

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
 $y \in \{1, \dots, C\}$ e.g. 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.

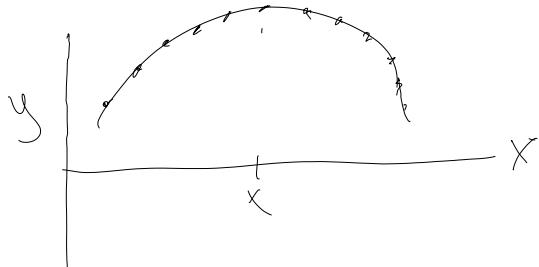
2) feature selection / Dimensionality Reduction

finding low dimensional features of high dimensional data in compression.

JPEG is a form of dimensionality reduction

e.g. this is an unlikely image in natural. But this image is highly probable.

Model

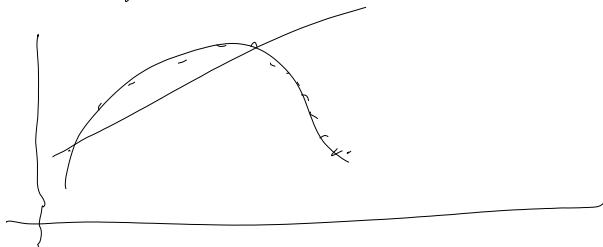


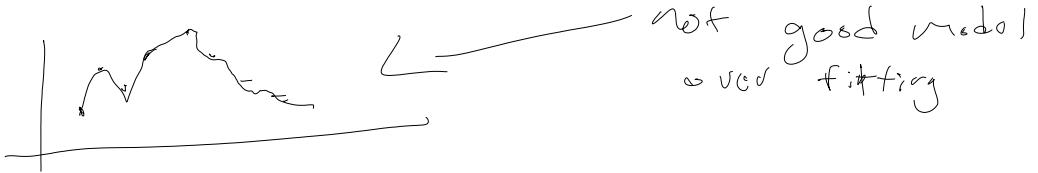
e.g.

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



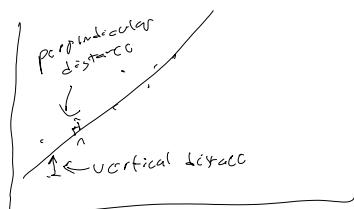
Loss function

measures and quantifies performance on data set

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

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

loss funct ex/



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

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

1 // -1 .. 1

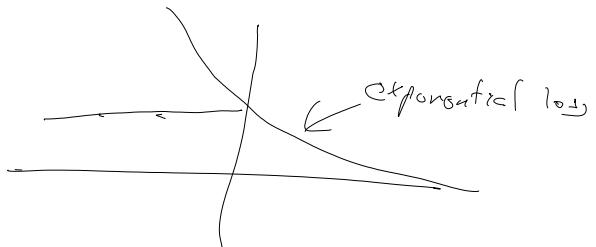
$\forall x \in \mathbb{R}^m$ given $y \in C^{-1}, 1)$



$$\operatorname{sgn}(y f(x))$$

$$\frac{1}{2}((1 - \operatorname{sgn}(y f(x)))^2)$$

loss function, # of curves you created



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

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

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

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

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

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

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

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

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

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

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

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

$$x = \frac{1}{N} \sum_i x_i$$

objectivo

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

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

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

$$\frac{\partial E}{\partial w} = -2 \sum_i ((y_i - \bar{y}) - w(x_i - \bar{x})) (x_i - \bar{x}) \left/ \begin{array}{l} \frac{\partial}{\partial w_i} \sum f(x_i, w)^2 \\ = \sum_i f(x_i, w) \frac{\partial}{\partial w_i} f(x_i, w) \end{array} \right.$$

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

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

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

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

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

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

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

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

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

\downarrow error

objective

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

$$\tilde{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \quad \tilde{X} = \begin{bmatrix} \tilde{x}_1^\top & \vdots & 1 \\ \tilde{x}_2^\top & \vdots & 1 \\ \vdots & \ddots & 1 \end{bmatrix} \quad N \text{ rows}$$

$d+1$ columns

$$\begin{aligned} E(\tilde{w}) &= \|\tilde{y} - \tilde{X}\tilde{w}\|_2^2 \quad \rightarrow \begin{cases} \text{two normed square} \\ \sqrt{\sum \text{of the elements}} \end{cases} \\ &= (\tilde{y} - \tilde{X}\tilde{w})^\top (\tilde{y} - \tilde{X}\tilde{w}) \\ &= \tilde{y}^\top \tilde{y} - 2\tilde{w} \end{aligned}$$

$\|v\|_2^2 = \sum v_j^2 = \tilde{v}^\top v$

Input: $\vec{x} \in \mathbb{R}^d$, $\vec{x} = (x)$

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

Note: Assign is out

Start if
already !

e.g. predict your grade using previous grades.

e.g. predict expected weight of child

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.

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

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

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

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

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

jth element of ith training sample

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

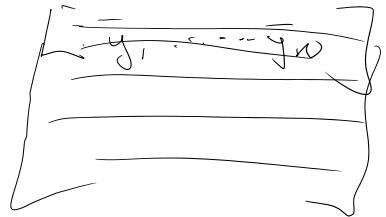
$$= \sum_{i=1}^N \sum_{j=1}^k (y_{ij} - \tilde{w}_j^T \vec{x}_i)^2$$

loss and error are interchangeable in this case

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



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

output
 vector
 matrix
 weights

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

, shows $k+1$ least squares problems

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

Frobenius norm

Frobenius norm

norm (vector first norm applied to matrices)

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

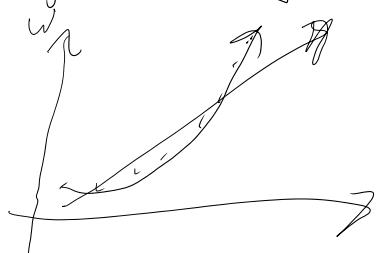
Two norms

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

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. Best class of function easy to solve: Basis Function regression

Basis Function Regression

$$y = f(x) = \sum_{n=1}^K w_n [b_n(x)]$$

— use sin

— use linear

— use poly nomial

Two basis functions

① polynomials

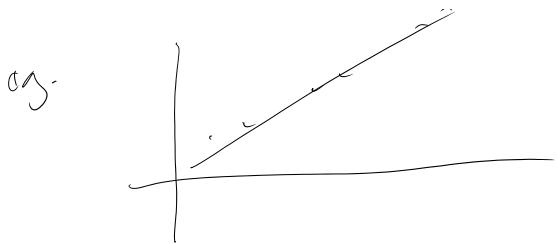
$b_n(x) = x^n \leftarrow x^n$ is monomial because it has only 1 order

Weighted sum of monomial is polynomial.

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

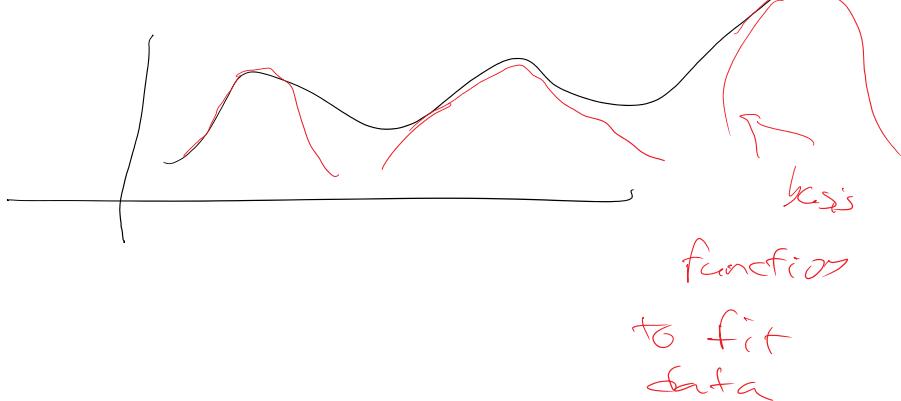
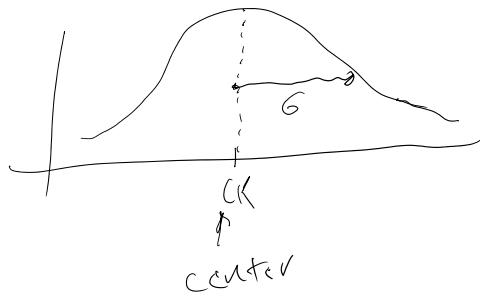
1st assignment - fit polynomial function to data.





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

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

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

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

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

↓

all the inputs vary

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

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

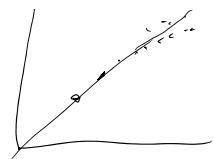
Take gradient of E

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

↑
Cross +
scaled
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 functions, it's better than 20 basis functions.

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

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

regulation
parameter
L2 controls
balance

smoothness term

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

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

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

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

$$\cancel{E} = 0$$

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

$$\therefore \bar{w}^* = B^\top B + \lambda I^{-1} B^\top \bar{y}$$

Ridge regression

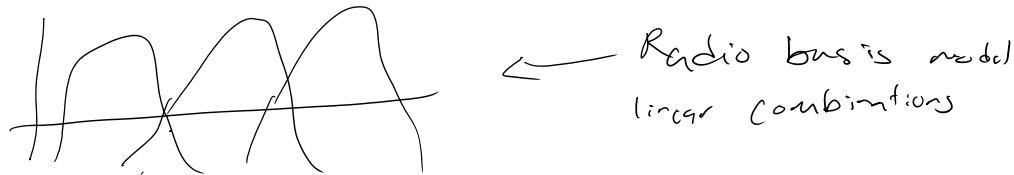
form of bias regression using regularization parameter to encourage your model to be smooth.

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 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

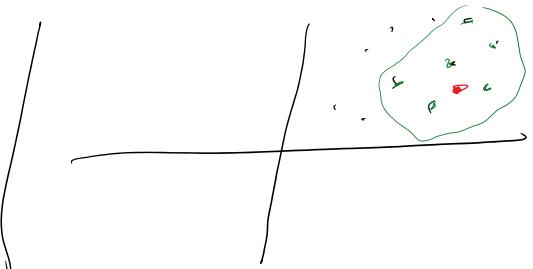
We don't usually choose $k=1$ because data is noisy

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 set

- training point

• - nearest neighbor
point, average
from

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

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

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

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

$$P(B \cap C) = 0$$

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

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

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

$$P(A) = \sum_i P(A, B_i) \leftarrow \text{Joint marginal.}$$

hodroschmidtii: some data points have more noise than others.



Bayes Rule

$P(M | D)$ → A condition on the factor
 ↑
 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.

Likelihood: replicability you will see the data with the given data

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

prior - e.g. flipping coin
prior belief is $\frac{1}{2}$

posterior

Inference: In ML, computing prob distⁿ 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 (modo)

A model that maximizes

$$m = \omega \cdot \text{color}(n) \text{color}(n)$$

A model that maximizes

$$\Theta_{MAP} = \underset{\theta}{\text{argmax}} p(\theta | D) = \underset{\theta}{\text{argmax}} p(D|\theta)p(\theta)$$

parameter
of model

specific instance of the model class which
maximize posterior distribution

2) ML (Maximum likelihood) Estimation

$$\Theta_{ML} = \underset{\theta}{\text{argmax}} p(D|\theta)$$

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.

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

parameter

prior

$$\begin{aligned} p(\theta) &= 1 \\ \theta &\sim U(0, 1) \end{aligned}$$

assume \perp

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

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

estimation:

map

$c_1 \sim 1 \sim 1 \sim 1$

estimation:

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

$$N = 1000$$

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

then

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

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

$$c_{1:5} = \text{HTHTT}$$

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

$$p(\theta | c_{1:1000}) = k \underbrace{p(c_{1:1000} | \theta)}_{\theta^{750} (1-\theta)^{250}} \underbrace{p(\theta)}_1$$

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

minimize neg log posterior

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

Minimize it: # take derivative, set it = 0

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

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

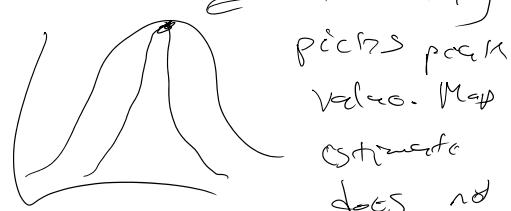
$$\frac{750}{\theta} = \frac{250}{1-\theta}$$

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

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

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

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



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

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

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

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

Regression

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

$/ u \in \mathbb{R}$

Regression

$$y = \bar{w}^T \bar{x} + \varepsilon \quad \leftarrow \text{model}$$

d-dimensional weights: treat as r.v.

↑ scalar

d-dimensional set of measurements

source of noise

$$\begin{cases} y \in \mathbb{R} \\ \bar{x} \in \mathbb{R}^d \\ \bar{w} \in \mathbb{R}^d \\ \varepsilon \sim N(0, \sigma^2) \\ \underbrace{\qquad\qquad}_{\text{uncertainty in}} \\ \text{measurements} \end{cases}$$

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

$$\text{variance: } \sigma^2$$

$y \sim$ variable that is gaussian

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

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

\downarrow dimension is
0 because of different weights

$$p(\bar{w}) \quad \downarrow \quad \text{mean variance}$$

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

Isotropic prob distribution is the same direction

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