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 ≈ 36%
- Quiz after assignment

- midterm: 15%
- final: 49%

piazza site is set up for this class.

3 main types of learning

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

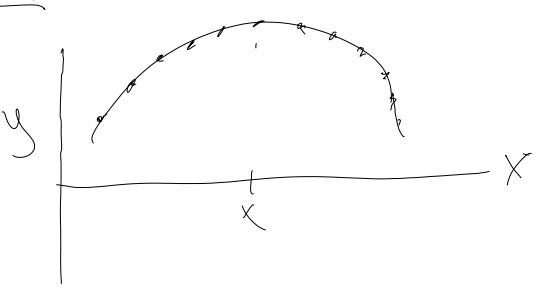
JPEG is a format of dimensionality reduction

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

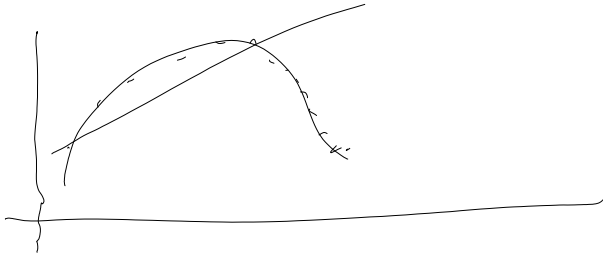


A family of function used to tune parameters to make a prediction.

eg.

Good model:

Person, should I fit a line? or curve?



not good model
over fitting

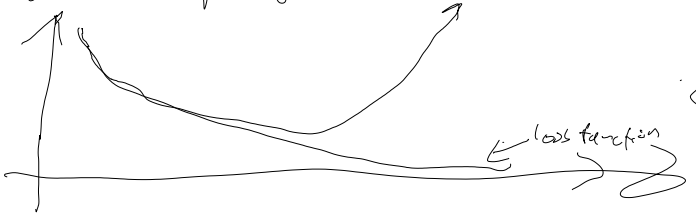
underfitting: errors in training data is too large

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

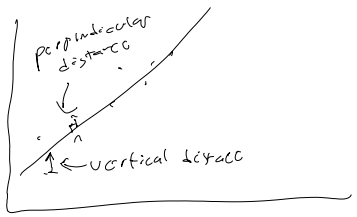
Loss function

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



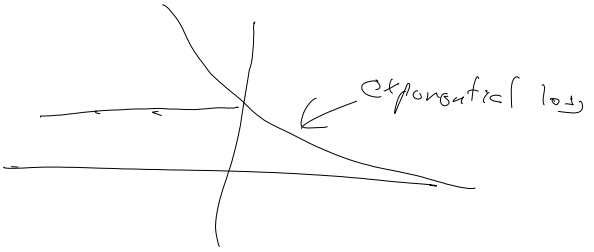
Ex Bin prob $y \in \{-1, 1\}$ $\text{sign}(yf(x))$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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: $\vec{x} \in \mathbb{R}^d$, $\vec{x} = \begin{bmatrix} x_1 \\ \vdots \end{bmatrix}$

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)

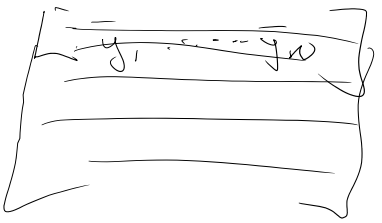
$= \sum_{i=1}^{N} \sum_{j=1}^{K} (y_{ij} - \tilde{w}_j^T \tilde{x}_i)^2$

loss and error are interchangable in this

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

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

y — describes columns
y' - describes rows



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

output for jth column matrix weights

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

, sum of K + least squared problems

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

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

Frobenius norm

normal vector form now applied to matrices

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

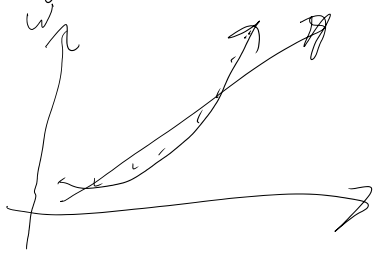
Two norm

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

Non linear regression

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

height vs weight



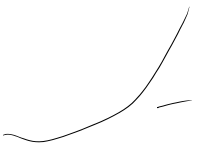
Think relation between feature and outputs.

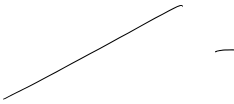
non linearity makes life difficult. But class of function easy to solve: Basis Function regression

Basis Function Regression

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

 — use sin

 — use poly nomial

 — use linear

Two basis functions :

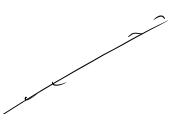
① polynomials

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

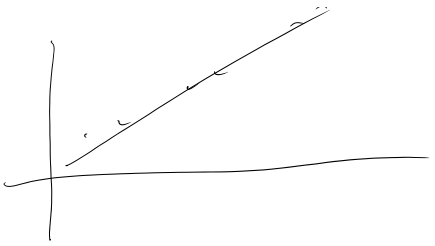
Weighted sum of monomial is polynomial.

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

1st assignment - fit polynomial function to data.

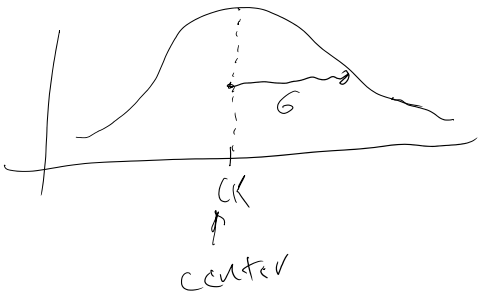
eg. 

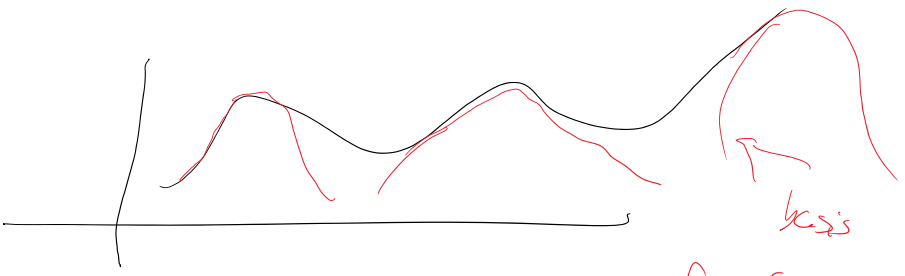
eg.



Radio Basis Function : (RBF)

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





basis function to fit data

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

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

Estimation

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

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

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

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

← 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 $\bar{w}$?

Take gradient of E

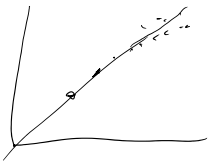
$$\nabla E = \frac{\partial E}{\partial \bar{w}} = 0$$

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

(least squared inverse)

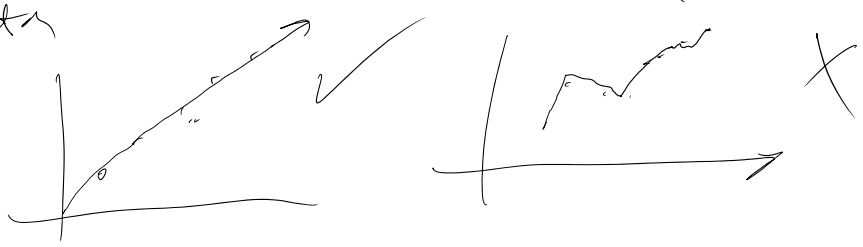
Reasons for overfitting

- 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 4 basis function, its better than 20 bases functions.

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

$$= \underbrace{\|\bar{y} - B\bar{w}\|^2}_{\text{data term}} + \underbrace{\lambda}_{\text{regularization parameter } L_2 \text{ controls balance}} \|\bar{w}\|^2 \leftarrow \text{extra terms to penalize } w\text{'s that are big } (\sum w^2)$$

smoothness term

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

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

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

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

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

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

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

$$E = 0$$

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

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

Ridge regression

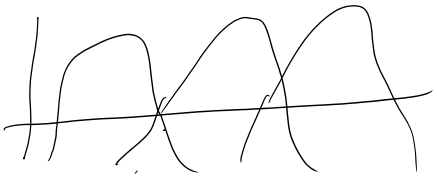
form of bias regression using regularization parameter to encourage your model to be smooth.

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 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

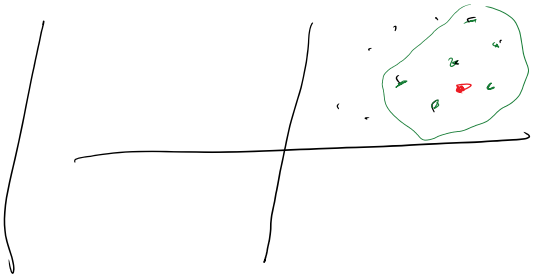
we don't usually choose $k=1$ because data is noisy

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, assign them

Estimation Theory, Bayes 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

model | Data

=> allows you to say which model 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

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

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

estimation:

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

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

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

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

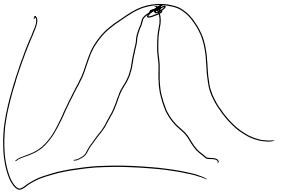
$$\frac{750}{\theta} = \frac{250}{1-\theta}$$

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

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

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

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



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

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

$$= \int \theta\, k\, \theta^{750}(1-\theta)^{250} \times 1 \; d\theta \quad \leftarrow \text{Prior}$$

$$\vdots$$

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

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

Regression

← d-dimensional wrigglers: threat any r.v

$\mid u \in \mathbb{R}$

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 ~ 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 10 because 5 biases & 5 different weights

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

Isotropic: prob distribution is the same direction

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

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

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