

Search David Fleet to get to website.

www.cs.toronto.edu/~fleet/courses/C11/index.html

Learn the notes!

Office hour: 2-3 Friday

3 assignments:

- 3 weeks to work on it $\approx 36\%$
- Quiz after assignment
- Midterm: 15%
- Final: 49%

Piazza site is set up for this class.

3 main types of learning

$$y = f(x; \theta)$$

↑ |
output input x : inputs are measurements
 θ : set of parameters
 y : outcome you care about

data $\{(x_i, y_i)\}_{i=1}^N$ N - training examples

Two classes in supervised learning

① classification - where y is one of distinct number of classes
 $y \in \{1, \dots, C\}$ eg. Binary classification, is this email spam or not spam

② regression - where y lives in real value vector space and contains continuous variables
 $y \in \mathbb{R}$. Eg acceleration in car not only 2 speeds.

2nd type of learning: Unsupervised Learning

No target data.

$$\{x_i\}_{i=1}^N$$

Ex/ Bunch of images or text documents is given to you.

Discovering with data when you don't have a target

3 things with unsupervised data

1) Clustering

- grouping or finding clusters that are similar

3) Density Estimation

This sequence is more probable than this.

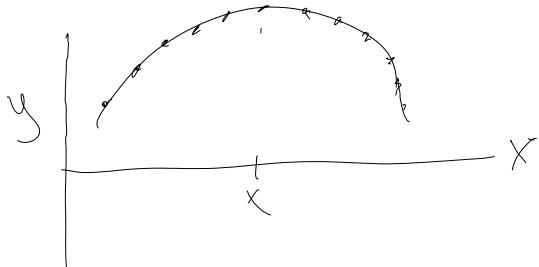
2) feature selection / Dimensionality reduction

finding low dimensional features of high dimensional data in compression.

JPEG is a form of dimensionality reduction

Eg. this is an unlikely image in natural. But this image is highly probable.

Model

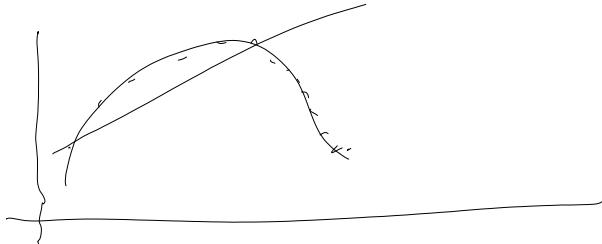


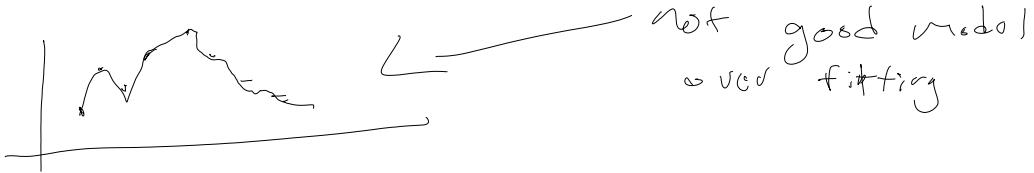
Eg.

A family of functions used to take parameters to make a prediction.

(Good model)

Decision, should I fit a line? or curve?





under fitting: errors in training data is too large

over fitting: fit the data too well.

It's not about fitting accurate model but predicting unseen test data.

In other words don't fit noise.

Eg fit polynomial model



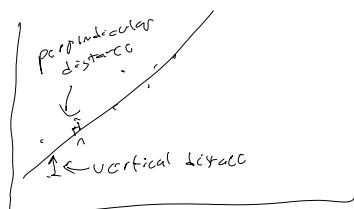
Loss function

measures and quantifies performance on data set

$$y = f(x; \theta)$$

parameter = one that minimizes loss function
cumulative error over the training set.

Loss function

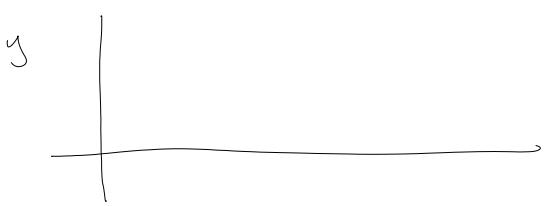


Ex Bin prob $y \in \{-1, 1\}$

$$\text{sgn}(yf(x))$$

1 // -1 . . . 1 . . . -1

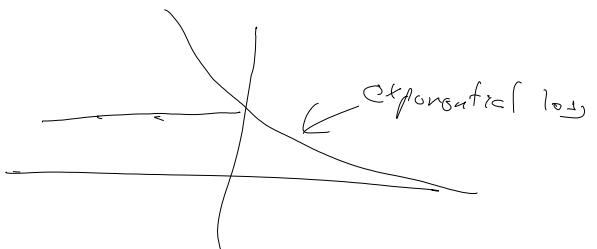
$x \in \mathbb{R}^m$ pred $y \in \{-1, 1\}$



$\text{sgn}(g f(x))$

$$\frac{1}{2}(1 - \text{sgn}(g f(x)))$$

loss function, it oft covers your errors



problem: not differentiable.

how to get a good loss function?

I'll figure it out later. But its important.

Linear Regression

$$\{(x_i, y_i)\}_{i=1}^N, x_i \in \mathbb{R}, y_i \in \mathbb{R}$$

model: $y = f(x) = w x + b$

goal: find good values for w & b .

define error

$e_i = \text{training error}$

$$e_i = y_i - f(x_i) = y_i - (w x_i + b)$$

loss: sum of the squared error

$$E(w, b) = \sum_{i=1}^N e_i^2 = \sum (y_i - (w x_i + b))^2$$

$$\nabla_{w,b} E = 0$$

$$\left(\frac{\partial E}{\partial w}, \frac{\partial E}{\partial b} \right)$$

$$= \sum_{i=1}^N (y_i^2 - 2y_i(wx_i + b) + (wx_i + b)^2)$$

$$= \sum_{i=1}^N y_i^2 - 2y_iwx_i - 2y_ib + w^2x_i^2 + 2bwx_i + b^2$$

$$\frac{\partial E}{\partial b} = \sum_i 0 + 0 - 2y_i + 0 + 2wx_i + 2b$$

$$= -2 \sum_i (y_i - wx_i - b)$$

$$= -2(\bar{y}_i - w\bar{x}_i - Nb)$$

$$b^* = \frac{1}{N} (\bar{y}_i - w\bar{x}_i)$$

$$\bar{y} = \frac{1}{N} \sum y_i \quad b^* = \bar{y} - w\bar{x}$$

$$\bar{x} = \frac{1}{N} \sum x_i$$

objective

$$(E(w, b)) = \sum_{i=1}^N e_i^2 = \sum (y_i - (wx_i + b))^2$$

$$E(w, b^*) = \sum_i (y_i - wx_i - \bar{y} + w\bar{x})^2$$

$$= \sum_i \{(y_i - \bar{y}) - w(x_i - \bar{x})\}^2 \quad \text{Ex/ expand}$$

$$\frac{\partial E}{\partial w} = -2 \sum_i ((y_i - \bar{y}) - w(x_i - \bar{x})) (x_i - \bar{x})$$

$\frac{\partial}{\partial w} \sum_i f(x_i, w)^2$
 $\frac{\partial}{\partial w} \sum_i f(x_i, w)^2$

$$= 2 \sum_i f(x_i, w) \frac{\partial}{\partial w} f(x_i, w)$$

$$= 0 \Rightarrow w^* = \underbrace{\sum_i (y_i - \bar{y})(x_i - \bar{x})}_{\sum (x - \bar{x})^2}$$

rare in ML. Usually a lot! x isn't usually a scalar but a vector of values.

$$\bar{x} \in \mathbb{R}^d$$

$$\bar{x} = (x_1, x_2, \dots, x_d)^\top \Rightarrow d - \text{dimensional vector space}$$

$$y = f(x) = \sum_{j=1}^d w_j x_j + b$$

$$= x^\top \bar{w} + b$$

$$\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_n \end{pmatrix}, \quad \tilde{w} = \begin{pmatrix} \tilde{w}_1 \\ \vdots \\ \tilde{w}_d \end{pmatrix}$$

$$\tilde{y} = f(\tilde{x}) = \tilde{w}^\top \tilde{x}$$

$$e_i = y_i - \tilde{w}^\top \tilde{x};$$

\downarrow error

objective

$$E(\tilde{w}) = \sum_{i=1}^N (y_i - \tilde{w}^\top \tilde{x})^2$$

$$\bar{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \quad \tilde{X} = \begin{bmatrix} \tilde{x}_1^\top & \vdots & () \\ \tilde{x}_2^\top & \vdots & | \\ \vdots & \ddots & | \\ \tilde{x}_N^\top & \vdots & | \end{bmatrix} \quad N \text{ rows}$$

$d+1$ columns

$$\begin{aligned} E(\tilde{w}) &= \|\bar{y} - \tilde{X} \tilde{w}\|_2^2 \quad \rightarrow / \text{two norm squared} \\ &= (\bar{y} - \tilde{X} \tilde{w})^\top (\bar{y} - \tilde{X} \tilde{w}) \quad / \sqrt{\sum \text{ of the elements}} \\ &= \bar{y}^\top \bar{y} - 2\bar{y}^\top \tilde{X} \tilde{w} \quad / \|\bar{v}\|_2^2 = \sum v_j^2 = \bar{v}^\top \bar{v} \end{aligned}$$

Input: $\vec{x} \in \mathbb{R}^d$, $\vec{x} = [x]$

Output: $y \in \mathbb{R}$,

e.g. predict your grade using previous grades.

Note: /Assigh is out

Start it
already!

e.g. predict expected weight of ch. L

possible inputs

- weight

- height

- etc ..

\vec{x} is features, measurements of the world which will base your predictions

$$y = f(\vec{x}) = \tilde{w}^T \vec{x} \leftarrow \text{model fitting hyper plane to data.}$$

If f is a vector instead of a scalar.

$$\vec{y} = f(\vec{x}) = \tilde{W}^T \vec{x}$$

$$\tilde{W} \in \mathbb{R}^{d+1} \times K$$

$$\tilde{W} = [\tilde{w}_1 \dots \tilde{w}_K]$$

$$= \begin{bmatrix} \tilde{w}_1 & \tilde{w}_2 & \dots & \tilde{w}_K \\ b_1 & b_2 & \dots & b_K \end{bmatrix}$$

$$\text{Data: } \{(\vec{x}_i, \vec{y}_i)\}_{i=1}^N$$

j^{th} element of i^{th} training sample

$$\hat{y}_{ij} = \tilde{w}_j^T \vec{x}_i \rightarrow y_{ij}$$

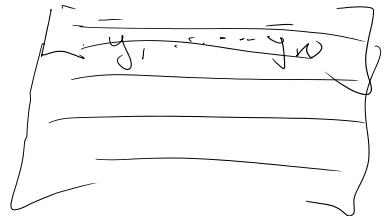
$$= \sum_{i=1}^N \sum_{j=1}^K (y_{ij} - \tilde{w}_j^T \vec{x}_i)^2$$

loss and error are interchangeable in this \Leftrightarrow

The loss is non-negative quantity. When loss = 0, you get a perfect solution.

\tilde{y}_j be j^{th} output for all N training points

y - describes columns
 y' - describes rows



$\tilde{x} = [\tilde{x}_1 \ \tilde{x}_2 \ \dots \ \tilde{x}_n]^T \leftarrow$ with transpose on loss matrix X .

$$E[\tilde{w}] = \sum_{j=1}^k \|\tilde{y}_j - \tilde{x}\tilde{w}_j\|^2$$

output vector
 weight matrix
 weights

, shows $k +$
squared problems

$$E(\tilde{w}) = \|\tilde{y} - \tilde{x}\tilde{w}\|_F^2$$

Frobeneus norm

$$\tilde{y} = [\tilde{y}_1 \ \tilde{y}_2 \ \dots \ \tilde{y}_k]$$

Frobenius norm

normal vector func. not applied to matrices

$$\|A\|_F^2 = \sum_i \sum_j a_{ij}^2$$

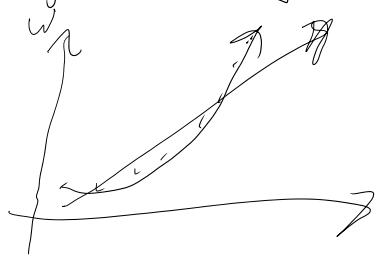
Two norms

$$\|A\|_2^2 = \sum_i \sum_j a_{ij}^2$$

Non linear regression

when you get data plot it, find a way to visualize it.

height vs weight



Think relation between feature and output

non-linearity make life difficult. But class of function easy to solve: Basis Function regression

Basis Function Regression

$$y = f(x) = \sum_{k=1}^K w_k [b_k(x)]$$

— use sin

— use poly

— use linear

Two basis functions I

① polynomials

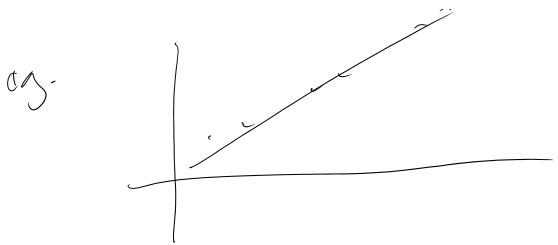
$b_n(x) = x^n \leftarrow x \underset{\text{if } n \geq 1}{\text{is}} \text{ monomial because it has only}$

Weighted sum of monomial is polynomial -

$$f(x) = w_0 + w_1 x + w_2 x^2 + \dots$$

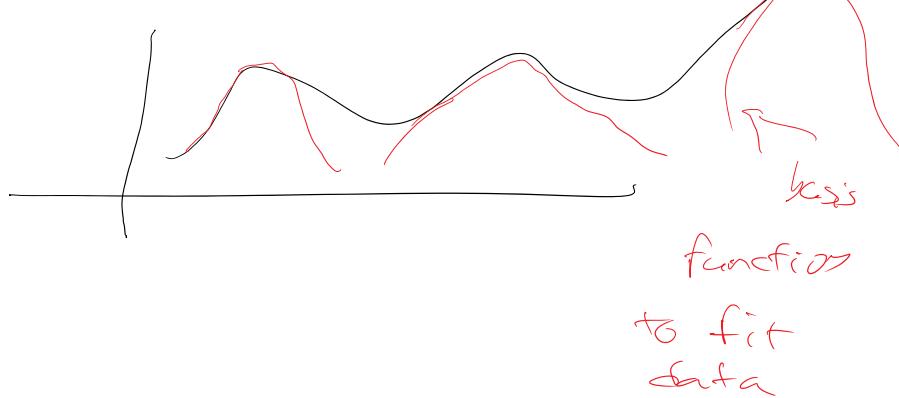
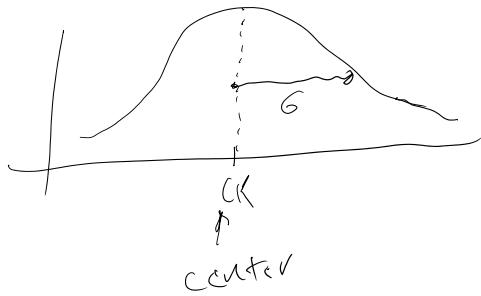
1st assignment - fit polynomial function \rightarrow data.





Radio Basis Function : (RBF)

$$b_k(x) = e^{-\frac{(x - c_k)^2}{G^2}} \quad \leftarrow \text{looks similar to normal}$$



$$y = f(x) = \sum w_k b_k(x)$$

$$\sum (x_i, y_i) \}_{i=1}^N$$

Estimation

$$E(\bar{w}) = \sum_i (y_i - f(x_i))^2$$

$$\bar{w} = (w_1, \dots, w_K)^T \quad , \quad \bar{y} = (y_1, \dots, y_N)^T$$

$$B = [B_{ik}] = B_{ik} = b_k(x_i)$$

$$= \begin{bmatrix} b_1(x_1) & b_2(x_1) & \dots & b_K(x_1) \\ b_1(x_2) & b_2(x_2) & \dots & b_K(x_2) \\ \vdots & & & \\ b_1(x_N) & \dots & \dots & b_K(x_N) \end{bmatrix} \quad \leftarrow \begin{array}{l} \text{feature set for } x, \\ \text{basis function vary} \\ \text{each row} \end{array}$$

qf

all the inputs vary

$$\bar{E}(\bar{\omega}) = \|\bar{y} - B\bar{\omega}\|^2$$
$$\text{objective} = (\bar{y} - B\bar{\omega})^T (\bar{y} - B\bar{\omega})$$
$$=$$

} how do we find optimal $\bar{\omega}$?

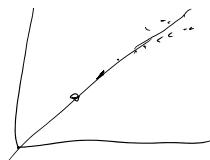
Take gradient of E

$$\nabla E = \frac{\partial E}{\partial \bar{\omega}} = 0$$
$$\Rightarrow \bar{\omega}^* = (B^T B)^{-1} B^T \bar{y}$$

↑
(cos) +
several
functions

Reasons for overfitting

- Too many parameters and too few data.
- modelling the noise process itself.
- model uncertainty



Regularization

Smooth models are better than non smooth

data



if you got a good fit using 4 basis functions, it's better than 20 basis functions.

$$E(\bar{\omega}) = \|\bar{y} - B\bar{\omega}\|^2 \leftarrow \text{objective function}$$

$$= \underbrace{\|\bar{y} - B\bar{\omega}\|^2}_{\text{data term}} + \lambda \|\bar{\omega}\|^2 \leftarrow \begin{array}{l} \text{extra terms} \\ \text{to penalize } \bar{\omega}'s \\ \text{that are big} \\ \langle \sum \omega^2 \rangle \end{array}$$

regulation
parameter
 L_2 controls
balance

smoothness term

$$y = f(x) = \omega_1 x + \omega_2 x^2 + \omega_3 \dots$$

$$\frac{\partial f}{\partial x} = \omega_1 + 2\omega_2 x + \dots$$

$$\frac{\partial f}{\partial x} = 0 + 2\omega_2 + \dots$$

$$\begin{aligned} E(\bar{\omega}) &= \|\bar{y} - B\bar{\omega}\|^2 + \lambda \|\bar{\omega}\|^2 \\ &= (\bar{y} - B\bar{\omega})^\top (\bar{y} - B\bar{\omega}) + \lambda \bar{\omega}^\top \bar{\omega} \\ &= \bar{\omega}^\top B^\top B \bar{\omega} + \lambda \bar{\omega}^\top \bar{\omega} - 2 \bar{\omega}^\top B^\top \bar{y} + \bar{y}^\top \bar{y} \\ &= \bar{\omega}^\top (B^\top B + \lambda I) \bar{\omega} - 2 \bar{\omega}^\top B^\top \bar{y} + \bar{y}^\top \bar{y} \end{aligned}$$

$$\cancel{E} = 0$$

$$2(B^\top B + \lambda I)\bar{\omega} - 2B^\top \bar{y} = 0$$

$$\therefore \bar{\omega}^* = B^\top B + \lambda I \cancel{B^\top \bar{y}}$$

\leftarrow inverse

Ridge regression

form of bias regression using regularization parameter to encourage your model to be smooth.

Regression

We have been using parametric models

parametric vs non parametric model?

parametric models have fixed # of parameters. # of parameters in the model does not depend on the training set. You don't have to remember the training data.



radio basis model is taking data in the local neighbourhood and fitting it with a bump.

Non-parametric model ex/

$$\{(x_i, y_i)\}_{i=1}^N$$

K-nearest neighbour Regression (k-NN Regression)

Given a test point x get the nearest points to the x and average them.

$$y = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$

set of neighbours Indices into training data given point x

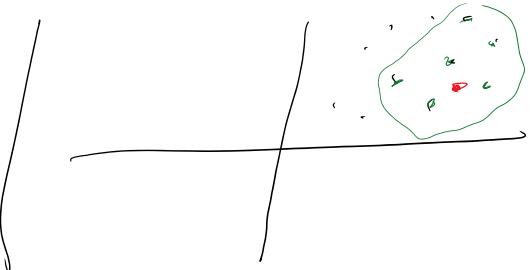
we don't usually choose $k=1$ because data is noisy

Think of k as a parameter like spacing of rbf functions

$k \sim$ hyperparameter.

hyperparameter of polynomial - power/order

larger value of k , smoother the fit



* - training point

* - nearest neighbor
point, assign
them

Estimation Theory, Bayes' optimal form of data fitting and prediction

Uncertainty

Prob: assign beliefs to events without observing them.

A: die shows 3

$$P(A) = \frac{|A|}{|S|} = \frac{1}{6}$$

B: die shows 1

$$P(B) = \frac{1}{6}$$

C: dice sum to 8, 2 dice

$$P(C) = \frac{|C|}{|B|} = \frac{6}{6^2} = \frac{1}{36}$$

$\begin{matrix} 5, 3 \\ 3, 5 \\ 2, 6 \\ 6, 2 \\ 4, 4 \end{matrix} \Bigg) \quad |C| = 6$

$$P(A, C) = P(A \cap C) \\ = \frac{1}{36}$$

$$P(B, C) = 0$$

$$P(C | A) = \frac{P(A \cap C)}{P(A)} = \frac{\frac{1}{36}}{\frac{1}{6}} = \frac{1}{6}$$

$$\perp P(A, B) = P(A) P(B)$$

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$p(A) = \sum_i p(A, B_i) \leftarrow \begin{array}{l} \text{Joint} \\ \uparrow \\ \text{marginal.} \end{array}$$

heteroschasticity; some data points have more noise than others



Bayes Rule

$P(M | D)$ \longrightarrow A condition on the data
 ↑
 model Data
 "Posterior distribution" - probability distribution
 of the model after
 you seen the
 data
 \Rightarrow allows you to say which models are conditioned
 to my data.

Bayes rule says $P(M | D) = \frac{P(D|M) p(M)}{P(D)}$

likelihood: replicability you will see the data with the given data
 prior - big flipping coin
 prior belief is
 evidence

Inference: In ML, computing prob dist' unknown parameter
 interest. Estimating prob distribution over unknown

$$p(M | D)$$

Estimation: estimating single model from data
 "best model"

1) MAP (maximum a posteriori) Estimation (model)

A model that maximizes

$$\pi(\theta) \cdot \text{prior}(\theta) \cdot \text{evidence}(\theta)$$

A model that maximizes

$$\underset{\theta}{\underset{\text{parameter of model}}{\text{MAP}}} = \underset{\theta}{\operatorname{argmax}} p(\theta | D) = \underset{\theta}{\operatorname{argmax}} p(D|\theta)p(\theta)$$

specific instance of the model class which
maximize posterior distribution

2) ML (Maximum likelihood) Estimation

$$\underset{\theta}{\operatorname{argmax}} p(D|\theta)$$

Difference between map and likelihood is we
are not using prior belief

Note: equal if uniform distribution.

N coin flips: $c_{1:N} = (c_1, c_2, \dots, c_N)$

c_1, \dots, c_N - outcome

flip 1: N times.

$$\underset{\theta}{\underset{\text{parameter}}{\text{P}}} : p(H) \quad \text{equivalently} \quad \begin{aligned} p(c=H) &= \theta \\ p(c=T) &= 1 - \theta \end{aligned}$$

prior assume \perp

$$p(\theta) = 1 \quad \underbrace{p(c_{1:N} | \theta)}_{\text{likelihood}} = \prod_{i=1}^N p(c_i | \theta)$$

$$p(\theta | c_{1:N}) = \frac{p(c_{1:N} | \theta) p(\theta)}{p(c_{1:N})}$$

estimation:

map

r_1, r_2, \dots, r_N

estimation:

$$\begin{aligned}\hat{\theta}^* &= \arg \max_{\theta} p(c_{1:N} | \theta) p(\theta) \\ &= \arg \max_{\theta} \log (p(c_{1:N} | \theta) p(\theta)) \quad \left. \begin{array}{l} \text{use falso} \\ \text{in interchangeable} \end{array} \right\} \\ &= \arg \min_{\theta} (-\log p(c_{1:N} | \theta) p(\theta))\end{aligned}$$

$$N = 1000$$

$$\# \text{ of heads} = 750$$

then

$$\# \text{ of tails} = 250, \text{ assume no coin landed on edge}$$

$$p(c_{1:N} | \theta) = \prod_{i=1}^{1000} p(c_i | \theta) = \theta^{750} (1-\theta)^{250}$$

$$c_{1:5} = H H T H T$$

$$\begin{aligned}p(c_{1:5} | \theta) &= p(c_1 | \theta) p(c_2 | \theta) p(c_3 | \theta) \dots \\ &= \theta \theta (1-\theta) (1-\theta) \\ &= \theta^3 (1-\theta)^2\end{aligned}$$

$$p(\theta | c_{1:1000}) = k \underbrace{p(c_{1:1000} | \theta)}_{\theta^{750} (1-\theta)^{250}} \underbrace{p(\theta)}_{1}$$

$$= K \theta^{750} (1-\theta)^{250}$$

minimize neg log posterior

$$= -\log(K) - 750 \log(\theta) - 250 \log(1-\theta), \text{ by proportion dawg}$$

Minimize it; # take derivative, set it = 0

$$\frac{d}{d\theta} (p(\theta | c_{1:100}))$$

$$0 - \frac{750}{\theta} + \frac{250}{1-\theta} = 0$$

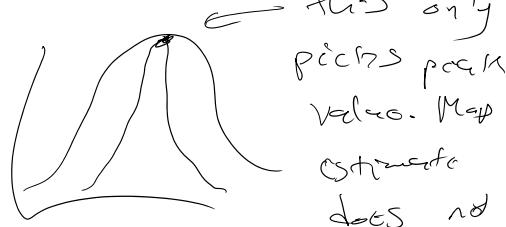
$$\frac{750}{\theta} = \frac{250}{1-\theta}$$

$$750(-\theta) = 250(\theta)$$

$$750 = 250(\theta) + 750(-\theta)$$

$$\frac{750}{1000} = \theta^* \text{ or } \theta_{\text{map}} \text{ frequency of occurrence.}$$

This says map estimate = $\frac{\bar{H}}{N}$



$$E[\theta] = \int \theta p(\theta | c_{1:100}) d\theta$$

$$= \int \theta K \theta^{750} ((-\theta)^{250} \times 1^{100}) d\theta$$

;

$$= \frac{750+1}{750+250+2}$$

$$= \frac{751}{1002} \quad \text{In general} = \frac{H+1}{N+2} \quad \begin{cases} \text{Bayes estimate} \\ (\text{mean}) \end{cases}$$

Regression

\nwarrow d-dimensional weights: treat as r.v

/ $u \in \mathbb{R}$

Regression

$$y = \bar{\omega}^T \bar{x} + \varepsilon \quad (= \text{model})$$

↑
scalar
d-dimensional set of measurements

d-dimensional weights: treat as r.v.
source of noise

$$\left\{ \begin{array}{l} y \in \mathbb{R} \\ \bar{x} \in \mathbb{R}^d \\ \bar{\omega} \in \mathbb{R}^d \\ \varepsilon \sim N(0, \sigma^2) \\ \underbrace{\qquad}_{\text{uncertainty in measurements}} \end{array} \right.$$

mean of y : $\bar{\omega}^T \bar{x}$

variance: σ^2

$y \sim$ variable that is gaussian

$$y | \bar{x}, \bar{\omega} \sim N(\bar{x}^T \bar{\omega}, \sigma^2)$$

$$p(y | \bar{\omega}, \bar{x}) = \frac{1}{\sqrt{2\pi \sigma^2}} e^{-\frac{(y - \bar{\omega}^T \bar{x})^2}{2\sigma^2}}$$

\downarrow dimension vs
0 weight & different weights

\downarrow mean variance

$p(\bar{\omega})$ assume $\bar{\omega} \sim N(0, \alpha I)$

Isotropic: prob distribution is the same direction

$$\begin{aligned} p(\bar{\omega}) &= \prod_{i=1}^d p(\omega_i) \\ &= \prod_{i=1}^d \frac{1}{\sqrt{2\pi\alpha}} e^{-\frac{1}{2\alpha}\omega_i^2} \\ &= \left(\frac{1}{\sqrt{2\pi\alpha}} \right)^d \frac{1}{2^d \alpha^{\frac{d}{2}}} e^{-\frac{1}{2\alpha}\bar{\omega}^T \bar{\omega}} \end{aligned}$$

Hand in written work in the drop box

IC400 - Drawel CSC11

Hand in code online. Assignment due before lecture. Quiz during lecture.

Midterm: After the reading week

Date: October 20th, Saturday Afternoon.

No cheat sheet.

Regression

$$y = \bar{x}^T \bar{w} + \epsilon$$

$$\bar{x} \in \mathbb{R}^d \quad \epsilon \sim N(0, \sigma^2 I)$$

$$\bar{w} \in \mathbb{R}^d$$

$$P(y | \bar{x}, \bar{w}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \frac{(y - \bar{x}^T \bar{w})^2}{\sigma^2}}$$

$$P(\bar{w}) \sim N(0, \frac{\alpha}{2} I)$$

Covariance matrix which is
Isotropic width or variance
at all directions

$$P(\bar{w}) = \frac{1}{(2\pi\alpha)^{\frac{d}{2}}} e^{-\frac{1}{2\alpha} \bar{w}^T \bar{w}}$$

Posterior

$$D = \sum_{i=1}^N (\bar{x}_i, y_i)$$

$$P(\bar{w} | D) = \frac{P(y_{1:N} | \bar{x}_{1:N}, \bar{w}) P(\bar{w})}{P(y_{1:N})} \quad \text{assume } \bar{w} \perp \text{ noise}$$

assume noise
 \downarrow
 \perp

$$\text{Independence } y_i = \bar{x}_i^T \bar{w} + \epsilon_i \quad \sum_i \epsilon_i \text{ is noise}$$

$$y_j = \bar{x}_j^T \bar{w} + \epsilon_j$$

$$P(\epsilon_1, \epsilon_2) = P(\epsilon_1) P(\epsilon_2)$$

$$\Rightarrow P(y_i, y_j | \bar{x}_i, \bar{x}_j, \bar{w}) = P(y_i | \bar{x}_i, \bar{w}) P(y_j | \bar{x}_j, \bar{w})$$

$$P(\bar{w} | D) = \prod_{i=1}^N \frac{P(y_i | \bar{x}_i, \bar{w}) P(\bar{w})}{C} \quad \text{C raised to quadratic multiplied by a raise}$$

$$P(y_{1:N} | \bar{x}_{1:N}, \bar{w}) = \left(\frac{1}{2\pi\sigma^2} \right)^{\frac{N}{2}} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \bar{x}_i^T \bar{w})^2} \quad \text{to the quadratic is sum of quadratic}$$

$$P(\bar{w} | D) = C' \left(\frac{1}{2\pi\sigma^2} \right)^{\frac{N}{2}} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \bar{x}_i^T \bar{w})^2} \left(\frac{1}{2\pi\alpha} \right)^{\frac{d}{2}} e^{-\frac{1}{2\alpha} \bar{w}^T \bar{w}}, \quad \text{take -log}$$

- log of product
 \downarrow
 \downarrow sum of
 \downarrow

$$-\log P(\bar{w} | D)$$

$$= -\cancel{\log C} + \frac{N}{2} \cancel{\log(2\pi\sigma^2)} + \frac{1}{2} \cancel{\log(2\pi\alpha^2)} + \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \bar{x}_i^T \bar{w})^2 + \frac{1}{2\alpha} \bar{w}^T \bar{w}, \quad \text{disappear}$$

$$\begin{aligned}
 & -\log p(w|D) \\
 & = -\cancel{\log \sigma} + \frac{N}{2} \cancel{\log(2\pi\sigma^2)} + \frac{1}{2} \cancel{\log(2\alpha)} + \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \bar{x}_i^T \bar{w})^2 + \frac{1}{2\alpha} \bar{w}^T \bar{w}, \quad \text{disappear because it doesn't depend on } w. \\
 & -\log p(\bar{w}|D) = k + \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \bar{x}_i^T \bar{w})^2 + \frac{1}{2\alpha} \bar{w}^T \bar{w}
 \end{aligned}$$

#Take derivative and set it to zero

#multiply by a constant

$$\begin{aligned}
 & = k' + \sum_{i=1}^N (y_i - \bar{x}_i^T \bar{w})^2 + \left(\frac{2\sigma^2}{2\alpha} \right) \bar{w}^T \bar{w} \\
 & \quad \text{, form of Ridge Regression} \\
 & \lambda = \frac{2\sigma^2}{2\alpha}
 \end{aligned}$$

What is a gaussian?

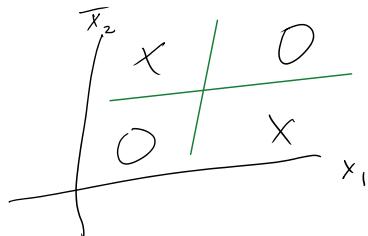
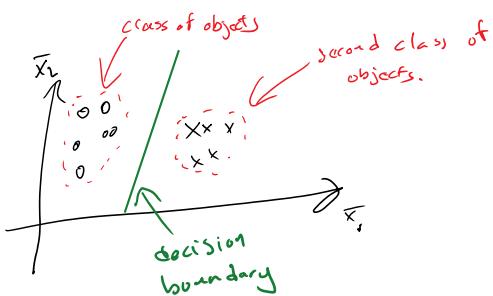
What is a prior?

Classification

$$y = f(x)$$

$$y \in \{-1, 1\}$$

$$y \in \{1, \dots, k\}$$



How to determine if X or O? Guess based on boundaries

Complexities are based on the model

linear separable: A linear function can separate the data.

Classification by regression:

$$\text{data: } \{(x_i, y_i)\}_{i=1}^N, y_i \in \{-1, 1\}$$

#Find the best separating hyperplane

$$\underset{w}{\text{weights}} \rightarrow w^* = \arg \min_{\bar{w}} \sum_{i=1}^N (y_i - \bar{x}_i^T \bar{w})^2 \quad \leftarrow \text{squared error}$$

$$f(\bar{x}) = \bar{w}^T \bar{x}$$

The classifier is the sign of $f(x)$

$$\text{classifier } \text{sgn}(f(x)) = \text{sgn}(\bar{w}^T \bar{x})$$

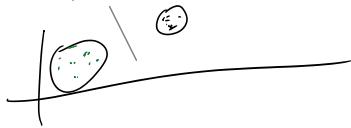
$\bar{w}^T \bar{x} > 0$, above hyper plane

$\omega^T(x) < 0$, below the hyper plane

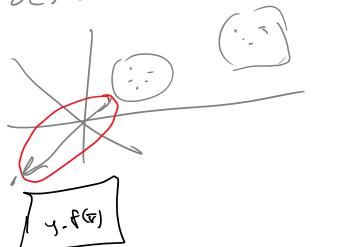
$$\text{sign}(z) = \begin{cases} -1 & z \leq 0 \\ 1 & z > 0 \end{cases}$$

This works well in only certain situations

- points closely clustered together
- spread is more or less the same



Doesn't work well for



since loss function
is huge



$$\frac{0-1 \text{ loss}}{L_{0-1}(y_i, f(x))} \leftarrow \underbrace{\quad}_{y f(x)}$$

$\sum_i L_{0-1}(y_i, f(x_i)) \leftarrow$ loss function that
counts # of mistakes

problem? hard to optimize

Suppose $f(x) = -2$

$$y = 1$$

$$f(x) \times (y) = -2 \leftarrow \text{error.}$$

penalize you for confident and right and
confident & wrong.

K-NN Classifier

Find the K nearest neighbours to test inputs (with indices in $N_k(x)$)

Let $y = \text{sign}\left(\sum_{i \in N_k(x)} y_i\right) \leftarrow$ This is a way of voting. Sum votes and look at the sign.

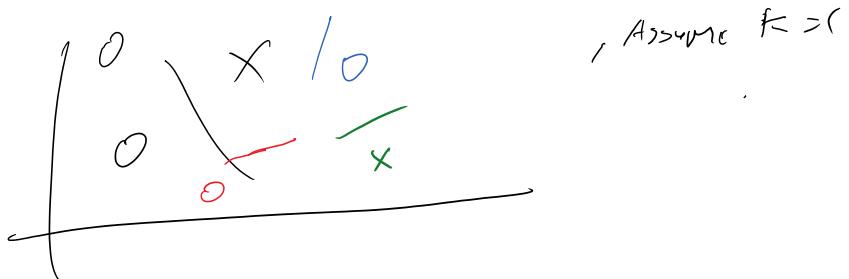
$$w(v_i) = e^{-\frac{1}{2\sigma^2} \|x - v_i\|^2} \quad \# \text{Question on mid-term how do } K-\text{NN classifier well.}$$

Bad: search, lots of point then you have to search all the points.

This is non parametric method, no fixed parameter. Algorithm depends on the amount of data.
parametric model: # of parameter doesn't change.

non parametric would do feature learning.

what does decision boundary look like



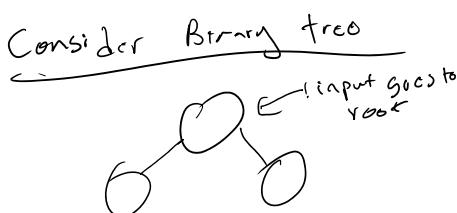
Decision boundary can be complex and can overfit.

larger value of $k \xleftarrow{\# \text{ of nearest neighbour}} \Rightarrow$ blurring out decision boundary

Decision Tree \hookrightarrow go to, often used in practice

stump \rightarrow tree \rightarrow forest

Decision tree can be used for regression or classification.



$$\begin{cases} M \text{ internal nodes} - \text{split nodes} \\ m+1 \text{ leaf nodes} \\ \text{at internal node } j \\ t_j(x) : \mathbb{R} \rightarrow \{-1, 1\} \end{cases}$$
$$T_j(x) = \begin{cases} -1, & \text{take left branch} \\ 1, & \text{take right branch} \end{cases}$$

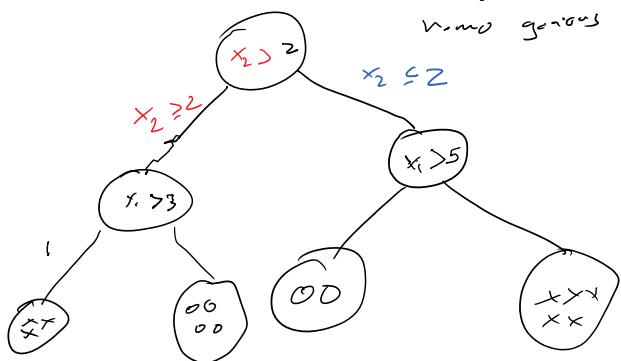
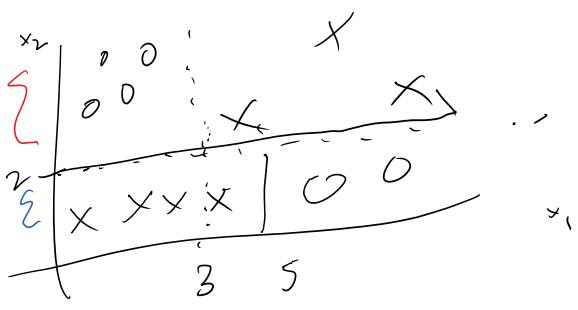
1. input goes to root of tree

2. gonna ask a feature eg $x \geq 2$ or $2 \leq x$

then go to the new node.

3. New tree, $x \geq 3$ or $x \leq 3$ then go down the new node.

goal of split: Separate class of nodes. Split it for nano it nano genres



Once you get to leaf node, take a vote-

of regions grows exponentially depending on depth of the tree.

⑧ leaves of tree define

$$p(c|\text{node } j) = \frac{N_{jF}}{N_j} \quad \left. \begin{array}{l} \text{fraction of data points that reach} \\ \text{that node that} \end{array} \right. \text{class } c.$$

N_j = # points reach node j

N_{jF} # points in node j with class c

Midterm covers everything up to today.

October 20th 5:00 - 6:00.

Decision Tree

Sequence of decisions, leaf nodes will give you prob distribution over classes.

Internal nodes $t_j(\bar{x}) : \mathbb{R}^d \rightarrow \{-1, 1\}$

Leaf nodes, \Leftrightarrow node j with
 N_j items $\text{prob}(y=c | \text{node } j)$

Leaf node j , with N_j items

$N_{j,c}$ with class C

$$P(y=c | j) = \frac{N_{j,c}}{N_j}$$

Decision Trees

node j $t_j(\bar{x})?$
split function

Given m -items $\{\bar{x}_i\} = D_j$ data at node j
(to)



what will the split function look like?

Assume univariate split

↳ split function that only depends on one variable