September 7, 2018 11:13 AM

Search David fleet to get to website.

www.cs.toront.edu/ufleet/courses/C11/index.html

Learn the notes!

Office hour: 2-3 friday

3 assignment:

- 3 week to work on it
- Quiz after assignment $\approx$ 36%

- midterm: 15%
- final: 49%

piazza site is set up for this class.

3 main types of learning

$y = f(x; \theta)$ x: inputs are measurements
↑ ↑ θ: set of parameters
output input y: outcome you care about

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

Two classes in supervised learning

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

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

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

2) feature selection / Dimensionality reduction

finding low dimensional feature of high dimensional data → compression.

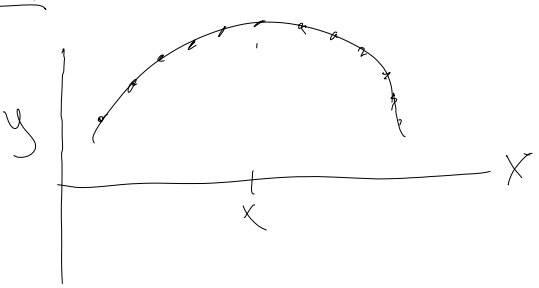
JPEG is a format of dimensionality reduction

3) Density Estimation

This sequence is more probable than this.

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

Model

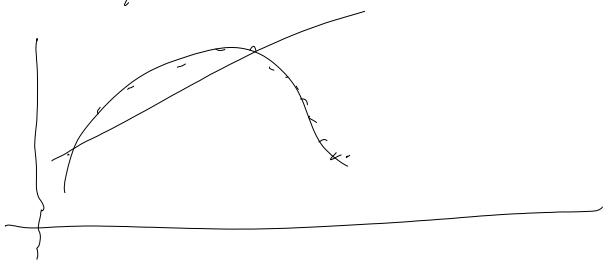


A family of function used to tune parameters to make a prediction.

eg.

Good model:

Person, should I fit a line? or curve?



not good model
over fitting

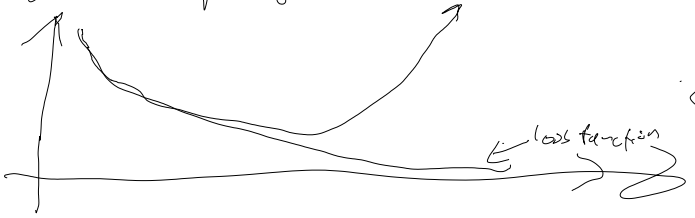
underfitting: errors in training data is too large

overfitting: fit the data too well.

Its not about fitting accurate model but predicting unseen test data.

In other words don't fit noise.

Eg fit polynomial model



← tells noise vs underlying process

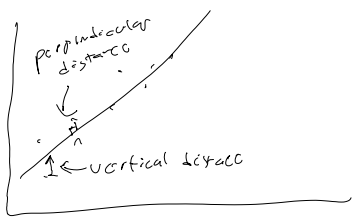
Loss function

measures and quantifies performance on data set

$y = f(x; \theta)$

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

loss functions



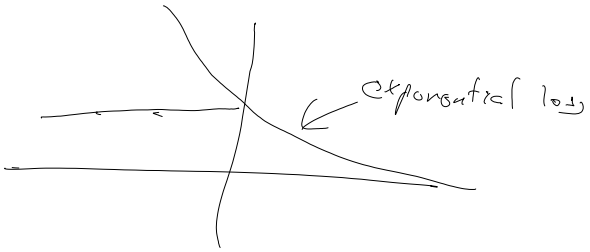
Ex Bin prob $y \in \{-1, 1\}$ $\text{sign}(yf(x))$

ex binary problem $y \in \{-1, 1\}$ $\text{sgn}(yf(x))$

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

loss function, # of errors you make

problem: not differentiable.



how to get a good loss function?

IDK 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: $y = f(x) = wx + b$

goal: find good values for w & B.

define error

e_i = training error

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

Loss: sum of the squared error

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

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

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

$$= \sum \left(y_i^2 - 2y_i(wx_i + b) + (wx_i + b)^2 \right)$$

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

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

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

$$= -2 \left(\sum_i y_i - w \sum_i x_i - Nb \right)$$

$$b^* = \frac{1}{N} \left(\sum_i y_i - w \sum_i x_i \right)$$

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

$$x \quad \frac{1}{N} \sum x_i$$

objective

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

b^*

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

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

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

Recall

$$\frac{d}{dw} \sum_i f(x_i, w)^2 = 2 \sum_i f(x_i, w) \frac{d}{dw} f(x_i, w)$$

$$= 0 \Rightarrow w^* = \frac{\sum (y_i - \hat{y})(x_i - \hat{x})}{\sum (x_i - \hat{x})^2}$$

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

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

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

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

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

$$\tilde{x} = \begin{pmatrix} \bar{x} \\ 1 \end{pmatrix}, \quad \tilde{w} = \begin{pmatrix} \bar{w} \\ b \end{pmatrix}$$

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

$$e_i = y_i - \tilde{w}^T \tilde{x}_i$$

↳ error

objective

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

$$\bar{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \qquad \tilde{X} = \begin{bmatrix} \bar{x}_1^T & : & 1 \\ \bar{x}_2^T & : & 1 \\ \bar{x}_N^T & : & 1 \end{bmatrix} \quad N \text{ rows}$$

$d+1$ columns

$$E(\tilde{w}) = \|\bar{y} - \tilde{X}\tilde{w}\|_2^2 \rightarrow$$

two normed squares

$\sqrt{\sum}$ of the elements

$$\|v\|_2^2 = \sum v_j^2 = \bar{v}^T v$$

$$= (\bar{y} - \tilde{X}\tilde{w})^T (\bar{y} - \tilde{X}\tilde{w})$$

$$= \bar{y}^T y - , 2w$$

Input: $\vec{x} \in \mathbb{R}^d$, $\tilde{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}$

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

Notice: Assign is out

Start it already!

eg. predict your grade using previous grades.

eg. predict expected weight of child
possible inputs
- weight
- height
- etc...

x is features, measurements of the world which will base your prediction

$y = f(\vec{x}) = \tilde{w}^T \tilde{x}$ ← model fitting hyperplane to data.

If f is a vector instead of a scalar.

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

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

$\tilde{W} = [\tilde{w_1} \cdots \tilde{w_k}]$

$= \begin{bmatrix} w_1 & w_2 & \cdots & w_k \\ b_1 & b_2 & \cdots & b_k \end{bmatrix}$

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

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

$\hat{y}_{ij} = \tilde{w}_j^T \tilde{x}_i \longrightarrow y_{ij}$ (prediction)

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

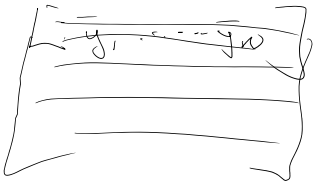
loss and error are interchangable in this class

the loss is non-negative quantity. when loss = 0, you get a perfect solution.

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

pg. 5

y – describes columns
y' – describes rows



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

output target vector, matrix, weights

$$E[\tilde{w}] = \sum_{j=1}^{K} \| \tilde{y}_j' - \tilde{X}\tilde{w}_j \|^2$$

, sums K squared problems

$$E(\tilde{w}) = \| Y - \tilde{X}\tilde{W} \|_F^2$$ ← Frobenius norm

$$Y = [\tilde{y}_1' \ \tilde{y}_2' \ \dots \ \tilde{y}_K']$$

Frobenius norm

normal vector form now applied to matrices

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

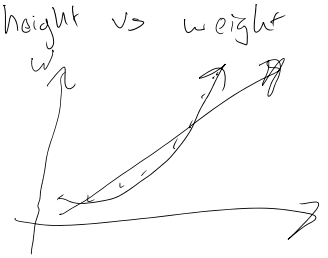
Two norm

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

Non linear regression

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

height vs weight



Think relation between features and outputs

non linearity makes life difficult. Best class of function easy to solve: Basis Function regression

Basis Function Regression

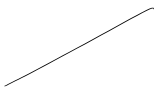
$$y = f(x) = \sum_{k}^{K} w_k \boxed{b_k(x)}$$

Basis function regression

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


— use sin


— use polynomial


— use linear

Two basis functions :

① polynomials

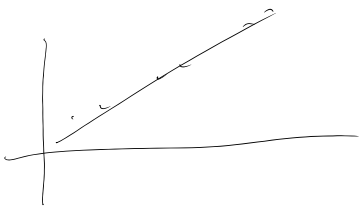
$b_k(x) = x^k$ ← x^k is monomial because it has only 1 order

Weighted sum of monomial is polynomial.

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

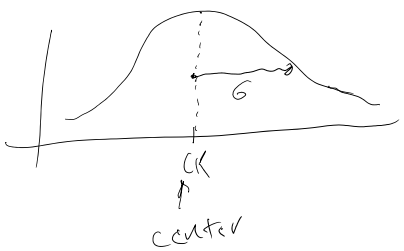
1st assignment - fit polynomial function to data.

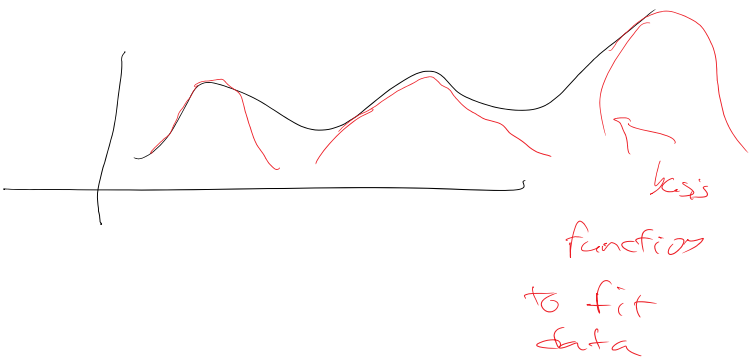
eg.



Radio Basis Function : (RBF)

$b_k(x) = e^{-\frac{(x-c_k)^2}{\sigma_k^2}}$ ← looks similar to normal





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

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

Estimation

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

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

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

$$= \begin{bmatrix} b_1(x_1) & b_2(x_1) & \cdots & b_K(x_1) \\ b_1(x_2) & b_2(x_2) & \cdots & b_K(x_2) \\ \vdots & & & \\ b_1(x_N) & \cdots & \cdots & b_K(x_N) \end{bmatrix}$$

← feature set for x_i, basis function using each row

↑ all the inputs vary

$$E(\bar{w}) = \| \bar{y} - B\bar{w} \|^2$$

objective $= (\bar{y} - B\bar{w})^T (\bar{y} - B\bar{w})$

$=$

How do we find optimal w?

Take gradient of E

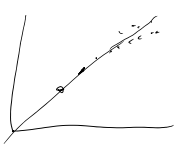
$$\nabla E = \frac{\partial E}{\partial w} = 0$$

$$\Rightarrow \bar{w}^* = (B^T B)^{-1} B^T \bar{y}$$

(using sequential inverse)

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 41 basis function, its better than 20 basis functions.

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

$$= \underbrace{\|\bar{y} - B\bar{w}\|^2}_{\text{data term}} + \underbrace{\lambda}_{\substack{\text{regularization} \\ \text{parameter} \\ \text{controls} \\ \text{balance}}} \underbrace{\|\bar{w}\|^2}_{\text{smoothness term}} \leftarrow \begin{array}{l}\text{extra terms} \\ \text{to penalize } w\text{'s} \\ \text{that are big} \\ (\lambda \sum w^2)\end{array}$$

$$y = f(x) = w_1 x + w_2 x^2 + w_3 \ldots$$

$$\frac{\partial f}{\partial x} = w_1 + 2w_2 x + \ldots$$

$$\frac{\partial f}{\partial^2 x} = 0 + 2w_2 + \ldots$$

$$E(\bar{w}) = \|\bar{y} - B\bar{w}\|^2 + \lambda \|\bar{w}\|^2$$

$$= (\bar{y} - B\bar{w})^T (y - B\bar{w}) + \lambda w^T \bar{w}$$

$$= \bar{w}^T B^T B\bar{w} + \lambda \bar{w}^T \bar{w} - 2\bar{w}^T B^T \bar{y} + \bar{y}^T \bar{y}$$

$$= \bar{w}^T (B^T B + \lambda I)\bar{w} - 2\bar{w}^T B^T y + y^T y$$

$$E = 0$$

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

$$\therefore \bar{w}^* = \underbrace{B^T B + (\lambda I)^{-1}}_{\leftarrow \text{inverse}} B^T \bar{y}$$

Ridge regression

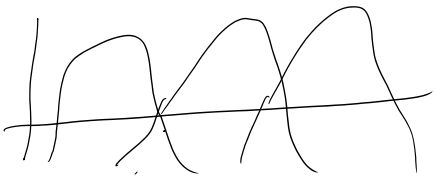
form of bias regression using regularisation parameter to encourage your model to be smooth.

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 linear combinations

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

Non-parametric model ex/

$$\{(\vec{x}_i, y_i)\}_{i=1}^{N}$$

K-nearest neighbour Regression (k-NN Regression)

Given a test point x get the nearest points to the $\vec{x}$ and average them.

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

← set of neighbours Indices into training data given point x

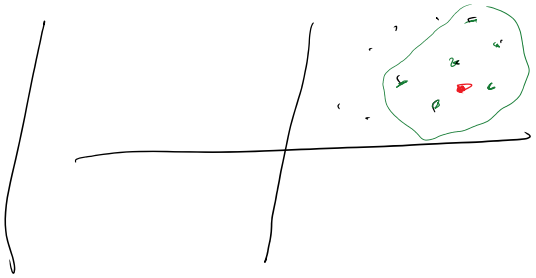
we don't usually choose $k=1$ because data is noisy

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 set

- training point

- nearest neighbor point, average them

Estimation Theory, Bay's optimal form of data fitting and prediction

Uncertainty

Prob: assign beliefs to events without observing them.

A: die1 shows 3

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

B: die2 shows 1

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

C: dice sum to 8, 2 dice

$$P(C) = \frac{|A|}{|B|} = \frac{6}{6^2} = \frac{5}{36}$$

(5,3), (3,5), (2,6), (6,2), (4,4) } = |A|

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

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

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

1 $P(A,B) = P(A)P(B)$

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

$$p(A) = \sum_i p(A, B_i)$$ ← joint

↑ marginal.

heteroschatisty: some data points have more noise than others.

Bayes Rule

$P(M \mid D)$ ⟶ A condition on the data

"Posterior distribution" – probality distribution of the model after you seen the data

M: model, D: Data

=> allows you to say which models are conditioned to my data.

likelihood: replikerlihood you will see the data with the given data

Bayes rule says $P(M \mid D) = \frac{P(D \mid M) P(M)}{P(D)}$ ← prior – eg flipping coin prior belief is $\frac{1}{2}$

evidence

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

$$p(M \mid D)$$

Estimation: estimating single model from data

"best model"

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

A model that maximizes

$p(D \mid M)$ $p(M)$ $p(M \mid D) \cdot p(D)$

A model that maximizes

$$\theta_{MAP} = \arg\max_{\theta} p(\theta \mid D) = \arg\max_{\theta} p(D \mid \theta) p(\theta)$$

(θ: parameter of model)

specific instance of the model class which maximize posterior distribution

2) ML (Maximum likelihood) Estimation

$$\theta_{ML} = \arg\max_{\theta} p(D \mid \theta)$$

Difference between map and likelihood is we are not using prior beliefs.

Note: equal if uniform distribution.

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

$c_1 \ldots c_n$ - outcome

flip 1: N times.

θ : $p(H)$ equivalently $p(c=H) = \theta$, $p(c=T) = 1-\theta$

(θ: parameter)

prior

$p(\theta) = 1$

$\theta \sim U(0,1)$

Assume I

$$p(c_{1:N} \mid \theta) = \prod_{i=1}^{N} p(c_i \mid \theta)$$

($p(c_{1:N} \mid \theta)$: likelihood)

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

estimation:

θ^* (|) ( )

estimation:

$$\theta^* = \arg\max_\theta p(C_{1:N}|\theta)p(\theta)$$

$$= \arg\max_\theta \log\left(p(C_{1:N}|\theta)p(\theta)\right)$$

$$= \arg\min_\theta \left(-\log p(C_{1:N}|\theta)p(\theta)\right)$$

} use these interchangable

$N = 1000$

# of heads = 750

then

# of tails = 250 , assume no coin landed on side

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

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

$$p(C_{1:5}|\theta) = p(C_1|\theta)p(C_2|\theta)p(C_3|\theta)\ldots$$

$$= \theta\,\theta\,(1-\theta)\,\theta\,(1-\theta)$$

$$= \theta^3(1-\theta)^2$$

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

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

minimize neg log posterior

$$= -\log(K) - 750\log(\theta) - 250\log(1-\theta)$$, by properties of log

Minimize it: # take derivative, set it = 0

$$\frac{d}{d\theta}\left(p(\theta \mid c_{1:1000})\right)$$

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

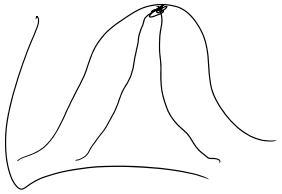
$$\frac{750}{\theta} = \frac{250}{1-\theta}$$

$$750(1-\theta) = 250(\theta)$$

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

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

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



← this only picks peak value. Map estimate does not categorize uncertainty

$$E[\theta] = \int \theta\, p(\theta \mid c_{1:1000})\, d\theta$$

$$= \int \theta\, k\, \theta^{750}(1-\theta)^{250} \times 1 \; d\theta \qquad \text{(1 is prior)}$$

$$\vdots$$

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

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

Regression

← d-dimensional wrigglers: treat as r.v

$\mu \in \mathbb{R}$

Regression

$y = \bar{w}^T \bar{x} + \varepsilon$ ← model

- y: scalar
- $\bar{x}$: d-dimensional set of measurements
- $\bar{w}$: d-dimensional weights; treat as r.v
- ε: source of noise

$y \in \mathbb{R}$
$\bar{x} \in \mathbb{R}^d$
$\bar{w} \in \mathbb{R}^d$
$\varepsilon \sim N(0, \sigma^2)$ — uncertainty in measurements

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

variance: σ^2

$y \sim$ variable that is gaussian

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

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

$p(\bar{w})$ — dimension is 0 because 0 distance & different weights

assume $\bar{w} \sim N(0, \alpha \bar{\bar{I}})$ (mean, variance)

Isotropic: prob distribution is the same direction

$$p(\bar{w}) = \prod_{i=1}^{d} p(w_i)$$

$$= \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi\alpha}} e^{-\frac{1}{2\alpha} w_i^2}$$

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

Hand in written work in the drop box
IC 4000 – Drawer 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

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

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

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

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

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

Covariance matrix which is Isotropic with a variance of α

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

Posterior

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

$$p(\bar{w}|D) = \frac{p(y_{1:N}|\bar{x}_{1:N},\bar{w})p(\bar{w})}{p(y_{1:N}|x_{1:N})}$$ assume $\bar{w}$ is $\perp$ of

assume noise is $\perp$

Independence $y_i = \bar{x}_i^T\bar{w} + \varepsilon_i$ ε is noise
$y_j = \bar{x}_j^T\bar{w} + \varepsilon_j$

$P(\varepsilon_i, \varepsilon_j) = p(\varepsilon_i)p(\varepsilon_j)$

$\Rightarrow p(y_i, y_j | x_i, x_j, \bar{w}) = p(y_i|x_i,\bar{w})p(y_j|x_j,\bar{w})$

$$p(\bar{w}|D) = \frac{\prod_{i=1}^N p(y_i|\bar{x}_i,\bar{w})p(\bar{w})}{c}$$

$$p(y_{1:N}|\bar{x}_{1:N},\bar{w}) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{N}{2}} e^{\frac{-1}{2\sigma^2}\sum_{i=1}^N (y_i - x_i^T\bar{w})^2}$$

e raised to quadratic multiplied by e raised to the quadratic is sum of quadratic

$$P(\bar{w}|D) = c'\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{N}{2}} e^{-\frac{1}{2\sigma^2}\sum(y_i - x_i^T\bar{w})^2} \left(\frac{1}{2\pi\alpha}\right)^{\frac{d}{2}} e^{-\frac{1}{2\alpha}\bar{w}^T\bar{w}}$$ take -log

log of product is sum of log

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

$$= -\log c + \frac{N}{2}\log(2\pi\sigma^2) + \frac{d}{2}\log(2\pi\alpha) + \frac{1}{2\sigma^2}\sum_{i=1}^N (y_i - x_i^T\bar{w})^2 + \frac{1}{2\alpha}\bar{w}^T\bar{w}$$ disappear

$-\log p(w|x)$

$= -\log c + \frac{N}{2}\log(2\pi\sigma^2) + \frac{d}{2}\log(2\pi\alpha) + \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - x_i^T w)^2 + \frac{1}{2\alpha}w^T w$, disappear because it doesn't depend on w.

$-\log p(w|D) = k + \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - x_i^T w)^2 + \frac{1}{2\alpha}w^T w$

Take derivative and set it to zero

multiply by a constant

$= k' + \sum_{i=1}^{N}(y_i - x_i^T w)^2 + \frac{2\sigma^2}{2\alpha}w^T w$

$\lambda = \frac{2\sigma^2}{2\alpha}$, form of ridge regression

What is a gaussian?

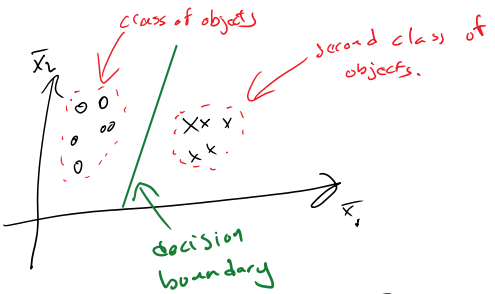
What is a prior?

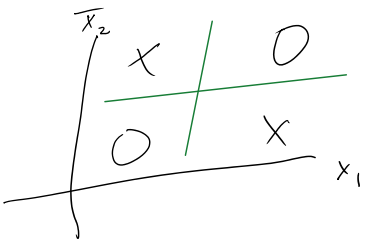
Classification

$y = f(x)$

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

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





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

Complexities are based on the model.

linear seperable: A linear function can seperate the data.

Classification by regression:

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

Find the best seperating hyperplane

weights $w \rightarrow w^* = \arg\min_w \sum_{i=1}^{N}(y_i - x_i^T w)^2$ ← squared error

$f(x) = w^{*T}x$

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

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

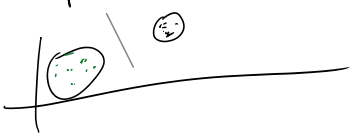
$w^T(x) > 0$, above hyperplane

$w^T(x) < 0$ / below the hyperplane

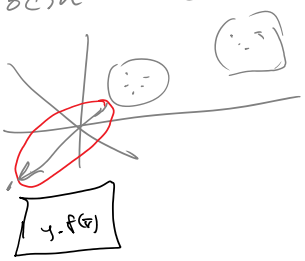
$$\text{sign}(z) = \begin{cases} -1 & z \leq 0 \\ 1 & z > 0 \end{cases}$$

This works well in only certain situation

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



Doesn't work well for



since loss function is huge

0-1 loss

$L_{0-1}(y, f(\vec{x}))$

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

problem? hard to optimize

Suppose $f(\vec{x}) = -2$

$y = 1$

$f(x) \times (y) = -2$ ← error.

penalize you for confident and right and confident & wrong.

K-NN Classifier

Find the k nearest neighbours to test inputs (with indicies in $N_k(\vec{x})$)

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

$w(x_i) = e^{\frac{-1}{2\sigma^2}\|\vec{x} - \vec{x}_i\|}$

Question on midterm how do K-NN classifier work.

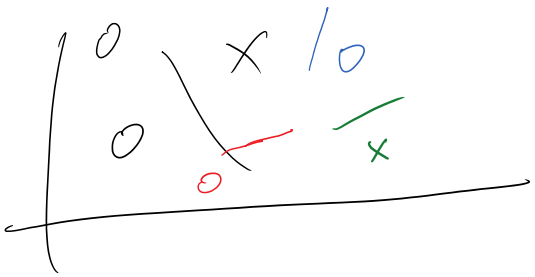
Bad: search, lots of point than you have to search all the points.

This is non parametric methods, no fixed parameter. Algorithm depends on the amount of data.

parametric model: # of parameter doesn't change.

non parametric won't do feature learning.

what does decision boundry look like



, Assume K = 1

Decision boundy can be complex and can overfit.

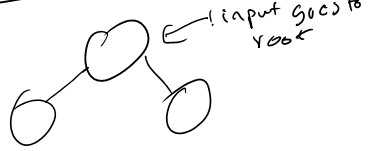
larger value of K (# of nearest neighbour) => blurring out decision boundrie

Decision Tree <= go to, often used in practice

Stump → Tree → forest

Decision tree can be used for regression or classification.

Consider Binary tree



M internal nodes — split nodes

M+1 leaf nodes

$$T_j(x) = \begin{cases} -1, & \text{take left branch} \\ 0, & \text{take right branch} \end{cases}$$

at internal node j

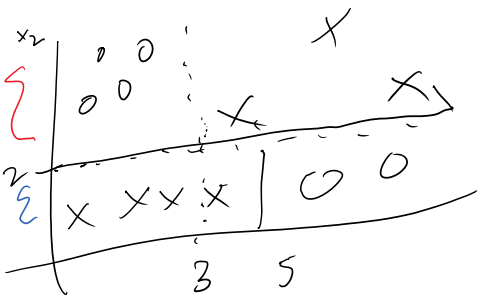
$t_j(x): \mathbb{R}^d \to \{-1, 1\}$

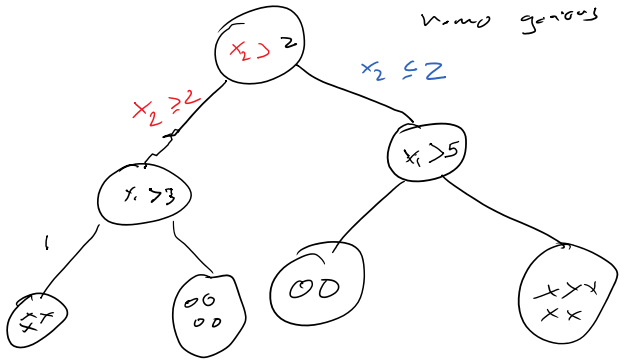
1. input goes to root of tree
2. gonna ask a feature eg $x \geq 2$ or $2 \leq x$

then go to the new node.

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

goal of split: separate class of data. split it for node it most grows





Once you get to leaf node, take a vote.

# of regions grows exponentially depending on depth of the tree.

(1) leaves of tree define

$p(c|\text{node } j) = \frac{N_{jc}}{N_j}$ } fraction of data points that reach that node that are class c.

N_j = # points reach node j

N_{jc} = # points @ node j with class c

Midterm covers everything up to today.

October 20th 5:00-6:00.

Decision Tree

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

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

take the left branch

take right branch.

leaf nodes @ node j with

N_j items prob(y=c | node j)

leaf node j, with N_j items

$N_{j,c}$ with class c

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

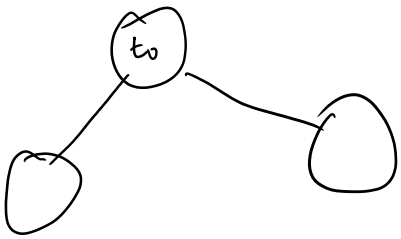
Decision Trees

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

split function

Given M-items $\{\bar{x}_i\} = D_j$ ← data at node j

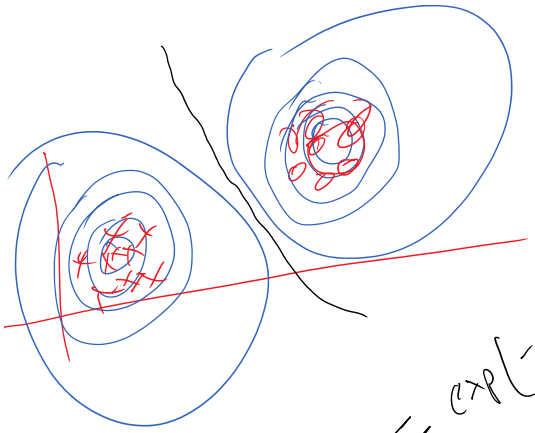
(to)



what will the split function look like?

Assume univariate split

↳ split function that only depe



$$\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}(y - w^T x)^2\right)$$

$$\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N \exp\left(\frac{-1}{2\sigma^2}\sum_{i=1}^{N}(y_i - w^T x_i)^2\right)$$

$$\|y - w^T x\|^2$$

$$(y - w^T x)^T (y - wx)$$

Only 3 assignments

Assignment is getting extended !
↳ Monday or Tuesday

Midterm marks within next few days.

Next week in tutorial go over questions & solutions.

K nearest neighbour: cons - store all data
- classifying all data

Decision Trees: many competitions in ML are won by random forest classifiers

Random Forest: avg of decision trees

Class Conditional model:
naive Bayes: assumption features on class is ⊥

logistic Regression:

5 Types of classifier

$$p(y|x) = k\,p(x|y)p(y)$$

$p(y|x)$: solving this thing is discriminative model

$p(x,y)$: generative model

If you know the class you can generate data from it.

discriminative model: give me measurement, give you back classes. Eg Decision tree of height, width, colour, texture etc..

10 measurements and run decision tree algorithm. Assume works with 2 measurements, therefore the other 8 measurements is unused

Generative model does more work and solving a harder problem than you need to

Discriminative models are usually better.

logistic regression, assume $p(y=1|x) = \frac{1}{1+e^{-w^T x}}$

presuppose parametric forms.

Gaussian class conditioning - generative model
Naive Bayes - generative model.

How to choose a good model!

Simple way: take training set to break it up into training set and validation set.

Pick a validation set because suppose train data is like test data. Take 80% and 20%. 20% is a validation set and train only on 80%

how to choose validation set: At random.

Sample without replacement from random.

Problem with validation set techniques
lack of data. 1000 training data
−200 validation
800

Problem with validation set techniques

lack of data. 1000 training data
- 200 validation
800 training left

less to train on. Weak when you don't have enough data.

When lack of data:

Validation: Try and cheese split 80,20 60,40 etc... You need enough for validation.

N-fold Cross Validation

Take data, rather than hold 20 out for validation do 5 random samples of 20. Break into partitions of 5 subsets. Build 5 models and test it on 20%. Average over 5 samples.

How big is N? $N \uparrow$ when data set $\downarrow$

limit $\Rightarrow$ LOOCV

↳ leave one out cross validation

Avg errors and that will tell you how well your model is

Let $\hat{y}_i$ be the prediction of model trained on $N-1$ pts (i.e. w/o i^{th})

$$LOOCV = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

LS regression

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

(w^*: compute optimal weights)

$$\hat{y} = Xw^* = \underbrace{X(X^TX)^{-1}X^T}_{H}\bar{y} = H\bar{y}$$

($\hat{y}$: predictions; H: some matrix H times y.)

$$LOOCV = \frac{1}{N}\sum_i MSE_i$$

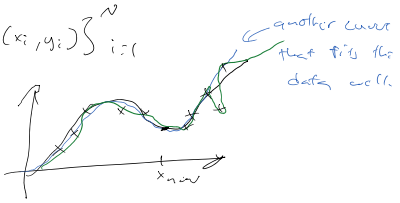
$$= \frac{1}{N}\sum_i \left(\frac{y_i - \hat{y}_i}{1-h_i}\right)^2$$

h_i is i^{th} diagonal element of H, leverage of i^{th} point.

Bayesian Method

Predict y, given $D = \{(x_i, y_i)\}_{i=1}^{N}$

$y = f(x; w)$ w_{ML}, w_{MAP}



$$p(w|D) = \frac{p(D|w)p(w)}{p(D)}$$

* Find entire distribution $p(\bar{w}|D)$
* find $p(y_{new}|x_{new}, D)$

Don't never really care about w but more y.

$$p(y|\bar{x}, D) = \int p(y, \bar{w}|\bar{x}, D)\,dw$$

$$p(y|x, D) = \int p(y, \bar{w}|\bar{x}, D)\,dw$$

$$= \int p(y|\bar{w}, \bar{x}, D)\,p(\bar{w}|\bar{x}, D)\,dw$$

1) $y = f(x, \bar{w})+n$ ← close to data; uncertainty

2) $p(\bar{w}|D)$ ← far from data

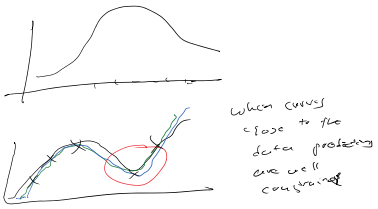
$$p(y|x,D) = \int p(y,\bar{w}|\bar{x},D)\,d\bar{w}$$
$$= \int p(y|\bar{w},\bar{x},D)\,p(\bar{w}|\bar{x},D)\,d\bar{w}$$

1) uncertainty

2) $p(\bar{w}|D)$ ← fun from data

↑ uncertainty

what you get for Bayesian is a whole probability distribution over y.



when curves close to the data producing one well constrained

Bayesian Regression / Gaussian process

$y = \bar{w}^T \bar{b}(x) + n$ ← random noise

$n \sim N(0, \sigma^2)$

$y \in \mathbb{R}$
$x \in \mathbb{R}$
$\bar{b}(x) \in \mathbb{R}^d$

$$p(y_{1:N}|x_{1:N},w) = \prod_{i=1}^{N} p(y_i|x_i,\bar{w})$$ ← statistical independence

Assume a prior, $\bar{w} \sim N(0, \frac{1}{\alpha} I)$ ← variance of each diagonal is $\frac{1}{\alpha}$

$$p(\bar{w}|D) = \frac{\text{likelihood} \cdot \text{prior}}{\text{probability of data}}$$

$$= \frac{\prod_{i=1}^{N} p(y_i|x_i,\bar{w})\,p(\bar{w})}{p(D)}$$

$$-\log p(\bar{w}|D) = -\sum_{i=1}^{N} \log p(y_i|x_i,\bar{w}) - \log p(\bar{w}) + C$$

$$y_i|x_i,w \sim N(\bar{b}(x_i)^T\bar{w}, \sigma^2) = \frac{1}{2\sigma^2}\sum\left(y_i - \bar{b}(x_i)^T\bar{w}\right)^2 + \frac{\alpha}{2}\|\bar{w}\|^2 + C'$$

(variance; mean)

$$= \frac{1}{2\sigma^2}\|\bar{y} - B\bar{w}\|^2 + \frac{\alpha}{2}\|w\|^2 + C'$$

$$\bar{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \qquad B = \left[\bar{b}(x_1)\ \ b(x_2)\ \ \cdots\ b(x_n)\right]^T$$

$$= \frac{1}{2\sigma^2}(\bar{y} - B\bar{w})^T(\bar{y} - B\bar{w}) + \frac{\alpha}{2}(w)^T(w) + C'$$

$$= \frac{1}{2\sigma^2}(\bar{y}^T - \bar{w}^T B^T)(\bar{y} - B\bar{w}) + \frac{\alpha}{2}(w)^T(w) + C'$$

$$= \frac{1}{2\sigma^2}\left(y^T y - \bar{y}^T B\bar{w} - \bar{w}^T B^T y + \bar{w}^T B^T B\bar{w}\right) + \frac{\alpha}{2}w^T w + C'$$

$$= \frac{1}{2\sigma^2}\left(y^T y - \bar{y}^T B\bar{w} - \bar{w}^T B^T y + \bar{w}^T B^T B\bar{w}\right) + \frac{\alpha}{2}w^T w + C'$$

- start trans with

$2\sigma^2$

$$= \frac{1}{2\sigma^2}\left(y^T y - 2\bar{y}^T B\bar{w} + \bar{w}^T B^T B\bar{w}\right) + \frac{\alpha}{2}\bar{w}^T\bar{w} + c'$$

$$= \frac{1}{2}\left(\bar{w}^T\left(\frac{B^T B}{\sigma^2} + \alpha I\right)\bar{w}\right) + c'$$

$$= \frac{1}{2}\bar{w}^T\left(\frac{B^T B}{\sigma^2} + \alpha I\right)\bar{w} - \bar{y}^T\frac{B\bar{w}}{\sigma^2} + c''$$ / complete square

$$-\log p(w|D) = \frac{1}{2}(\bar{w}-\hat{w})^T K^{-1}(w-\hat{w})$$ / where

$$\hat{w} = K B^T\bar{y}/\sigma^2$$

$$K = \left(\frac{B^T B}{\sigma^2} + \alpha\right)$$

once you see this form log prob quadratic $\Rightarrow w \sim$ gaussian

where $\hat{w}$ is mean

K is covariance

$$p(y_{new}|x_{new}, D) = \int \underbrace{p(y_{new}|x_{new}\bar{w})}_{\sim N(\bar{b}(x_{new})^T\bar{w}, \sigma^2)} \overbrace{p(\bar{w}|D)}^{N(\hat{w},K)}$$

$$y_{new}|x_{new}, D \sim N(\hat{w}^T\bar{b}(x_{new}), \sigma^2 + \bar{b}^T K\bar{b})$$

$\bar{b}(x_{new})$

collect terms with
w, put the rest
in the constant

the

~ mean

$\int^{-1}$ ~ covariance