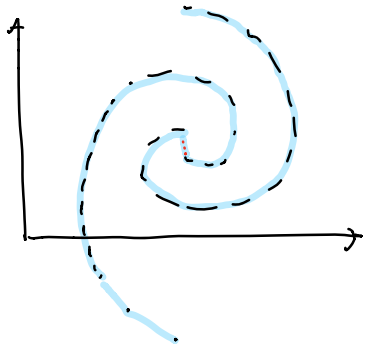
#steps

single linkage example $x \in R^2$



3.24 LEC

avg/center

K-means clustering

↑ # of clusters

Input: n data points $x_1 \cdots x_n$

where $x_i = \begin{pmatrix} x_{i1} \\ \vdots \\ x_{id} \end{pmatrix} \in R^d$

K-means Algorithm:

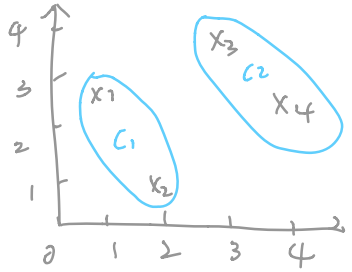
1. Init k cluster centers $c_1 \cdots c_k \in R^d$ (not overlapping)
2. Repeat until no changes in cluster assignment
3. For each item x_i $i = 1 \cdots n$, find its closest cluster center. Let $y_i \in \{1, 2 \cdots, k\}$ be that center
4. For each cluster center c_j, $j = 1 \cdots k$

$$c_j = \frac{\sum_{i: y_i = j} x_i}{\sum \cdots 1}$$

$i:y_i=j$

Ex. in 2D with $n=4$ points and $k=2$ clusters

$x_1=\begin{pmatrix}1\\2\end{pmatrix}$ $x_2=\begin{pmatrix}2\\1\end{pmatrix}$ $x_3=\begin{pmatrix}3\\4\end{pmatrix}$ $x_4=\begin{pmatrix}4\\3\end{pmatrix}$



k-means

① step 1 arbitrary $c_1=\begin{pmatrix}3\\3\end{pmatrix}$ $c_2=\begin{pmatrix}4\\3\end{pmatrix}$

Iteration 1:

$d(x_1, c_1)=|x_1-c_1|$

$y_1=1$ (x_1 belongs to c_1) $y_2=1$ $y_3=1$ $y_4=2$

Update $c_1=\frac{\sum_{i:y_i=1} x_i}{\sum_{i:y_i=1} 1}=\frac{x_1+x_2+x_3}{3}=\begin{pmatrix}\frac{1+2+3}{3}\\ \frac{2+1+4}{3}\end{pmatrix}=\begin{pmatrix}2\\2\frac{1}{3}\end{pmatrix}$

$c_2=\begin{pmatrix}4\\3\end{pmatrix}$

Iteration 2:

$y_1=1$ $y_2=1$ $y_3=2$ $y_4=2$

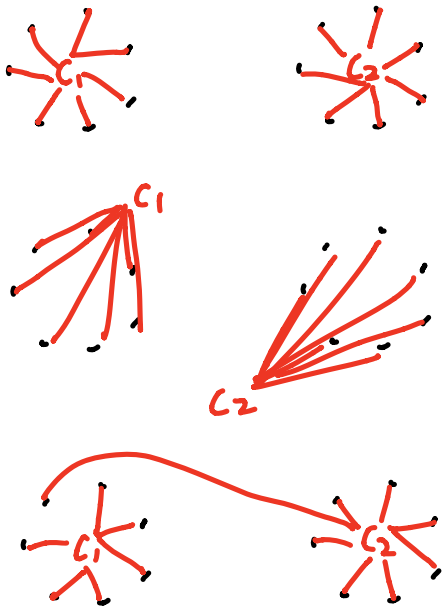
update $c_1=\frac{x_1+x_2}{2}=\begin{pmatrix}\frac{3}{2}\\ \frac{3}{2}\end{pmatrix}$ $c_2=\frac{x_3+x_4}{2}=\begin{pmatrix}\frac{7}{2}\\ \frac{7}{2}\end{pmatrix}$

Iteration 3:

$y_1=1$ $y_2=1$ $y_3=2$ $y_4=2$

Stop k-means, return $c_1 \cdots c_k$, $y_1 \cdots y_k$

$\sum_{i=1}^{n} \|x_i - c_{y_i}\|^2$



good c_1 c_2
good $y_1 \cdots y_n$

Bad c_1 c_0
good $y_1 \cdots y_n$

good $c_1 c_2$, $y_2 \cdots y_n$
bad y_1

The optimization problem of k-means

$$\min_{c_1 \cdots c_k \in R^d \; y_1 \cdots y_n \in \{1 \cdots k\}} \sum_{i=1}^{n} ||x_i - c_{y_i}||^2$$

Fix $c_1 \cdots c_k$, solve $\min_{y_1 \cdots y_n \in \{1 \cdots k\}} \sum_{i=1}^{n} ||x_i - c_{y_i}||^2$

$\Leftrightarrow$ $\min_{y_i \in \{1 \cdots k\}} ||x_i - c_{y_i}||^2$ for $i = 1 \cdots n$

$\Leftrightarrow$ assign x_i to the closest cluster center

$y_i = \arg\min_{j = \{1 \cdots k\}} ||x_i - c_j||$ for $i = 1 \cdots n$

Fix $y_1 \cdots y_n$, optimize $c_1 \cdots c_k$

$$\min_{c_1 \cdots c_k \in R^d} \sum_{i=1}^{n} ||x_i - c_{y_i}||^2 \iff \text{for } j = 1 \cdots k \quad \min_{c_j \in R^d} \sum_{x_i = y_i = j} ||x_i - c_j||^2$$

$$\Leftrightarrow \nabla_{c_j} \sum_{x_i : y_i = j} \|x_i - c_j\|^2$$

$$= \sum_{x_i : y_i = j} \nabla_{c_j} \left((x_i - c_j)^T (x_i - c_j) \right)$$

$$= \sum_{x_i : y_i = j} 2 (x_i - c_j)(-1) = \vec{0}$$

$$\sum_{x_i : y_i = j} (x_i - c_j) = \vec{0}$$

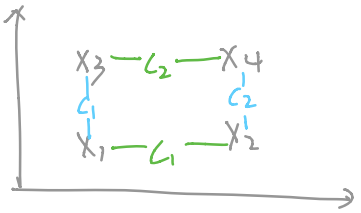
$$\sum_{x_i : y_i = j} x_i - \sum_{x_i : y_i = j} c_j = \vec{0}$$

$$\sum_{x_i : y_i = j} x_i = c_j \sum_{x_i : y_i = j} 1$$

$$c_j = \frac{\sum_{x_i : y_i = j} x_i}{\sum_{x_i : y_i = j} 1}$$

avg of points assigned to j

k-means Algorithm is a **coordinate-descent** alg to heuristically minimize the objective



Local min for k-means
global min

Smart initialization of $c_1 \cdots c_k$: k means ++

① choose c_1 uniformly from $x_1 \cdots x_n$

② For $i = 2 \cdots k$

Let $D(x)$ be the distance from x to the closest initialized cluster center

D^2 weighting choose $c_j = x_i$ according to the distribution

$$P(x_i) = \frac{D^2(x_i)}{\sum_{i=1}^{n} D^2(x_i)} \qquad i = 1 \cdots n$$

3.26 LEC Linear regression

1-D ordinary least squares (OLS) linear regression

Training set: $(x_1 \in \mathbb{R}, y_1 \in \mathbb{R}), \cdots, (x_n, y_n)$

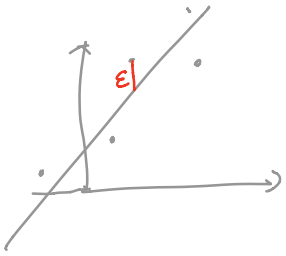
(x_i: item, y_i: label; labeled item)

Training: estimate a good model

Model: linear regression

$$y = \beta_0 + \beta_1 x + \varepsilon \qquad \varepsilon \sim N(0, \sigma^2)$$

(β_0: offset, β_1: slope)



Model parameter $\beta_0, \beta_1, \sigma^2$

Parameter estimation = training

view 1: maximum likelyhood

view 2: least squares

Loss func: $\ell(x, y, \beta_0, \beta_1)$

least squares uses squared loss: $\ell(x, y, \beta_0, \beta_1) = (\beta_0 + \beta_1 x - y)^2$

Risk minimization

$$\text{Risk} = \sum_{i}^{n} \ell(x_i, y_i, \beta_0, \beta_1)$$

$$\hat{\beta}_0, \hat{\beta}_1 = \underset{\beta_0, \beta_1}{\operatorname{argmin}} \sum_{i=1}^{n} \ell(x_i, y_i, \beta_0, \beta_1)$$

2nd order: $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$

d^{th} order: $y = \beta_0 + \beta_1 x + \cdots + \beta_d x^d + \varepsilon$

from input x, define a feature vector $\phi(x) = \begin{bmatrix} 1 \\ x \\ \vdots \\ x^d \end{bmatrix} \in \mathbb{R}^{d+1}$

define $\beta = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_d \end{bmatrix} \in \mathbb{R}^{d+1}$

Linear regression model $y = \phi(x)^T \beta + \varepsilon$

Risk $\underbrace{\sum_{i=1}^{n} (\phi(x_i)^T \beta - y_i)^2}_{\text{Risk}}$ or $\underbrace{\frac{1}{n} \sum_{i=1}^{n} (\phi(x_i)^T \beta - y_i)^2}_{\text{MSE}}$

Estimation (MLE)

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^{d+1}}{\operatorname{argmin}} \sum_{i=1}^{n} (\phi(x_i)^T \beta - y_i)^2$$

Define "design matrix" $X_{n \times p}$ (n: # items, p: # dim in $\phi(x)$)

$$X = \begin{bmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_n)^T \end{bmatrix}_{n \times p} \qquad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}_{n \times 1}$$

$n \times p \quad p \times 1 \quad (n \times 1)$

model: $y = X\beta + \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}$

MLE: $\hat{\beta} = \underset{\beta \in R^p}{\operatorname{argmin}} |X\beta - y|^2$

$\nabla |X\beta - y|^2 = \vec{0}$

$2X^T(X\beta - y) = 0$

$X^TX\beta - X^Ty = 0$

$\underset{p\times p}{X^TX}\ \underset{p\times 1}{\beta} = \underset{p\times n}{X^T}\ \underset{n\times 1}{y}$

assume X^TX is invertible (full rank)

$(X^TX)^{-1}(X^TX)\beta = (X^TX)^{-1}X^Ty$

$\hat{\beta} = (X^TX)^{-1}X^Ty$

Training set $(x_1, y_1) \cdots (x_n, y_n)$

Introduce test set $(x_{n+1}, y_{n+1}) \cdots (x_{n+m}, y_{n+m})$

(Both drawn from same distribution)

We train $\hat{\beta}$ on the training set

For each test item (x^*, y^*)

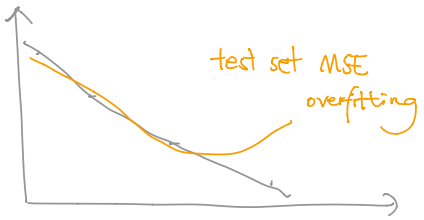
we predict its label as $\hat{y} = x^{*T}\hat{\beta}$

we suffer loss $\ell(x^*, y^*, \hat{\beta}) = (\hat{y} - y^*)^2$

$= (x^{*T}\hat{\beta} - y^*)^2$

on the whole test set, we suffer

MSE $\frac{1}{m}\sum_{i=n+1}^{n+m} \ell(x_i, y_i, \hat{\beta})$



To minimize overfitting, use a tuning set

① get a labeled data set $(x_1, y_1) \cdots (x_N, y_N)$

② randomly split the data set into 3 sets

random shuffle

③ Train $\hat{\beta}^{(0)}, \hat{\beta}^{(1)}, \cdots \hat{\beta}^{(n-1)}$ on training

④ Measure their tuning set MSE

pick the best model

$$\hat{j}^* = \text{argmin}_{\text{tuningset}} \; MSE(\hat{\beta}^{(j)})$$

we pick model $\hat{\beta}^{(\hat{j}^*)}$

⑤ We report test-set MSE with model $\hat{\beta}^{(\hat{j}^*)}$

3.3| LEC K nearest neighbor classifier

KNN algorithm:

↳ Input: $(x_1, y_1) \cdots (x_n, y_n)$ where $x_i \in R^d$, y_i a class label, a distance func $dist(x, x')$

↳ Given a new item $x \in R^d$

↳ Find the k nearest neighbors of x in the training set under dist

↳ Predict a label $\hat{y}$ as the majority label of the k nearest neighbors (break tie arbitrarily)

$loss(x, u, \hat{u}) = \begin{cases} 1 & \text{if } u \neq \hat{u} \\ \end{cases}$

$\ell(x, y, \hat{y}) = \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{if } y \neq \hat{y} \end{cases}$ 0-1 loss

$= \mathbb{1}[y \neq \hat{y}]$

Training set error (rate)

$$\frac{1}{n}\sum_{i=1}^{n} \ell(x_i, y_i, \hat{y}_i) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}[y_i \neq \hat{y}_i]$$

Test set error

$$\frac{1}{m}\sum_{i=n+1}^{n+m} \mathbb{1}[y_i \neq \hat{y}_i]$$

Why need a test set?

→ Machine learning assumes an underlying joint distribution

→ Training set is an independent and identically-distributed (i.i.d) sample from P

True error:

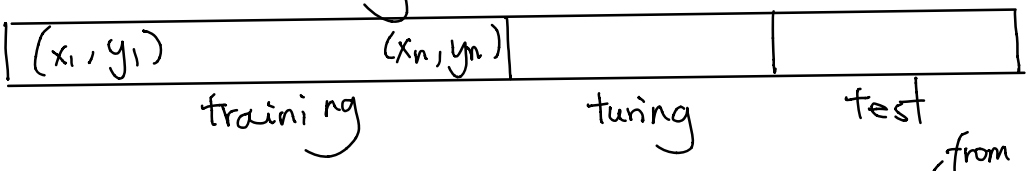
$$\mathop{\mathbb{E}}_{(x,y)\sim P(x,y)} \mathbb{1}[y \neq \hat{y}]$$

↑ not known

Accuracy = 1 - error

How to choose k?

Method 1: use training set



$\hat{k} = \underset{k=1,2\cdots}{\text{argmin}}$ [tuning error with respect to KNN predictions] ← from training set

report $\hat{k}_{KNN}$ prediction's test error

Method 2: cross validation

k-fold cross validation

For $i = 1 \cdots k$

use fold i as the tuning set

all other sets as training set

get tuning error E_i

Pick model parameter (k in KNN)

$$\hat{k}_{KNN} = \underset{k_{KNN}}{\operatorname{argmin}} \frac{1}{k_{fold}} \sum_{i=1}^{k_{fold}} E_i$$

Retrain model on all folds as training set

Report test set error

4.2 LEC Regression

Logistic Regression

→ Input (training data)

$(x_1, y_1) \cdots (x_n, y_n)$

where $x_i = \begin{pmatrix} 1 \\ x_{i1} \\ \vdots \\ x_{id} \end{pmatrix} \in R^{d+1}$

or $y_i \in \{-1, 1\}$ binary classification

$y_i \in \{1, \cdots, k\}$ k-classes

① Binary classification

parameter vector

$$P_w(y=1|x) = \frac{1}{1+\exp(-x^T w)}$$

if $x^T w = 0$ $\quad \frac{1}{1+e^{-0}} = \frac{1}{2}$

if $x^T w = \infty$ $\quad \frac{1}{1+\frac{1}{\infty}} = 1$