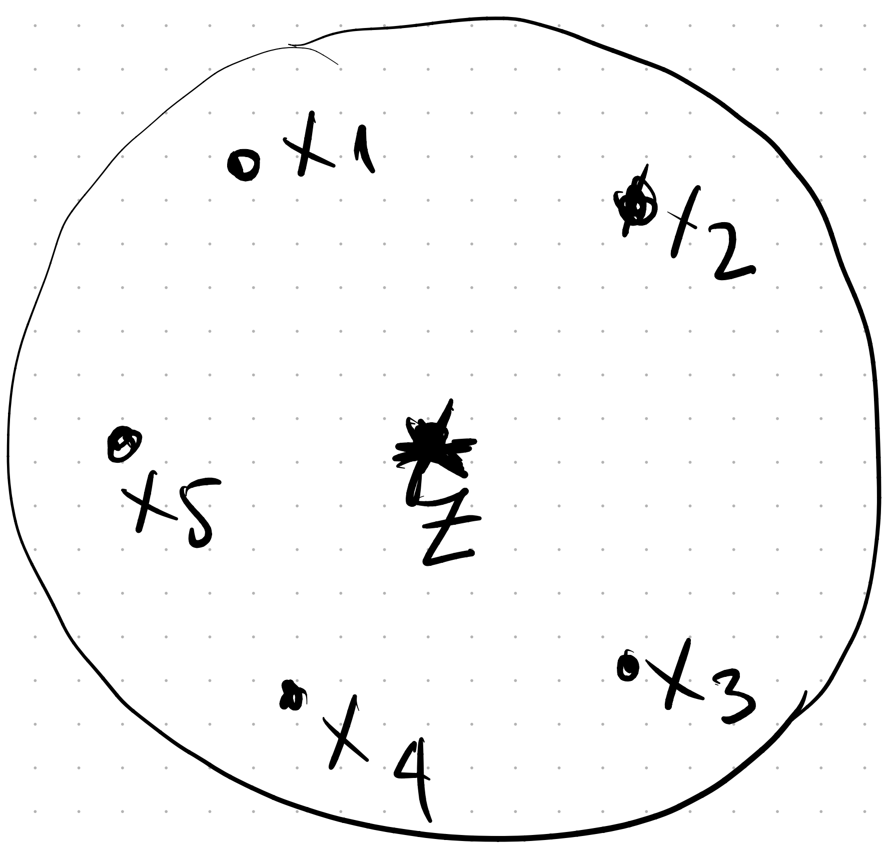
Lecture 7/18

- $\boxed{\text{KNN}}$
- optional: tSNE
- Kernel PCA
- HW5 demo

K Nearest Neighbor: For each point $z \in$ Test

- define neighbourhood $N_z = \{$ set of closest training points to $z \}$
- predict label / class / target by estimating from N_z



classification: predict class Y_z that is most frequent present in N_z $\quad Y_z = \underline{\text{mode}}\{N_z\}$

quantity/regression: predict AVG label $\{N_z\}$

ex y = house price $\Rightarrow$ pred $Y_z = \underset{x_i \in N_z}{\text{AVG}} \{y_i\}$

not really training, just pred Y_Z = estimate from closest neighbors to Z

- need dist/similarity (kernel) across datapoints
  $K_{ij} = \text{sim}(x_i, x_j)$
  $K_{iz} = \text{sim}(x_i, Z)$

appropriate for data and for task

ex x_i = patients
y = diabetes
$\Rightarrow K_{ij}$ = similarity of patients w.r.t diabetes

---

x_i = images
y = prod. price
$\Rightarrow K_{ij}$ = similarity of image prod w.r.t. price (willingness to pay)

---

x_i = email
y = spam/not
$\Rightarrow K_{ij}$ = similarity of emails w.r.t. spam

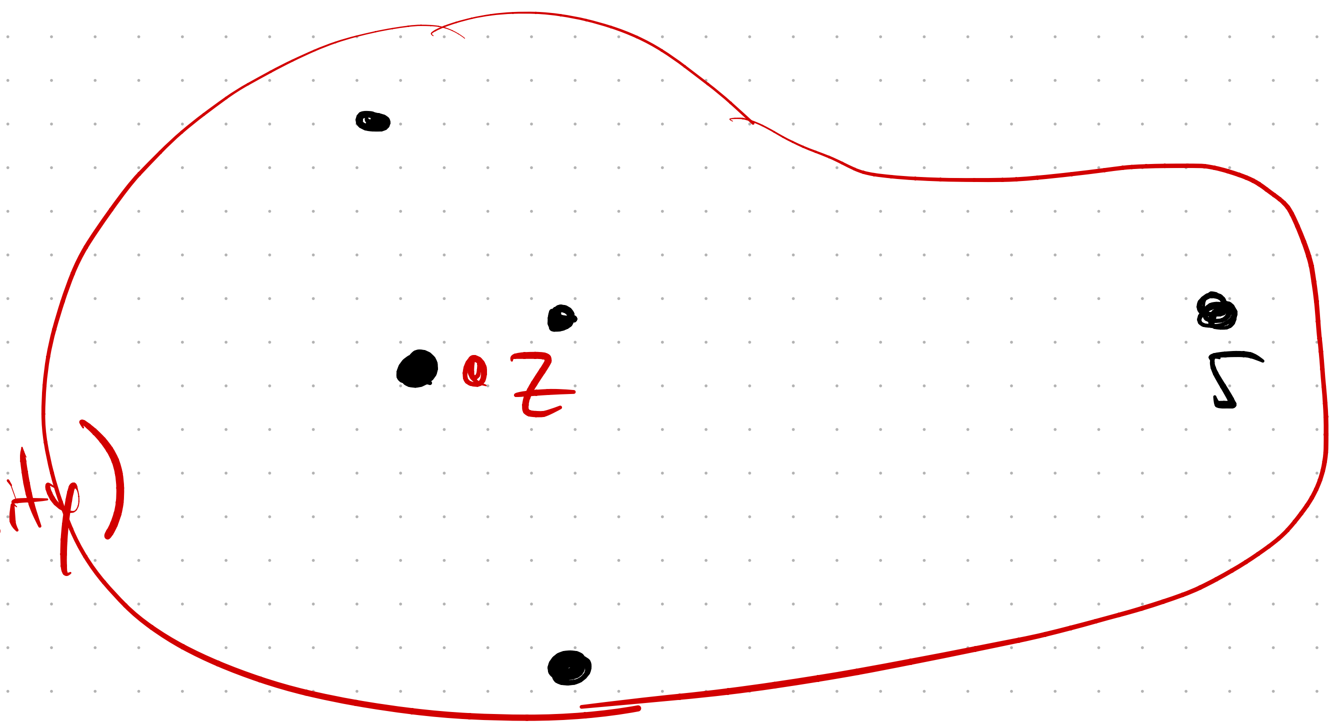
---

x_i = movies
y = rating
$\Rightarrow K_{ij}$ = similarity of movies w.r.t user satisfaction

KNN 3 variants

- K = fixed ; ex ⇒ K = 5. N_z always has 5 training points.
  $N_z = \{$ closest 5 to $z \}$
  - rank all training points by K_{zi} similarity.
  - select top K → N_z.

disadvantage: some z don't have 5 close neighbors (low density)



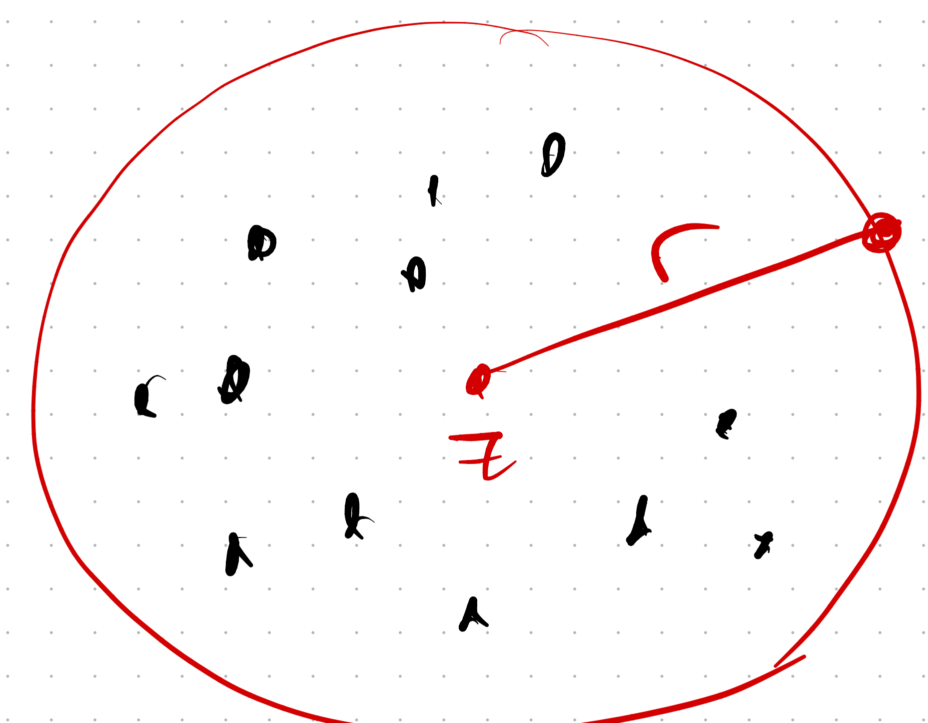
- range = fixed & variable

$N_z = \{ x_i \text{ training} \mid K_{iz} \geq r \}$ ← range

$\text{dist}(i, z) \leq r$

disadvantage: some z very few neighbors in range

some z : many neighbors in range



- use $\boxed{\text{all}}$ training points weighted by similarity $K_{iz} = sim(x_i, z)$

regression $predict(z) = \dfrac{\sum_{i=1}^{N} K_{iz} \cdot y_i}{\sum_{i=1}^{N} K_{iz}}$

labels / targets

weights

classification

class ℓ

Label

$score(z, y_\ell) = \sum_{i=1}^{N} K_{iz} \cdot \boxed{1[y_i = y_\ell]}$

filter

N_z = all training set, but weighted.

K_{iz} = high $\Rightarrow$ $x_i \approx z$ $\Rightarrow$ y_i counts a lot

K_{iz} = low $\Rightarrow$ x_i not similar to z $\Rightarrow$ y_i doesn't count much

Kernel PCA

- rewrite PCA primal $\Longrightarrow$ dual form.

motiv.

dual variables.

$$\text{PCA word} = X \cdot \begin{bmatrix} | & | & & | \\ e_1 & e_2 & \cdots & e_D \\ | & | & & | \end{bmatrix} \overset{?}{=} \underbrace{\boxed{XX^T}}_{K} \cdot \boxed{\beta}$$

eigenvectors (Σ_{covar})
sorted by desc. eigen val.

choose K

- can use any valid kernel K (not just linear XX^T)

  ex $K = \text{gaussian} = e^{-\|x_i - x_j\|^2/2\sigma^2}$

- compute β for that kernel.

compute

$$\beta = \text{eigenvect}(K)$$

intuition: we want eigen vectors (primal var) $= X^T \boxed{\beta}$ dual.

detail (skip): show that $e = X^T \beta$ possible for every e

calculate/find β? : e eigen vector of $\underset{\text{COVAR}}{\Sigma'} \Rightarrow \Sigma' e = \boxed{\lambda} e$ eigen val

derivation: $\Sigma' e = \lambda e$

$$\boxed{\Sigma' = \frac{1}{N} X^T X} \quad \text{covar estimation}$$

$$\boxed{e = X^T \beta} \rightarrow \text{want}$$

$$\underbrace{\boxed{\left(\frac{1}{N} X^T X\right)}}_{\Sigma'} \overbrace{\boxed{X^T \beta}}^{e} = \lambda \overbrace{\boxed{X^T \beta}}^{e}$$

$$X^T X X^T \beta = N \lambda X^T \beta$$

$\cdot X$ left:

$$\boxed{X X^T} \boxed{X X^T} \beta = N \lambda \boxed{X X^T} \beta$$

get rid of K?

$$\cancel{K} \cdot K \cdot \beta = N \lambda \cancel{K} \cdot \beta$$

$$K \cdot \beta = \underbrace{\boxed{N \lambda}}_{\text{scalar}} \cdot \beta \longrightarrow \beta = \text{eigenvector}(K)$$

$N\lambda$ = coresp eigenval