L3

- Data: (x, y); $x \in \mathbb{R}^{d+1}$ $y \in \mathbb{R}$
- Goal: Find (linear) $f: \mathbb{R}^{d+1} \to \mathbb{R}$ such that $y \approx f(x; w)$ for data pair (x, y)
- LSS: Find w that minimises SSE (or SSR) $\to f(x; w) = x^T w$

$$L = \sum_{i=1}^{n} (y_i - x_i^T w)^2 = \|y - Xw\|^2 = (y - Xw)^T (y - Xw) \qquad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad X = \begin{bmatrix} - x_1 - \\ \vdots \\ - x_n - \end{bmatrix} \begin{matrix} \leftarrow \text{training} \\ \\ \\ \end{matrix} \quad \begin{matrix} \uparrow \\ \text{features} \\ \downarrow \end{matrix}$$

$$= y^T y - y^T X w - w^T X^T y - w^T X^T X w$$

$$= y^T y - 2 w^T X^T y - w^T X^T X w$$

as $(y^T X w)^T = w^T X^T y$ and both are scalars

$$\nabla_w L = \nabla_w y^T y - \nabla_w 2 w^T X^T y - \nabla_w w^T X^T X w$$

$$= 2 X^T X w - 2 X^T y = 0 \quad \Rightarrow w_{LS} = (X^T X)^{-1} X^T y$$

- LS has useful probabilistic interpretation :-
- Gaussian density in n dimensions, assuming specific form of covariance matrix and mean:

~~$\Sigma = \sigma I$~~ variance-covariance matrix and mean:

restrict $\mu = Xw$ and $\Sigma = \sigma^2 I$

Density: $p(y | \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(y - \mu)^T (y - \mu)\right)$ - y is an r.v. as not a vector here (hence σ)

- $\mu = Xw$ ← unknown
  - vector
  - known

- Substituting $\mu = Xw$ into MVG and solve for w using MLikelihood:

$$w_{ML} = \arg\max_w \ln\left(p(y | \mu = Xw, \sigma^2)\right)$$

$$= \arg\max_w \ln\left(\frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(y - \mu)^T (y - \mu)\right)\right)$$

$$= -\frac{n}{2} \ln\left(\frac{1}{2\pi\sigma^2}\right) - \frac{1}{2\sigma^2} \|y - Xw\|^2$$

LS: $\arg\min_w \|y - Xw\|^2$ ⇔ ML: $\arg\max_w -\frac{1}{2\sigma^2} \|y - Xw\|^2$

- w that minimises SSR
- w that maximises log-likelihood of data

Essentially, assumption is independent Gaussian noise assumption about error i.e. ~~$\epsilon_i = y_i$~~ $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$ $\forall i$ $\epsilon_i = y_i - x_i^T w$

- Probabilistic understanding of maximum likelihood and least squares:-

- Given $y \sim N(Xw, \sigma^2 I)$ (a modelling assumption about DGP under a frequentist paradigm of random data and fixed parameters, albeit unknown)

- $E[w_{ML}] = E[(X^TX)^{-1}X^Ty] = \int [(X^TX)^{-1}X^Ty]\, p(y|X,w)\, dy$ (nothing)

  - as y is an r.v. and expectation operator requires assumption of distribution you are taking over

$= (X^TX)^{-1}X^T E[y]$ as $(X^TX)^{-1}X^T$ does not contain r.v. y and is constant

$= (X^TX)^{-1}X^TXw$ as $E[y] = \mu = Xw$

$= (X^TX)^{-1}(X^TX)w$

$= w$ (unknown)

$\Rightarrow w_{ML}$ is an unbiased estimator of true population parameter w

$\Rightarrow$ least squares / maximum likelihood gives us 'the' parameter in expectation for the modelling assumption we've used.

Given 'ground truth' for w and inputs $\underline{X}$, we generate y according to modelling assumption $y \sim N(Xw, \sigma I)$ and find maximum likelihood estimator for w, expectation of estimator is equal to ground truth / population parameter which is fixed but unknown!

Even though that may be true in principle, in practice how close do we get? Need to look at ~~variance of y~~ (co)variance of y and w_{ML}:

$y = \begin{bmatrix} y_1 \\ y_n \end{bmatrix}$ is a random vector containing n random variables indexed by training set

~~for~~ Examine variance-covariance matrix:-

$Var(y) = E[(y - E[y])(y - E[y])^T]$

$= E[yy^T (y - E[y])(y^T - E[y]^T)]$

$= E[yy^T - yE[y]^T - E[y]y^T + E[y]E[y]^T]$

$= E[yy^T - y\mu^T - \mu y^T + \mu\mu^T]$ Recall $E[y] = \mu$

$= E[yy^T] - \mu^T E[y] - \mu E[y^T] + \mu\mu^T$ via linearity of expectation $E(X+Y) = E(X) + E(Y)$

$= E[yy^T] - 2\mu\mu^T + \mu\mu^T$

$= E[yy^T] - \mu\mu^T$

giving us $\boxed{\text{Var}(y) \text{ or } = \Sigma = E[yy^T] - \mu\mu^T}$ and $E[yy^T] = \Sigma + \mu\mu^T$

- Now turn to finding $\text{Var}(w_{ML})$

$$\text{Var}(w_{ML}) = E[(w_{ML} - E[w_{ML}])(w_{ML} - E[w_{ML}])^T]$$

$$= E[w_{ML} w_{ML}^T] - E[w_{ML}]E[w_{ML}]^T$$

make subs. $w_{ML} = (X^TX)^{-1}X^Ty$ in 1st bracket and $E[w_{ML}] = w$

$$= E[(X^TX)^{-1}X^Ty\,(X^TX)^{-1}X^Ty)^T] - ww^T$$

- $X \in \mathbb{R}^{n \times (d+1)} \Rightarrow (X^TX) \in \mathbb{R}^{(d+1)\times(d+1)} \Rightarrow (X^TX)$ is square

For square A: $(A^{-1})^T = (A^T)^{-1}$

$(X^TX)^T = X^TX$

$$= E[(X^TX)^{-1}X^Tyy^TX(X^TX)^{-1}] - ww^T$$

$$= (X^TX)^{-1}X^TE[yy^T]X(X^TX)^{-1} - ww^T$$

$E[yy^T] = \Sigma + \mu\mu^T$ and $\mu = Xw$, $\Sigma = \sigma^2 I$

$$= (X^TX)^{-1}X^T(\sigma^2 I + Xww^TX^T)X(X^TX)^{-1} - ww^T$$

multiplying out

$$= (X^TX)^{-1}X^T\sigma^2 I X(X^TX)^{-1} + (X^TX)^{-1}X^TXww^TX^TX(X^TX)^{-1} - ww^T$$

$$= \sigma^2(X^TX)^{-1}X^TX(X^TX)^{-1} + ww^T - ww^T$$

$$= \sigma^2(X^TX)^{-1}$$

$$\boxed{\text{Var}(w_{ML}) = \sigma^2(X^TX)^{-1}}$$

- Under $y \sim N(Xw, \sigma I)$ Gaussian assumption; $E[w_{ML}] = w$, $\text{Var}(w_{ML}) = \sigma^2(X^TX)^{-1}$
- Large values of $\text{Var}(w_{ML})$ in $\sigma^2(X^TX)^{-1}$ leads to values of w_{ML} which are sensitive to measured data y (often in co-variates X^TX)
- Ridge regression is a solution to this potentially high variance of w_{ML}.

Regularised least squares or ridge regression

- $\text{Var}(w_{ML})$ may be large (through σ^2 or $(X^TX)^{-1}$)
- we can constrain model parameters

$$w_{OPT} = \underset{w}{\text{argmin}} \|y - Xw\|^2 + \lambda g(w)$$

λ - regularisation parameter ($\lambda > 0$)

$g(w)$ - penalty function encouraging desired properties for parameter vector w

- Ridge regression is regularised LS with specific functional form as penalty function $g(w): \mathbb{R}^{d+1} \to \mathbb{R}$ as $\|w\|^2$ i.e. squared L^2 norm of regression coeff vector

$$w_{RR} = \underset{w}{\operatorname{argmin}} \|y - Xw\|^2 + \lambda \|w\|^2$$

penalises magnitude of w

- $g(w) = \|w\|^2$ penalises large values in w

Tradeoff controlled between 1st and 2nd terms controlled by λ

- Case $\lambda \to 0$: $w_{RR} \to w_{LS}$ $\quad \lambda \to \infty$ $w_{RR} \to \underline{0}$

- Ridge regression solution :-

$$\begin{aligned} L &= \|y - Xw\|^2 + \lambda \|w\|^2 \\ &= (y - Xw)^T (y - Xw) + \lambda w^T w \\ &= y^T y - y^T X w - w^T X^T y + w^T X^T X w + \lambda w^T w \\ &= y^T y - 2 w^T X^T y + w^T X^T X w + \lambda w^T w \end{aligned}$$

$$\underline{0} = \nabla_w L = \nabla_w y^T y - \nabla_w 2 w^T X^T y + \nabla_w w^T X^T X w + \nabla_w \lambda w^T w$$

$$\Rightarrow \quad 2X^T X w - 2X^T y + 2\lambda w = \underline{0}$$

$$\Rightarrow \quad (X^T X + \lambda I) w = X^T y$$

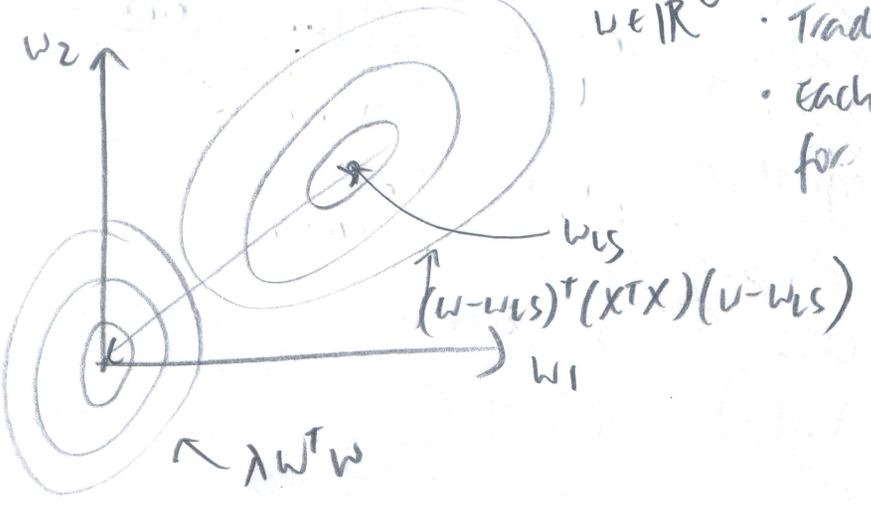
$$\Rightarrow \quad \boxed{w_{RR} = (X^T X + \lambda I)^{-1} X^T y}$$

Ridge regression geometry

$$\|y - Xw\|^2 + \lambda \|w\|^2 = (w - w_{LS})^T (X^T X)(w - w_{LS}) + \lambda w^T w + (\text{constant wrt } w)$$

$w \in \mathbb{R}^2$

- Tradeoff captured by λ
- Each level set has same for value for combinations of $\underline{w} = (w_1, w_2)$
- λ controls degree to which you move along lvl



- $(X^TX)(X^TX)^{-1} = VS^2V^TVS^{-2}V^{-T} = I$ check ✓
- $\therefore Var(w_{ML}) = \sigma^2(X^TX)^{-1} = \sigma^2 VS^{-2}V^T$
- Inverse or $(X^TX)^{-1}$ component of $Var(w_{ML})$ becomes large when
  i) Columns of X highly correlated (feature vectors) ⇒ ii) S_{ii} is very small for some i
- S_{ii} small ⇒ values of S^{-2} (diagonals) are very large ($\frac{1}{S_{ii}^2}$)

Ridge regression and least squares relation

- Symmetric matrices A, B; $(AB)^{-1} = B^{-1}A^{-1}$ · Find way to include w_{LS} in w_{RR} formula

$$w_{RR} = (\lambda I + X^TX)^{-1}X^Ty$$

$$= (\lambda I + X^TX)^{-1} I X^Ty \qquad I = (X^TX)(X^TX)^{-1}$$

$$= (\lambda I + X^TX)^{-1}(X^TX)(X^TX)^{-1}X^Ty \qquad (X^TX)^{-1}X^Ty = w_{LS}$$

$$= \left[(X^TX)\left(\lambda(X^TX)^{-1} + I\right)\right]^{-1}(X^TX)\,w_{LS} \qquad I = (X^TX)(X^TX)^{-1} \text{ and factor out } (X^TX)$$

$$= \left(\lambda(X^TX)^{-1} + I\right)^{-1}(X^TX)^{-1}(X^TX)\,w_{LS}$$

$$= \left[\lambda(X^TX)^{-1} + I\right]^{-1} w_{LS}$$

- $\|w_{RR}\|_2 \leq \|w_{LS}\|_2$ shrinks (as in shrinkage method towards 0)
- Using SVD in context of relation between w_{RR} and w_{LS}

$X = USV^T \rightarrow (X^TX)^{-1} = VS^{-2}V^T$

$$w_{RR} = \left(\lambda(X^TX)^{-1} + I\right)^{-1} w_{LS}$$

$$= \left(\lambda VS^{-2}V^T + I\right)^{-1} w_{LS}$$

$$= V\left(\lambda S^{-2} + I\right)^{-1} V^T w_{LS} \qquad \text{(2)}$$

$$:= VMV^T w_{LS}$$

- $(\lambda S^{-2} + I)$ is diagonal
- As V is square orthonormal
- and I and S^{-2} are diagonal

Letting $M = (\lambda S^{-2} + I)^{-1}$

- Preprocessing in ridge regression
- Ridge regression penalises dimension of w equally as $\lambda\|w\|^2 = \lambda w^T w$
- Scale of ~~x_i in certain feature d~~ d^{th} feature vector has to be adjusted

- subtract mean from y

$$y \leftarrow y - \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$x_{ij} \leftarrow \frac{(x_{ij} - \bar{x}_{\cdot j})}{\hat{\sigma}_j} \qquad \hat{\sigma}_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - x_{\cdot j}^2)}$$

- standardise dimensions of x_i
- scale of each j^{th} element of X will affect penalty
- subtract empirical/sample mean ca divide by sample s.d.

- no need for 1s dimension

- $w_{LS} = (X^TX)^{-1}X^Ty \quad w_{RR} = (\lambda I + X^TX)^{-1}X^Ty$

Analysing ridge regressions via SVD

- SVD is very useful for statistical analysis
- use SVD to compare w_{RR} and w_{LS}; and analyse mechanics of $Var(w_{ML})$ via SVD of $(X^TX)^{-1}$
- SVD: can write any $n \times d$ matrix X ($n \gg d$) as $X = USV^T$
  - U - $n \times d$, orthonormal in columns i.e. $U^TU = I$ (left sing vector)
  - S - $d \times d$ non-negative diagonal matrix $S_{ii} \geq 0$ and $S_{ij} = 0$ for $i \neq j$
  - V: $d \times d$ orthonormal i.e. $V^TV = VV^T = I$ (square) (right sing vector)

- orthogonal vectors $\Rightarrow x^Ty = 0$
- orthonormal also have $\|x\|_2 = 1$

$$X^TX = (USV^T)^T(USV^T) = VS^TU^TUSV^T = VS^TISV^T = VS^2V^T$$

$$XX^T = (USV^T)(VS^TU^T) = US^2U^T$$

$S^TS = S^2$ as S is ~~not~~ diagonal

- Assuming $S_{ii} \neq 0 \ \forall i$ (i.e. X is full-rank)
- Full rank $\Rightarrow \text{rank}(X) = \min(n,d)$
- S_{ii} known as singular values

- $(X^TX)^{-1} = (VS^2V^T)^{-1}$

$= (V^T)^{-1}S^{-2}V^{-1}$ (?)

$= (V^{-1})^{-1}S^{-2}V^{-1}$

$= VS^{-2}V^T$

~~$\forall \in \mathbb{R}^{d \times d}$~~ (non-singular/proper inverse)

$(V^{-1})^T = (V^T)^{-1} = V^{-T}$

~~$V, S \in \mathbb{R}^{d \times d}$~~

$(VS)^{-1} = S^{-1}V^{-1}$

As $V^TV = I$, $\boxed{V^T = V^{-1}}$ (orthonormal)

(as opposed to $\text{rank}(A_{k \times m}) \leq \min(n,d)$)

- column rank = row rank
- column rank of X is largest subset of columns of X that constitute linearly independent set.

~~$(X^TX)^{-1} = VS^{-2}V^T$~~

$\boxed{X = USV^T \rightarrow (X^TX)^{-1} = VS^{-2}V^T}$

Applying SVD in context of LLS $X = USV^T$, $X^T = VS^TU^T = VSU^T$ as $S^T = S$

$(X^TX)^{-1} = VS^2V^T$

$w_{LS} = (X^TX)^{-1}X^Ty = (VS^{-2}V^T)VSU^Ty = VS^{-1}U^Ty$; $\boxed{w_{LS} = VS^{-1}U^Ty}$

$w_{RR} = VMV^Tw_{LS} = VMV^TVS^{-1}U^Ty$

$= VMS^{-1}U^Ty$

$= V(\lambda S^{-2} + I)^{-1}S^{-1}U^Ty$

$= V\underbrace{(\lambda S^{-1} + S)^{-1}}_{S_\lambda^{-1}}U^Ty$

$M = (\lambda S^{-2} + I)^{-1}$ - diagonal matrix with $M_{ii} = \frac{s_{ii}^2}{\lambda + s_{ii}^2}$

$= VS_\lambda^{-1}U^Ty$, $S_\lambda^{-1} = \begin{bmatrix} \frac{s_{11}}{\lambda + s_{11}^2} & & 0 \\ & \ddots & \\ 0 & & \frac{s_{dd}}{\lambda + s_{dd}^2} \end{bmatrix}$ $d \times d$

λ puts floor on smallest value of denom.

S_λ^{-1} is a $d \times d$ diagonal matrix with diagonal $\frac{s_{11}}{\lambda + s_{11}}$

- when $\lambda = 0$, $s_{dd} \to 0$ $\Rightarrow \frac{s_{dd}}{\lambda + s_{dd}^2} \to \infty$

- Note that $\lambda = 0 \Rightarrow S_\lambda^{-1} = S^{-1}$ and hence $w_{RR} = w_{LS}$ - when $\lambda > 0$ $s_{dd} \to 0 \Rightarrow \frac{s_{dd}}{\lambda + s_{dd}^2} \to 0$

Ridge regression as a special case of least squares (objective fn)

~~Define~~ - Assume standardisation

- Define $\hat{y} \triangleq \hat{X}w$ ← by i) appending d zeros to column vector y

ii) A $d \times d$ diagonal matrix containing $\sqrt{\lambda}$ to covariates X

$$d \text{ rows} \left\{ \begin{bmatrix} y \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right. \triangleq \left. \begin{matrix} \\ d \end{matrix} \right\} \begin{bmatrix} - & X & - \\ \sqrt{\lambda} & & 0 \\ & \ddots & \\ 0 & & \sqrt{\lambda} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$

- Finding w_{LS} for above regression $\Rightarrow w_{RR}$ of original problem
- calculate $(\hat{y} - \hat{X}w)^T(\hat{y} - Xw)$ in 2 parts $\Rightarrow$

$$(\hat{y} - \hat{X}w)^T(\hat{y} - \hat{X}w) = (y - Xw)^T(y - Xw) + (\sqrt{\lambda}w)^T(\sqrt{\lambda}w)$$

$$= \|y - Xw\|^2 + \lambda\|w\|^2$$

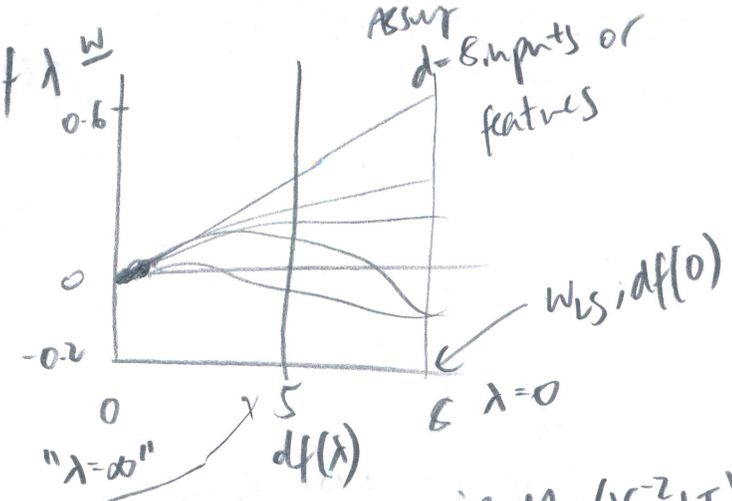
(?) - show this

selecting λ

- Define degrees of freedom as function of λ

$$df(\lambda) = \text{trace}\left[X(X^TX+\lambda I)^{-1}X^T\right]$$

$$= \sum_{i=1}^{d} \frac{s_{ii}}{\lambda + s_{ii}^2}$$



Absur
d - 8 inputs or features

$\lambda \to \infty \Rightarrow df(\lambda) \to 0$

$\lambda \to 0 \Rightarrow df(\lambda) \to d$

- Plot shows $\underline{w}$ (regression coeffts vector) as function of $df(\lambda)$

i.e $M = (\lambda S^{-2} + I)^{-1}$

~~trace~~ trace (M) = ~~trace~~

- trace - sum of diagonals

value of λ that gives $df(\lambda) = 5$

- How do parameter estimates of $\underline{w}$ change as λ and hence $df(\lambda)$ vary

4 Regression with/without regularisation

- Remember standardisation here :-
- $(x_1, y_1), \dots (x_n, y_n)$ where $x \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ (?)
- standardisation ⇒ each dimension of x is zero mean unit variance
  y is zero mean
- model often defined as $y \hat{} f(x; w)$ and focus $f(x; w) = x^T w$ (linear)
- often minimise objective: $\mathcal{L} = \sum_{i=1}^{n} (y_i - x_i^T w)^2 + \lambda w^T w \Leftrightarrow \mathcal{L} = \|y - Xw\|^2 + \lambda \|w\|^2$
  $\lambda = 0 \Rightarrow$ LS $\lambda > 0 \Rightarrow$ regularisation
- Bias-variance for linear regression
- Hypothesise a generative model $y \sim N(Xw, \sigma^2 I)$ and a true (but unknown) underlying value for parameter vector $\underline{w}$
- $w_{LS} = (X^T X)^{-1} X^T y$ $E[w_{LS}] = w$ $\text{Var}[w_{LS}] = \sigma^2 (X^T X)^{-1}$
  ~~potentially high~~
- $w_{RR} = (\lambda I + X^T X)^{-1} X^T y$
  - checked $E[w_{RR}]$ OK
  - $\text{Var}[w_{RR}]$?
- $$E[w_{RR}] = (\lambda I + X^T X)^{-1} X^T X w$$
  $$\text{Var}[w_{RR}] = \sigma^2 Z (X^T X)^{-1} Z^T$$
- Above expressions ($E(w_{RR})$ and $\text{Var}(w_{RR})$) converge to $E[w_{LS}]$ and $E[w_{RR}]$ as $\lambda \to 0$
- ~~Characterise~~ how well we can hope to learn w mathematically in cases where our model assumption is correct; and generalise to new data
- Let (x_0, y_0) be future data where x_0 is available, but not y_0
- LS prediction: $y_0 = x_0^T w_{LS}$ RR prediction: $y_0 = x_0^T w_{RR}$
- Process:
  - Imagine I know X and x_0 and assume an unknown but fixed parameter $\underline{w}$
  - Generate $y \sim N(Xw, \sigma^2 I)$ (responses for test set) and approximate w with $\hat{w} = w_{LS}$ or w_{RR}
  - Predict $y_0 \sim N(x_0^T w, \sigma^2)$ using $y_0 \hat{} x_0^T \hat{w}$
- What is expected squared error of prediction :-

- Expected squared error of prediction of y_0, given x_0 and previous data $X = (x_1, \dots, x_n)$

$$E[(y_0 - x_0^T \hat{w})^2 \mid X, x_0] = \int_{\mathbb{R}} \int_{\mathbb{R}^n} (y_0 - x_0^T \hat{w})^2 \, p(y \mid X, w) \, p(y_0 \mid x_0, w) \, dy \, dy_0$$

- Note $\hat{w} = w_{LS}$ or w_{RR} - ridge regression or least squares estimate of w
- distributions on y and y_0 are Gaussian family with true but unknown w
  $y \sim N(Xw, \sigma^2 I)$ and $y_0 \sim N(x_0^T w, \sigma^2)$
- condition on knowing x_0 ~~and~~, $x_1, \dots, x_n$ (X)
- $\hat{w}$ and y_0 are treated as random

- $\int_{\mathbb{R}^n} [\dots] \, dy$ - integrate over distribution on y given data X and true but unknown vector w. For particular w, we calculate distribution on y $p(y \mid X, w)$
- $\int_{\mathbb{R}} [\dots] \, dy_0$ - integrate over distribution on new response y_0 given true but unknown vector w and new data x_0

- Recall $\hat{w} = w_{LS}$ and w_{RR} are both dependent on vector y and X
- We see how EPE changes as function of data X, x_0 and true but unknown parameter w.

conditioning on x_0 and X

$$E[(y_0 - x_0^T \hat{w})^2] = E[(y_0 - x_0^T \hat{w})(y_0 - x_0^T \hat{w})]$$

$$= E[(y_0^2 - y_0 x_0^T w - x_0^T \hat{w} y_0 + x_0^T \hat{w} \hat{w}^T x_0)] \quad \text{via linearity of } E[\cdot]$$

$$= E[y_0^2] - E[y_0 x_0^T w] - E[x_0^T \hat{w} y_0] + E[x_0^T \hat{w} \hat{w}^T x_0] \quad \text{conditioning on } x_0$$

$$= E[y_0^2] - 2 x_0^T E[y_0] E[\hat{w}] + x_0^T E[\hat{w} \hat{w}^T] x_0 \quad (?) \otimes E[y_0 w] = E[y_0] E[w]$$

as y_0 and w are independent

- Analogous* to $E[yy^T] = Var[y] + E[y] E[y]^T$ - recall y is a vector populated with random variables
- $E[\hat{w} w^T] = Var[\hat{w}] + E[\hat{w}] E[\hat{w}]^T$
- $E[y_0^2] = \sigma^2 + (x_0^T w)^2$ - as $y_0 \sim N(x_0^T w, \sigma^2)$

$$E[(y_0 - x_0^T \hat{w})^2] = \sigma^2 + (x_0^T w)^2 - 2 x_0^T (x_0^T w) E[\hat{w}] + x_0^T \left(Var[\hat{w}] + E[\hat{w}] E[\hat{w}]^T \right) x_0$$

$$= \sigma^2 + (x_0^T w)^2 - 2 (x_0^T w) x_0^T E[\hat{w}] + x_0^T E[\hat{w}] E[\hat{w}]^T x_0 + x_0^T Var[\hat{w}] x_0$$

quadratic

note $x_0^T = x_0$ (scalar)

$$= \sigma^2 + (x_0^T w)^2 -$$

• $w^T = w$?

$$\Rightarrow \boxed{E[(y_0 - x_0^T \hat{w})^2 | X, x_0] = \underbrace{\sigma^2}_{\text{noise (1)}} + \underbrace{x_0^T (w - E[\hat{w}])(w - E[\hat{w}])^T) x_0}_{\text{squared bias (2)}} + \underbrace{x_0^T \mathrm{Var}[\hat{w}] x_0}_{\text{variance (3)}}}$$

if $y \sim N(Xw, \sigma^2 I)$ and $y_0 \sim N(x_0^T w, \sigma^2)$ i.e. generate y and y_0 (w is true but unknown)
and if we approximate w with $\hat{w}$ according to algorithm

- Bias-variance trade-off for generalisation error (linear regression)
- Generalisation error comprises i) measurement noise ii) bias (how close solution is on average) iii) variance (sensitivity of solution to data)
- ii) and iii) correspond to uncertainty about $\hat{w}$ (estimator) and its relation to "w".
- can make substitutions for $E[\hat{w}]$ and $\mathrm{Var}[\hat{w}]$ for RR and LS; in case of RR unknown w remains and we don't know it; can account by giving candidate values of w.

Bias-variance tradeoff (general) (see 7.10.)

- more generally, for $y = f(x; w) + \epsilon$ $E(\epsilon) = 0$ $\mathrm{Var}(\epsilon) = \sigma^2$
- Approximate f by minimising loss fn: $\hat{f} = \underset{f}{\mathrm{argmin}}\, L_f$
- Apply $\hat{f}$ to new data $y_0 \approx \hat{f}(x_0) \equiv \hat{f}_0$
- Integrating out (y, X, y_0, x_0):

$$E[(y_0 - \hat{f}_0)^2] = E[y_0^2] - 2E[y_0 \hat{f}_0] + E[\hat{f}_0^2]$$

✓ check

$$= \sigma^2 + \hat{f}_0^2 - 2 f_0 E[\hat{f}_0] + E[\hat{f}_0]^2 + \mathrm{Var}[\hat{f}_0]$$

$$= \sigma^2 + (f_0 - E[\hat{f}_0])^2 + \mathrm{Var}(\hat{f}_0)$$

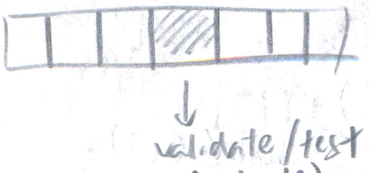
noise — squared bias — variance

- f usually can't be calculated

Cross validation

- easier way to evaluate
- K-fold X-validation :-
  - i) Split data into K groups
  - ii) Learn model on (K-1) groups, predict held out Kth group
  - iii) Carry out K times, holding out each group once
  - iv) Evaluate performance (according to a measure) using summarise set of predictions

- In context of regularisation λ, above sequence run for values of λ with best performance chosen.
- split dataset → test
  → training



↓ validate/test

Bayes Rule (discrete)

- Regularised least squares and ridge regression can be seen as a way of imposing prior beliefs on suitable values of $\underline{w}$; Bayesian statistics helps here
- For 2 (un)related events can specify (marginal) probabilities, conditional probability and joint probabilities

1. $P(A,B) = P(A|B)P(B) = P(B|A)$ (relation between joint, conditional and marginal probabilities)

2/3. $P(A) = \sum_b P(A,B=b)$ and $P(B) = \sum_a P(A=a,B)$ (marginal probability mass function is sum of joint probabilities over all outcomes in question)

- $P(A|B) = \frac{P(A,B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$ For Bayes theorem
- Best visualised with tree diagrams in frequentist probability

- $P(A|B) = \frac{P(B|A)P(A)}{\sum_a P(A=a,B)} = \frac{P(B|A)P(A)}{\sum_a P(B|A=a)P(A=a)}$ and $P(B|A) = \frac{P(A|B)P(A)}{\sum_b P(A|B=b)P(B=b)}$

- These are generally conditional probabilities until we impose some kind of ordering on events A and B e.g. probability of B given A happens/is observed

- $P(B|A) = \frac{P(A|B)P(B)}{P(A)}$ — prior
  posterior, likelihood, marginal

"posterior is proportional to the prior times the likelihood"

- Frequentist interpretation:- probability measures 'proportion of outcomes' ~~trials~~
- Bayesian — " — :- probability measures 'degrees of belief'
- In context of continuous r.vs. and parameter-data/hypothesis-evidence context:-
- For a continuous model parameter θ and data X:-

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)\,d\theta} = \frac{p(X|\theta)p(\theta)}{p(X)}$$

- $p(X|\theta)$ is likelihood of data given model parameters and is a generative distribution on data given model (we define)
- $p(\theta)$ is a prior distribution of model parameters which we define
- $p(X)$ is integral of numerator over all possible values of model parameters; marginal probability of data
- Given $p(X|\theta)$ and $p(\theta)$ likelihood and prior, we can (in principle) calculate $p(\theta|X)$ posterior

coin bias example

- coin with bias π towards heads ad $H=1$ $T=0$
- flip coin n times ad get sequence of n numbers $(x_1, \ldots, x_n)$ (observed)
- Assume independence of flips $\Rightarrow$

$$p(x_1, \ldots, x_n | \pi) = \prod_{i=1}^{n} p(x_i|\pi) = \prod_{i=1}^{n} \pi^{x_i} (1-\pi)^{1-x_i}$$

$X = (x_1, \ldots, x_n)$

$\theta = \pi$

- choose prior $p(\pi)$ as Beta distribution
- $p(\pi) = \text{Beta}(\pi|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \pi^{a-1}(1-\pi)^{b-1}$
- what is posterior distribution of parameter π given $x_1, \ldots, x_n$?
- Via Bayes: $p(\pi|x_1, \ldots, x_n) = \frac{p(x_1, \ldots, x_n|\pi)\, p(\pi)}{\int_0^1 p(x_1, \ldots, x_n|\pi)\, p(\pi)\, d\pi}$
- denominator normalises numerator, does not depend on π (i.e. calculated)
- $p(\pi|x) \propto p(x|\pi) p(\pi)$

TRICK (multiply): $p(\pi|x_1, \ldots, x_n) \propto \left[\prod_{i=1}^{n} \pi^{x_i}(1-\pi)^{1-x_i}\right]\left[\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \pi^{a-1}(1-\pi)^{b-1}\right]$

constant cancels out due to $\propto$

$$\propto \pi^{\sum_{i=1}^{n} x_i + a - 1} (1-\pi)^{\sum_{i=1}^{n}(1-x_i)+b-1}$$

- $p(\pi|x_1, \ldots, x_n) = \text{Beta}\left(\sum_{i=1}^{n} x_i + a, \sum_{i=1}^{n}(1-x_i) + b\right)$

maximum a posteriori

- with (x_i, y_i) and linear model $y_i \hat{} x_i^T w$, $w_{LS} = \arg\min_w (y-Xw)^T(y-Xw) = \|y-Xw\|^2$ was equivalent to w_{ML} when $y \sim N(Xw, \sigma^2 I)$
- probabilistic connection with w_{RR}
- For $w \in \mathbb{R}^d$, likelihood model $p(y|w)$ is $y \sim N(Xw, \sigma^2 I)$ as in earlier discussion
- Assume a Gaussian prior for w i.e. $w \sim N(0, \lambda^{-1} I)$ a d-dimensional multivariate Gaussian
- Recall for an d-dimensional multi-variate Gaussian with covariance $\Sigma = \sigma^2 I$:

$$p(y|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}(y-\mu)^T(y-\mu)\right) \quad \text{subs. } \sigma^2 = \lambda^{-1},\ \mu = 0,\ y = w$$

$$= \frac{1}{(2\pi\lambda^{-1})^{d/2}} \exp\left(-\frac{1}{2\lambda^{-1}} w^T w\right)$$

$$p(\pi) = \left(\frac{\lambda}{2\pi}\right)^{d/2} \exp\left(-\frac{\lambda}{2} w^T w\right)$$

- Find w that satisfies likelihood (data) and our prior beliefs/conditions about w

Maximum a posteriori estimation

- MAP seeks most probable value of $\underline{w}$ under the posterior

$$w_{MAP} = \underset{w}{\operatorname{argmax}} \ln p(w|y,X)$$

$$= \underset{w}{\operatorname{argmax}} \ln \frac{p(y|w,X)\,p(w)}{p(y|X)}$$

slight twist on this additionally conditioning on X

$$= \underset{w}{\operatorname{argmax}} \ln p(y|w,X) + \ln p(w) + \underbrace{\ln p(y|X)}_{\text{not needed as does not depend on } w \text{ and because of argmax}}$$

$p(w|X) = p(w)$ via independence of w and X

- ML only focuses on likelihood (frequentist)
- ~~to~~ Normalising constant $\ln p(y|X)$ does not depend on w; focus on maxing first 2
- often don't know & $p(y|X)$ can't use $\ln(p(y|X))$

MAP estimation of ridge regression

- MAP maximises posterior under a zero-mean Gaussian prior assumption on $\underline{w}$

$$w_{MAP} = \underset{w}{\operatorname{argmax}} \ln p(y|w,X) + \ln p(w)$$

- Recall that $y \sim N(Xw, \sigma^2 I)$ and $w \sim N(0, \lambda^{-1} I)$

$$\Rightarrow p(y|w,X) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}(y-Xw)^T(y-Xw)\right)$$

$$p(w) = \left(\frac{\lambda}{2\pi}\right)^{d/2} \exp\left(-\frac{\lambda}{2} w^T w\right)$$

$$w_{MAP} = \underset{w}{\operatorname{argmax}} \ln\left[\frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}(y-Xw)^T(y-Xw)\right)\right] + \ln\left[\left(\frac{\lambda}{2\pi}\right)^{d/2} \exp\left(-\frac{\lambda}{2} w^T w\right)\right]$$

$$= \underset{w}{\operatorname{argmax}} \underbrace{-\frac{1}{2\sigma^2}(y-Xw)^T(y-Xw) - \frac{\lambda}{2} w^T w}_{= L} + \underbrace{\left(\frac{d}{2}\ln\left(\frac{\lambda}{2\pi}\right) - \frac{d}{2}\ln(2\pi\sigma^2)\right)}_{\text{constant}}$$

$$L = -\frac{1}{2\sigma^2}\left(y^Ty - y^TXw - w^TX^Ty + w^TX^TXw\right) - \frac{\lambda}{2} w^Tw$$

$$= -\frac{1}{2\sigma^2}\left(y^Ty - 2w^TX^Ty + w^TX^TXw\right) - \frac{\lambda}{2} w^Tw$$

$$= -\frac{1}{2}\sigma^2 y^Ty + \frac{1}{\sigma^2} w^TX^Ty - \frac{1}{2\sigma^2} w^TX^TXw - \frac{\lambda}{2} w^Tw$$

$$\nabla_w L = \frac{1}{\sigma^2} X^Ty - \frac{1}{\sigma^2} X^TXw - \lambda w = \underline{0}$$

$$\Rightarrow \lambda w + \frac{1}{\sigma^2} X^T X w = \frac{1}{\sigma^2} X^T y$$

$$\Rightarrow \left(\lambda I + \frac{1}{\sigma^2} X^T X\right) w = \frac{1}{\sigma^2} X^T y$$

$$\Rightarrow \left(\lambda \sigma^2 I + X^T X\right) w = X^T y$$

$$\therefore \boxed{w_{MAP} = \left(\lambda \sigma^2 I + X^T X\right)^{-1} X^T y} = w_{RR} \text{ with } \lambda_{new} = \lambda \sigma^2$$

- w_{LS} corresponds to w_{ML} under Gaussian likelihood $y \sim N(Xw, \sigma^2 I)$
- w_{RR} —"— w_{MAP} under Gaussian likelihood $y \sim N(Xw, \sigma^2 I)$ and Gaussian prior $w \sim N(0, \lambda^{-1} I)$

Ridge regression maximises posterior, least squares maximises likelihood

L5

Bayesian LR

$y \in \mathbb{R}^n$ $X \in \mathbb{R}^{n \times d}$ (design matrix); i^{th} row of y and X - (y_i, x_i)

- Bayesian setting: likelihood $y \sim N(Xw, \sigma^2 I)$ - how well observed data agrees with w
  prior $w \sim N(0, \lambda^{-1} I)$ - prior beliefs on w

unknown model variable $w \in \mathbb{R}^d$

- Bayesian LR → define prior on unknown parameter, learn its posterior, given data

MAP inference

- MAP inference returns maximum of joint likelihood

Joint likelihood: $p(y, w|X) = p(y|w, X)\, p(w|X)$

- $p(w|X) = p(w)$ as w independent of X
- $p(A, B) = p(A|B)\, p(B)$

$$w_{MAP} = \underset{w}{\operatorname{argmax}} \ln \underbrace{p(w|y, X)}_{\text{posterior}}$$

(likelihood, prior)

$$= \underset{w}{\operatorname{argmax}} \ln \left[\frac{p(y|w, X)\, p(w)}{p(y|X)} \right]$$

($p(y|X)$: normalising constant)

$$= \underset{w}{\operatorname{argmax}} \ln \left[\frac{p(y|w, X)\, p(w)}{\int_{\mathbb{R}^d} p(y|w = w_i, X)\, p(w = w_i)\, dw} \right]$$

$$= \underset{w}{\operatorname{argmax}} \ln(p(y|w, X)) + \ln(p(w)) + \ln \underbrace{\left[\int_{\mathbb{R}^d} \dots dw \right]}_{\text{omitted as i) constant ii) no } w \text{ and argmax. prop.}}$$

$$= \underset{w}{\operatorname{argmax}} \ln(p(y|w, X)) + \ln(p(w))$$

$$= \underset{w}{\operatorname{argmax}} -\frac{1}{2\sigma^2}(y - Xw)^T(y - Xw) - \frac{\lambda}{2} w^T w + \text{constant}$$ - see L4 (under G.A.)

- Maximising posterior of w ⇔ maximisation of joint likelihood
- $w_{MAP} = (\lambda\sigma^2 I + X^T X)^{-1} X^T y$ ⇔ w_{RR} (under G.A.)

Point estimates vs Bayesian inference

- w_{MAP}, w_{ML} are point estimates of model parameters (specific value for unknown param according to inference proc.)
- Find a specific value for w that maximises obj.
- ML: Only data model considered $p(y|w, X)$ - normally written $p(y; w)$
- MAP: Accounts for model prior $p(y, w|X) = p(y|w, X)\, p(w)$
- Bayesian inference characterises uncertainty about the values of w using Bayesrule and returns a distribution on w, not just point estimate.

Posterior calculation

- As w is a continuous r.v. in $\mathbb{R}^d$, the posterior distribution of w, given y and X:

$$p(w|y, X) = \frac{p(y|w, X)\, p(w)}{\int_{\mathbb{R}^d} p(y|w, X)\, p(w)\, dw}$$

- updated distribution on w through prior → likelihood → posterior
- Posterior is proportional to likelihood × prior

- update posterior $p(w|y,X)$

prior → likelihood → posterior

$p(w)$	$p(y\|w,X)$	$p(w\|y,X)$
$w \sim N(0, \lambda^{-1}I)$	$y \sim N(Xw, \sigma^2 I)$	calculated from last two terms

Bayesian inference (mLR)

- In our special case, we can update posterior $p(w|y,X)$ analytically

$$p(w|y,X) \propto p(y|w,X)\,p(w)$$

$$\propto \underbrace{\frac{1}{(2\pi\sigma^2)^{d/2}}} \exp\left(-\frac{1}{2\sigma^2}(y-Xw)^T(y-Xw)\right) \underbrace{\left(\frac{\lambda}{2\pi}\right)^{d/2}} \exp\left(-\frac{\lambda}{2}w^Tw\right)$$

- Eliminating terms which do not involve w (valid under $\propto$)

$$\propto \exp\left(-\frac{1}{2\sigma^2}(y-Xw)^T(y-Xw) - \frac{\lambda}{2}w^Tw\right)$$

$$\propto \exp\left(-\frac{1}{2\sigma^2}(y^Ty - y^TXw - w^TX^Ty + w^TX^TXw) - \frac{\lambda}{2}w^Tw\right)$$

$$\propto \exp\left(-\frac{\lambda}{2}w^Tw - \frac{1}{2\sigma^2}w^TX^TXw + \frac{1}{\sigma^2}w^TX^Ty\right)\exp\left(-\frac{1}{2\sigma^2}y^Ty\right)$$

$$p(w|y,X) \propto \exp\left(-\frac{1}{2}\left\{w^T(\lambda I + \sigma^{-2}X^TX)w - 2\sigma^{-2}w^TX^Ty\right\}\right) \propto$$

- Eliminating

- We have to normalise above i.e. divide by $\int_{R^d} p(y|w,X)\,p(w)\,dw$
- Instead, solve by 'inspection' rather than integrating RHS
- Notice in exponent: $w^T(\lambda I + \sigma^{-2}X^TX)w - 2\sigma^{-2}w^TX^Ty$ (quadratic in w; linear in w)

- $p(w|y,X)$ is Gaussian as i) multiply, divide by not w
  ii) gaussian has $(w-\mu)^T\Sigma^{-1}(w-\mu)$ in exponent
  iii) complete the square by adding terms not involving w

- General Gaussian for w:

$$p(w|\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\left(w^T\Sigma^{-1}w - 2w^T\Sigma^{-1}\mu + \mu^T\Sigma^{-1}\mu\right)\right)$$

$$= \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\left(w^T\Sigma^{-1}w - 2w^T\Sigma^{-1}\mu\right)\right)\exp\left(-\frac{1}{2}\mu^T\Sigma^{-1}\mu\right)$$

$$= \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}\exp\left(\frac{1}{2}\mu^T\Sigma^{-1}\mu\right)} \exp\left(-\frac{1}{2}\left(w^T\Sigma^{-1}w - 2w^T\Sigma^{-1}\mu\right)\right)$$

• Setting $\Sigma^{-1} = (\lambda I + \sigma^{-2} X^T X)$; $\Sigma^{-1}\mu = \frac{X^T y}{\sigma^2}$ and $z = (2\pi)^{d/2} |\Sigma|^{1/2} e^{\frac{1}{2}\mu^T \Sigma^{-1} \mu}$

(X) come back

We have $p(w|y,X) = \frac{1}{z} e^{-\frac{1}{2}(w^T(\lambda I + \sigma^{-2}X^T X)w - 2w^T X^T y \sigma^{-2})}$

Posterior distribution

• Posterior distribution of w:-

$$p(w|y,X) = N(w|\mu, \Sigma)$$

$$\text{with } \mu = (\lambda\sigma^2 I + X^T X)^{-1} X^T y \rightarrow w_{MAP} \text{ with } \lambda' = \lambda\sigma^2$$

$$\Sigma = (\lambda I + \sigma^{-2} X^T X)^{-1}$$

• $\mu = w_{MAP}$ with redefined $\lambda = \lambda\sigma^2$ (Bayes rule to calculate posterior of w ; $\mu = w_{MAP} = w_{RR}$)

• Σ captures uncertainty about w as $Var[w_{LS}]$ and $Var[w_{RR}]$, but now full probability distribution on w (a functional distribution that allows us to give densities and calculate probabilities)

uses of posterior distribution -

i) understanding parameter w

Q1) Is $w_i > 0$ or $w_i < 0$? Can we confidently say $w_i \neq 0$?

- marginal posterior $w_i \sim N(\mu_i, \Sigma_{ii})$ to characterise 'true' value of param.

Q2) How do w_i, w_j relate?

- Joint marginal posterior $\begin{bmatrix} w_i \\ w_j \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_i \\ \mu_j \end{bmatrix}, \begin{bmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{bmatrix}\right)$

ii) Predicting new data ★

Predicting new data - predictive distribution

• For a new pair (x_0, y_0) with x_0 measured, y_0 unknown

• $\hat{y}_0 = x_0^T w_{LS}$ or $\hat{y}_0 = x_0^T w_{RR}$: point estimates of w ⇒ point estimates of prediction $\hat{y}_0$ with no uncertainty

• With Bayes rule, we can make a probabilistic statement about y_0 using the posterior predictive distribution:

$$p(y_0|x_0, y, X) = \int_{\mathbb{R}^d} p(y_0, w|x_0, y, X)\, dw$$

(i) ; (ii)

$$= \int_{\mathbb{R}^d} p(y_0|w, x_0, y, X)\, p(w|x_0, y, X)\, dw$$

(iii) ; (iv)

i) Predictive distribution of y_0 or marginal likelihood of (unobserved) y_0 conditional on x_0, y, X

ii) Joint distribution of y_0 and w, conditioned on x_0, y, X

- Via $P(A \wedge B) = P(A|B)P(B)$
- Background x_0, y, X from conditioning to clarify

iii) Likelihood of (unobserved) y_0

iv) Prior on w (?) / Posterior

Key substitution: $p(y_0|w, x_0, y, X) = p(y_0|w, x_0)$ - y_0 is conditionally independent of y and X

via conditional independ assmts

likelihood

- x_0 and w give all info necessary to define distribution on $y_0 \sim N(x_0^T w, \sigma^2)$

$p(w|x_0, y, X) = p(w|y, X)$ - w is conditionally independ of x_0

- Knowing x_0 but not y_0 gives no extra info to update beyond y and x

Predictive distribution:

$$p(y_0|x_0, y, X) = \int_{\mathbb{R}^d} p(y_0|x_0, w)\, p(w|y, X)\, dw$$

(likelihood, posterior)

- intuitively, represent predictive distribution as marginal $p(y_0|x_0, y, X)$ over likelihood of new data given modelled variables $p(y_0|x_0, w)$ multiplied by posterior of those model variables given old data $p(w|y, X)$
- AND, we evaluate likelihood of a new y_0 for a particular w and observed x_0; we weight it by our current belief (~~Hard~~ prior?) about w given data (y, X)
- The integrate over all possible values of parameter w
- likelihood $p(y_0|x_0, w)$ is calculated with likelihood model $N(y_0|x_0^T w, \sigma^2)$ as $y \sim N(Xw, \sigma^2 I)$
- Posterior $p(w|y, X)$ is calculated using above as new prior model $N(w|0, \lambda^{-1})$ as $w \sim N(0, \lambda^{-1} I)$ and using Bayes rule (to find appropriate normalisation constant)
- posterior $p(w|y, X) = N(w|\mu, \Sigma)$ with $\mu = (\lambda\sigma^2 I + X^T X)^{-1} X^T y$ (w_{MAP} with λ'_{new})

$$\Sigma = (\lambda I + \sigma^{-2} X^T X)^{-1}$$

- under our assumptions, the predictive distribution can be calculated exactly as has Gaussian distribution:-

$$p(y_0|x_0, y, X) = N(y_0|\mu_0, \sigma_0^2)$$

$\mu_0 = x_0^T \mu$ - Expected value of y_0 $E[y_0]$ is MAP prediction as $\mu = x_0^T w_{MAP}$

$\sigma_0^2 = \sigma^2 + x_0^T \Sigma x_0$ - we can't quantify confidence in this prediction with variance σ_0^2.

Active learning

(iterative)

- Bayesian learning can be seen sequentially, rather than as batch
- Posterior after seeing old data becomes the prior for the next data
- In this case, the event to which linguistic terms 'posterior', 'prior' apply is the observation of new data (y_0, x_0)
- Have to be clear about temporal sequence
- Let (y, X) be 'old data'

  (y_0, x_0) be 'new data'
- Bayes Rule:- (giving clarity to lecture content)
- with 'old data' (y, X), calculate posterior $p(w|y, X)$

$$p(w|y, X) \propto p(y|w, X)\, p(w)$$

ACTION: observe (x_0, y_0) and calculate new posterior $p(w|y_0, x_0, y, X)$ including new data

$$p(w|y_0, x_0, y, X) \propto p(y_0|w, x_0, y, X)\, p(w_{new})$$

- $p(w_{new}) = p(w|y, X)$
- 'new prior on w after observing (x_0, y_0) is the 'old' posterior $p(w|y, X)$

$$p(w|y_0, x_0, y, X) \propto p(y_0|w, x_0)\, p(w|y, X)$$

$$p(w|y_0, x_0, y, X) \propto p(y_0|w, x_0)\, p(w|y, X)$$
i) ii) iii)

· $p(y_0|w, x_0, y, X) = p(y_0|w, x_0)$ via C.I. of y_0 from y, X : $y_0 \sim N(x_0^T w, \sigma^2)$

i) Full posterior of w given all data (y, X) and (y_0, x_0)
ii) Likelihood of new data (y_0) given given w
iii) New prior on w having observed (x_0, y_0) ; which is posterior calculated from old data (y, X)

· Prior → Posterior → Prior (lecture)

· updating distribution on w:

$t=0$: Prior → likelihood → Posterior

$t=1$: observe (x_0, y_0) → Prior → likelihood → Posterior

· sequential updating on distribution $p(w|y, X)$

· original $p(w|y, X) = N(w|\mu, \Sigma)$

with $\mu = (\lambda\sigma^2 I + X^T X)^{-1} X^T y$

$\Sigma = (\lambda I + \sigma^{-2} X^T X)^{-1}$

(*) clarity for representation of $(X^T X)$ and $(X^T y)$ given at end.

· observe (x_0, y_0)

· new $p(w|y_0, x_0, y, X) = N(w|\mu, \Sigma)$

$$\mu = \left(\lambda\sigma^2 I + \left(x_0 x_0^T + \sum_{i=1}^{n} x_i x_i^T\right)\right)^{-1}$$

$$\Sigma = \left(\lambda I + \sigma^{-2}\left(x_0 x_0^T + \sum_{i=1}^{n} x_i x_i^T\right)\right)^{-1}\left(x_0 y_0 + \sum_{i=1}^{n} x_i y_i\right)$$

· In contrast to entire batch :-

$$p(w|y_0, x_0, y, X) \propto p(y_0, y|w, X, x_0)\, p(w)$$
i) ii) iii)

i) posterior given all data
ii) likelihood of all data
iii) prior on w

· learning w and making predictions for new y_0 is a two-step procedure

- i) form predictive distribution $p(y_0|x_0, y, X)$ "measure y_0"
- ii) update posterior —"— $p(w|y, X, y_0, x_0)$ with rank 1 update

(*) see end of notes for a schematic diagram

Active learning Heuristic motivation

· How can we learn posterior / update it efficiently?
· Applies when we can choose the sequence of observations/measurements we have
· E.g. If we can pick which y_i to measure with knowledge of $D = \{x_1, \ldots, x_n\}$; can we come up with a way of sequentially picking y_i to measure to most efficiently update posterior of w
· Often when there is significant cost of measurement/observation of y_i.

Active learning Heuristic

- We have already measured dataset (y, X) and posterior $p(w|y, X)$
- Construct predictive distribution for remaining $x_0 \in D$ i.e. all x_0 in data set for we don't have corresponding response y_0

$$p(y_0|x_0, y, X) = N(y_0|\mu_0, \sigma_0^2)$$

$$\mu_0 = x_0^T \mu$$

- Posterior mean $\mu = (\lambda\sigma^2 I + X^T X)^{-1} X^T y$ (*)

$$\sigma_0^2 = \sigma^2 + x_0^T \Sigma x_0$$

- Posterior covariance $\Sigma = (\lambda I + \sigma^{-2} X^T X)^{-1}$

- For every/each x_0, we can calculate respective values $\mu_0 = x_0^T \mu$ and $\sigma_0^2 = \sigma^2 + x_0^T \Sigma x_0$
- σ_0^2 gives a measurement of how confident we are in our prediction
- For different x_0 we have different measurements of our confidence in predictions of μ_0

Heuristic process:-

1) Form predictive distribution $p(y_0|x_0, y, X)$ for all data $x_0 \in D$ that do not have measured response y_0
2) Pick the data point x_0 for which σ_0^2 is largest and measure y_0
3) Update the posterior distribution of w $p(w|y, X)$ using $y_{new} \leftarrow (y, y_0)$ and $X_{new} \leftarrow (X, x_0)$ i.e.
4.) Return to #1 using updated posterior

- In each iteration, choose to measure response that corresponds to a covariate vector that we are least certain about in prediction.

Active learning as differential entropy minimisation

- We find out what objective this heuristic is optimising
- For a continuous probability distribution $p(z)$,
- define the differential entropy $H(p)$ as

$$H(p) = -\int p(z) \ln p(z)\, dz$$

- $H(p)$ measures the spread of the distribution;
- $H(p) > 0$ $H(p) \to \infty$ → more uncertainty
- $H(p) < 0$ $H(p) \to -\infty$ - no uncertainty

- For a multivariate Gaussian $p(z) = N(w|\mu, \Sigma)$, differential entropy is

$$H(p) = H\left(N(w|\mu,\Sigma)\right) = \frac{d}{2}\ln\left(2\pi e|\Sigma|\right)$$

- Entropy for our multivariate Gaussian example depends on
  - i) d - dimensionality of distribution (of w)
  - ii) $|\Sigma|$ - (determinant) of the posterior covariance matrix of w
- How does posterior covariance change as new measurements are added, and how is this reflected in differential entropy H(p)?
- prior $\Sigma = (\lambda I + \sigma^{-2}X^TX)^{-1}$ $\overset{\text{transf}}{\Rightarrow}$ posterior $\left(\Sigma^{-1} + \sigma^{-2}x_0x_0^T\right)^{-1} = \left(\lambda I + \sigma^{-2}(x_0x_0^T + X^TX)\right)^{-1}$
- posterior covariance <u>does not depend</u> on y_0, it can be calculated without $y_0 \Rightarrow$
- differential entropy of prior and posterior related via 'rank-one update property of determinant':-

$$H_{post} = H_{prior} - \frac{d}{2}\ln\left(1 + \sigma^{-2}x_0^T\Sigma x_0\right)$$

(Σ: prior covariance of w; x_0: location of covariant vector)

differential entropy update is a function of x_0 and Σ

- The x_0 that minimises H_{post} also maximises σ_0^2
- H is being minimised myopically - greedy algorithm
- Pick x_0 for which predictive uncertainty σ_0^2 is the greatest
- is equivalent to minimising posterior uncertainty H_{post} of distribution of w.
- H_{post} (posterior 'uncertainty' measured by differential entropy of distribution)

<u>Model selection via Bayesian evidence maximisation</u>

- we had a prior on $w \sim N(0, \lambda^{-1}I)$
- How can we choose to select λ? Effectively a nuisance parameter
- cross-validation is one way

$$p(w|y,X,\lambda) = \frac{p(y|w,X)\,p(w|\lambda)}{p(y|X,\lambda)}$$

(i) (ii) (iii) (iv)

i) posterior of w given data (y, X) and λ

(?) ii) ~~prior~~ likelihood of data (*)

iii) prior conditional on λ

iv) Evidence is (marginal) likelihood of data given covariates (X) and hyperparameter (λ) with w integrated out

- set λ by maximising evidence $p(y|X,\lambda)$
- $\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}} \ln p(y|X,\lambda)$
- 'choose λ to be argmax of log of magical likelihood of the data having integrated out unknown parametes w'
- distribution of y $p(y|X,\lambda) = N(y|\underbrace{0}_{\mu}, \underbrace{\sigma^2 I + \lambda^{-1} XX^T}_{\Sigma})$
- requires algorithm
- This an example of Type-II ML or Empirical Bayes

  Type-I ML: maximise likelihood ow many parametes w

  Type-II ML: integrate out many parametes w and maximise ow hyperparametes λ (Empirical Bayes)
- Requires we can solve integral

sequential updating

measured data (y, X)

prior $p(w)$: $w \sim N(0, \lambda^{-1} I)$

likelihood $p(y|w, X)$: $y \sim N(Xw, \sigma^2 I)$

→ Posterior

$p(w|y, X) = N(w|\mu, \Sigma)$

$\mu = (\lambda\sigma^2 I + X^T X)^{-1} X^T y$

$\Sigma = (\lambda I + \sigma^{-2} X^T X)^{-1}$

predictive distribution (given we have x_0 but not y_0)

$$p(y_0|x_0, y, X) = \int_{\mathbb{R}^d} p(y_0|x_0, w) p(w|y, X) dw$$

$$= N(y_0|\mu_0, \sigma_0^2)$$

$\mu_0 = x_0^T \mu$

$\sigma_0^2 = \sigma^2 + x_0^T \Sigma x_0$ → select to observe corresponding y_0 of x_0 which has highest σ_0^2

observe y_0

New prior (posterior from last round) $p(w|y, X)$

New likelihood

$p(y_0|w, x_0)$ $y_0 \sim N(x_0^T w, \sigma^2)$

→ Updated posterior

$p(w|y_0, x_0, y, X) = N(w|\mu, \Sigma)$

$\mu = \left(\lambda\sigma^2 I + \left(x_0 x_0^T + \sum_{i=1}^{n} x_i x_i^T\right)\right)^{-1}$

$\Sigma = \left(\lambda I + \sigma^{-2}\left(x_0 x_0^T + \sum_{i=1}^{n} x_i x_i^T\right)\right)^{-1}$

$\left(x_0 y_0 + \sum_{i=1}^{n} x_i y_i\right)$

Incorporate $y \leftarrow y_0, y$ $X \leftarrow x_0, X$

✓

- Now write-up

+ Insert Prior + likelihood → posterior

$$p(w|y, X) = \frac{p(y|w, X) p(w)}{\int_{\mathbb{R}^d} p(y|w, X) p(w) dw}$$

- ~~spec~~
- directly specify predictive ~~$\Leftarrow$~~ $p(y_0 | x_0, y, X) = \int_{\mathbb{R}^d} p(y_0 | x_0, w) p(w | y, X) dw$
- provide a sequential update rule for posterior

$$p(w | y_0, x_0, y, X) = N(w | \mu, \Sigma)$$

$$\mu = (\lambda\sigma^2 I + X^T X)^{-1} X^T y$$

$$\Sigma = (\lambda I + \sigma^{-2} X$$

$$p(w | y_0, x_0, y, X) = N(w | \mu + [\dots], \Sigma + [\dots])$$

- loop $p(w | y_0, x_0, y, X) = \text{new } p(w | y, X)$

- setup:-
- Regression problem with $y = Xw$ $y \in \mathbb{R}^n$ $X \in \mathbb{R}^{n \times d}$ $w \in \mathbb{R}^d$
- Key condition → $(d >> n)$ i.e. high dimensionality and more features than observations
- mathematically, more variables than there are equations available
- infinite solutions w satisfying $y = Xw$

$$\begin{Bmatrix} y \end{Bmatrix} = \begin{bmatrix} \underbrace{X}_{d} \end{bmatrix} \begin{bmatrix} w \end{bmatrix}$$

- Key applications → gene analysis (computational biology)
  image analysis (computer vision)
  polynomial regressions

Minimum l_2 regression (least norm solution)

- one possible solution:

$$w_{ln} = X^T(XX^T)^{-1}y \Rightarrow Xw_{ln} = XX^T(XX^T)^{-1}y$$

- Can find a rule for generating other solutions by appealing to null spaces in linear algebra.
- Add to w_{ln} a non-zero vector $\partial \neq 0 \in \mathbb{R}^d$ that lies in the <u>null space</u> of X:

$$\partial \in N(X) \Rightarrow X\partial = 0 \text{ and } \partial \neq 0$$

$$\Rightarrow X(w_{ln} + \partial) = Xw_{ln} + X\partial = y + 0$$

- For a general matrix $A \in \mathbb{R}^{m \times n}$, the nullspace $N(A)$ is the set of all vectors equal to 0 when multiplied by A
  $N(A) = \{x \in \mathbb{R}^n : Ax = 0\}$

- $(d >> n) \Rightarrow$ infinite number of possible ∂ and hence solutions "w".

show that w_{ln} is solution to underdetermined problem with ~~least~~ smallest l_2 norm

least norm solution via analysis)

Prove that $w_{ln} = \text{argmin} \|w\|^2$ st $Xw = y$

Let w_{sol} be another arbitrary solution to $Xw = y$; then $Xw_{sol} - Xw_{ln} = 0$

and hence $X(w_{sol} - w_{ln}) = 0$ $\quad \partial = (w_{sol} - w_{ln})$

observe that :-

$$(w_{sol} - w_{ln})^T w_{ln} = (w_{sol} - w_{ln})^T X^T (XX^T)^{-1} y$$

$$= \underbrace{\left(X(w_{sol} - w_{ln})\right)^T}_{=0} (XX^T)^{-1} y$$

$$= 0$$

- substituting $w_{ln} = X^T(XX^T)^{-1}y$ on RHS of product only
- rewriting via transpose

Hence $(w_{sol} - w_{ln})$ is orthogonal to w_{ln}

- 2 vectors are <u>orthogonal</u> if $x^Ty = 0$

difference between an arbitrary solution and least norm solution is orthogonal to least norm solution.

The l_2 norm squared of w_{sol} can be decomposed to include w_{ln}

$$\|w_{sol}\|^2 = \|w_{sol} - w_{ln} + w_{ln}\|_2^2 = \|w_{sol} - w_{ln}\|_2^2 + \|w_{ln}\|_2^2 + \underbrace{2(w_{sol} - w_{ln})^T w_{ln}}_{=0} > \|w_{ln}\|^2$$

- As $\|w_{sol} - w_{LN}\|_2^2$ and $\|w_{LN}\|_2^2$ must be non-negative ~~$\Rightarrow \|w_{sol}\|$~~

- $\|w_{sol} - w_{LN}\|_2^2 + \|w_{LN}\|_2^2 \geq \|w_{LN}\|_2^2$

with equality when $w_{sol} = w_{LN}$

least L_2^2 norm via Lagrage multiplies

- start from optimisation problem :-

$$w_{LN} = \underset{w}{\operatorname{argmin}}\ w^T w \text{ subject to } Xw = y$$

- objective fn: $f(\underline{w}): \mathbb{R}^d \to \mathbb{R}$ $w \mapsto w^T w$
- constraint fn: $g(\underline{w}) = \underline{0} : \mathbb{R}^d \to l$ $w \mapsto y - Xw$
  $y - Xw$
- $\eta \in \mathbb{R}^n$

Define a Lagrangian $\mathcal{L}(\underline{w}, \underline{\eta}) = w^T w - \eta^T (y - Xw)$
$= w^T w + \eta^T (Xw - y)$

i) $\nabla_w \mathcal{L}(\underline{w}, \underline{\eta}) = 2\underline{w} + \eta^T X - \underline{0}$; ii) $\nabla_\eta \mathcal{L}(\underline{w}, \underline{\eta}) = X\underline{w} - y = \underline{0}$

- Taking gradients of Lagrangian wrt each vector argument
- $w^T X^T \eta = \eta^T X w$ (?)

i) $\Rightarrow \underline{w} = -\frac{\eta^T X}{2} = -\frac{X^T \eta}{2}$

i) $\to$ ii) : iii) $X\left(-\frac{X^T \eta}{2}\right) - y = 0 \Rightarrow \eta = -2y(XX^T)^{-1}$

iii) $\to$ i) $\Rightarrow 2w + X^T \eta = \underline{0} \Rightarrow w_{LN} = X^T (XX^T)^{-1} y$

sparse L_1 regression

- LS and RR generally not suitable for underdetermined / high dimensional problems, characterise many modern applications
- often only a few important for prediction of $y \to$ feature selection (see Tibshrani)
- LS and RR treat all dimensions equally without favouring subsets
- relevant dimensions averaged with irrelevant $\Rightarrow$ i) poor generalisation (overfitting)
  ii) poor interpretability

- Ridge regression uses objective/loss :-

$$\mathcal{L} = \sum_{i=1}^{n} (y_i - f(x_i; w))^2 + \lambda \|w\|_2^2 = \quad \|w\|_2^2 \text{ - penalty } \quad f(x_i; w) = x_i^T w$$

- more generally :-

Total cost/loss = goodness of fit term + penalty term
(i) (ii) (iii)

ii) measures how well model approximates data (in-sample?)

iii) allows preference over solutions by making them 'expensive' wrt $\mathcal{L}$

- what kinds of solutions does $\|w\|^2$ favour?
- Quadratic penalty intuitions :-

penalty: $\lambda \|w\|_2^2 : \lambda \sum_{i=1}^{d} w_i^2$ For an individual w_j, $\frac{\partial \lambda \sum_{i=1}^{d} w_i^2}{\partial w_j} = 2w_j \lambda$

- rate of change of penalty increase is a function of w_j and hence its magnitude and starting point
- favour vectors $\underline{w}$ whose entries are of similar size and small

- l_2 norm penalty $\not\Rightarrow$ sparsity and encourages solutions which are equal in size (small, non-zero)
- In our regression problem, we would like to select a small subset of the d features and 'switch off' the rest (make them subsidiary to explanation/prediction)
- Switching off dimensions - illustration :-
  - Each w_j corresponds to a dimension of data x
  - $w_j = 0$; $f(x_j w) = x^T w = w_1 x_1 + \dots + 0 \cdot x_j + w_d x_d$
  - $\Rightarrow$ Prediction does not depend on j^{th} 'regressor/dimension'
- Feature selection : Find $\underline{w}$ that i) predicts well
  ii) small no. of non-zero entries
- $\underline{w}$ for which most w_j entries are $0 \Rightarrow$ sparse solution
- Maybe this is done with respect to our intuitions about how localised/diffuse causality prediction is.

(ETT) Sparse solutions can be achieved with linear penalty terms
- If w_k is large and other w_j small and non-zero
Quadratic penalty : Favour entries with similar size and push w_k towards those (average) small values
linear (sparse) penalty : Keep w_k, push w_j to zero

LASSO - L_1 regularised regression

LASSO - Least Absolute Shrinkage and Selection Operator
L_1 regularised regression

$$w_{lasso} = \underset{w}{\operatorname{argmin}} \|y - Xw\|_2^2 + \lambda \|w\|_1$$

where $\|w\|_1 = \sum_{j=1}^{d} |w_j|$

Ridge regression vs LASSO level sets; coefficient profiles

Make sure you can draw the diagrams; understand intuitions (practise)
Level sets in red represent locus ~~of~~ of parameter combinations in $d=2$ case (w_1, w_2) which pay same ~~fixed~~ for 'price' under goodness of fit
Level sets in blue represent locus of parameter combinations in $d=2$ case (w_1, w_2) which pay same penalty
This is a visualisation for $d=2$, but can generalise intuition to $d \gg n$
L_1 regularisation tends to select point where $w_1 = 0$ and $w_2 > 0$
— " — where $w_1 > 0$ and $w_2 > 0$ but smaller and 'averaged'.

- Coeffcient profiles :-
- under LASSO/l_1 regularisation, 3 out of 8 parameters are estimated be positive

l_p regression

- l_p norms
- Norm penalties can be extended to all norms :-

$$\|w\|_p = \left(\sum_{j=1}^{d} |w_j|^p\right)^{1/p} \quad \text{for } 0 \leq p \leq \infty$$

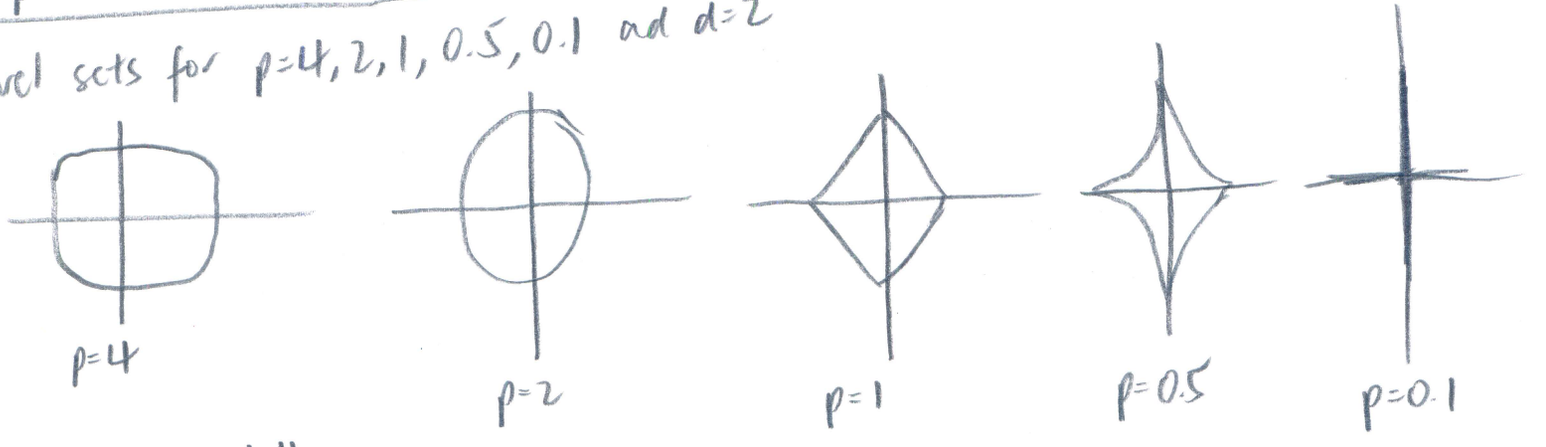
- l_p regression :-

$$w_{l_p} := \underset{w}{\operatorname{argmin}} \|y - Xw\|_2^2 + \lambda \|w\|_p^p$$

- For $p=1 \rightarrow$ LASSO
  $p=2 \rightarrow$ ridge

l_p penalisation terms (level sets)

Level sets for $p=4, 2, 1, 0.5, 0.1$ and $d=2$



Behaviour of $\|\cdot\|_p$

$p=\infty$ - Norm measures largest absolute entry: $\|w\|_\infty = \max_j |w_j|$
$p>2$ - Norm focuses on large entries
$p=2$ - Large entries expensive (ridge)
$p=1$ - Sparsity encouraged (LASSO)
$p<1$ - Encourages sparsity, but unitow set not convex $\Rightarrow$ implications for optimisation procedure
$p \to 0$ - Records whether an entry is non zero $\|w\|_0 = \sum_j \mathbb{I}(w_j \neq 0)$

Solutions to l_0 - l_p regularised regressions

	GOF	PENALTY	SOLUTION
• least squares	$\|y-Xw\|_2^2$	-	Analytic solution if X^TX invertible
• Ridge reg.	$\|y-Xw\|_2^2$	$\|w\|_2^2$	Analytic sol. always, closed form
• LASSO	$\|y-Xw\|_2^2$	$\|w\|_1$	Numerical optimisation from convex optimisation a (global solution)

ℓ_p $(p \geq 1, p \neq 2)$ - Find solutions via convex optimisation
- Global solution can be found via ~~convex opti~~ numerical algorithms

$(p < 1)$ - only approximate solution can be found using iterative algorithms (ie. no guarantee of global solution ie can only find best in neighbourhood)

Classification problem

Inputs: measurements $(x_1, \dots, x_n)$ in an input space $\mathcal{X} = \mathbb{R}^d$ (2)

Output: discrete output space $\mathcal{Y}$ of K classes:-

$\mathcal{Y} = \{-1, +1\}$ or $\{0, 1\}$ (binary classification) $\mathcal{Y} = \{1, \dots, K\}$ (multiclass classification)

- classification assigns x to a category; for (x, y), y is class of x
- define a function f (classifier) to map input x to class y

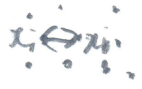
$y = f(x)$: f takes x in $\mathcal{X}$ and declares class to be $y \in \mathcal{Y}$

- Task is to define classifier f and estimate its 'parameters' using labelled data

Nearest neighbour (NN) classifier

- Given $(x_1, y_1), \dots, (x_n, y_n)$ construct classifier $\hat{f}(x) \to y$ as follows:- [see diag]
- For a new input x not in training data:-
  1. Let x_i be point among $x_1, \dots, x_n$ closest to x
  2. Return label y_i.

- data in $\mathbb{R}^2$



- Find the closest neighbour to input and return neighbour label

Q: How to measure distance (~~not~~ what metric do we define)?

default distance for data in $\mathbb{R}^d$

l_2 norm/Euclidean: $\|u - v\|_2 = \left(\sum_{i=1}^{d} (u_i - v_i)^2\right)^{1/2}$ (line of sight)

l_p for $p \in [1, \infty]$: $\|u - v\|_p = \left(\sum_{i=1}^{d} |u_i - v_i|^p\right)^{1/p}$

- For data not in $\mathbb{R}^d$, use edit distance for strings and correlation distance for signals
- Distance measure may depend on the nature of the input data.

OCR with NN-classifier example

- Note that $y \in \{0, 1, \dots, 9\}$ each $x_i \in \mathbb{R}^{784}$
- Handwritten digits data → 28×28 grayscale pixel images
- Training error $\text{err}(\hat{f}, S) = 0$ → declare class to be its own class
  Test error $\text{err}(\hat{f}, T) = 0.0309$ → using l_2 distance and NN-classifier (97% accuracy)
- Mistakes could be avoided by setting 3 nearest neighbours and majority vote.

(K-NN) K-nearest neighbour classifier

- Given data $(x_1, y_1) \dots (x_n, y_n)$ construct KNN classifier $\hat{f}(x) \to y$ as follows
- For a new input x
  1. Return k points closest to x, indexed $x_{i1}, x_{i2}, \dots x_{ik}$
  2. Return majority vote of $y_{i1}, y_{i2}, \dots y_{ik}$
- Break ties in both steps arbitrarily
- OCR with K-NN classifier → sweep for various values of K and choose the best one.

effect of K. (see diagram)

smaller K ⇒ smaller training error

large K ⇒ predictions more stable due to majority voting

- K can be interpreted as a smoothing parameter
- decision boundary is region of problem space in which output label of a classifier is ambiguous/changes
- diagram shows 15-NN on $\mathbb{R}^2$ data with 3 classes giving smoothe decision boundaries as each class 'stakes out a region'.

classifier quality

Q: How do we measure quality of a classifier?

prediction accuracy $P(f(x)=y)$ and prediction error $err(f) = P(f(x) \neq y)$

A: In order to calculate these values, we assume a distribution P over space of labelled examples generating the data

$$(x_i, y_i) \overset{iid}{\sim} P \quad i=1, \ldots, n$$

(x_i, y_i) is drawn iid / generated from an (unknown) ground truth distribution, joint distribution over covariates and class label

Even though we don't know P, we can still conduct discussion about it in abstract terms.

statistical learning theory

When is there hope of finding an accurate classifier?

Key assumption is philosophically a uniformity of nature assumption

$(x_1, y_1) \ldots (x_n, y_n)$ are iid labelled examples from distribution P

takes the claim that the past should look like the future → (a)

(a) ..... new unlabelled examples from P

past examples from P → learning algorithm → learned predictor → predicted label

iid / uniformity of nature assumption

ows a link between past and future examples; allows for generalisation

oretical guarantees on this in more detail are given in the field of statistical ning theory, pioneered by Vladimir Vapnik

nal classifiers - 2 key probability equalities

we talk about what an 'optimal' classifier looks like? (Yes - Bayes classifier)

ssume $(X, Y) \overset{iid}{\sim} P$ (where P is unknown) CONTEXT: Joint distribution of email contents and label spam/non-spam

y probability equalities (with P) - note not calculable in practice without access to known P

- theory only motivates characterisation of optimal classifier

1. Expectation of an indicator of an event is probability of event eg.

$\mathbb{E}_P[\mathbb{1}(Y=1)] = P(Y=1)$

$\mathbb{1}(\cdot)$ is an indicator function $= \begin{cases} 1 \text{ if } \cdot \text{ event is true} \\ 0 \text{ if } \cdot \text{ event is false} \end{cases}$

more explicitly:-

$E_P[\mathbb{1}(Y=1)] = 1 \cdot P(Y=1) + 0 \cdot P(Y \neq 1) + \ldots$

2. conditional expectations can be random variables (or fixed values), i) depending on nature of conditioning; their expectations 'remove' the randomness

- For 2 r.v.s A and B $E(A|B)$ is itself a random variable if the particular value of what is being conditioned on is not specified

  $E(A|B=b)$ is fixed if the particular value of B being conditioned on is specified

- i) $C = E[A|B]$; C and $E[A|B]$ is an r.v.; as A and B both r.v.s.; no specification of values being conditioned over

- ii) $E(C) = E[E[A|B]] = E[A]$; law of iterated/total expectations
  - Expectations of conditional expectations remove randomness
  - Expectation of C (over integrating B) is expectation of A (with B integrated out)

Optimal classifier

- maths, then comments/intuition:-

For any classifier $f: X \to Y$ its prediction error is:- (*) note

$$P(f(X) \neq Y) = E[\mathbb{1}(f(X) \neq Y)] = E[E[\mathbb{1}(f(X) \neq Y)|X]] \quad (\dagger)$$

For each $x \in X$ (or fixing

$$E[\mathbb{1}(f(X) \neq Y)|X=x] = \sum_{y \in Y} P(Y=y|X=x) \cdot \mathbb{1}(f(x) \neq y) \quad (\ddagger)$$

$(\ddagger)$ is minimised for a particular $x \in X$ when

$$f(x) = \underset{y \in Y}{\operatorname{argmax}}\ P(Y=y|X=x) \quad (\star)$$

- classifier f with property $(\star)$ is known as the Bayes classifier; has smallest prediction error ($\dagger$) amongst all classifiers

Comments/intuition:-

- optimal classifier minimises probability of making a mistake according to key assumption that X and Y are both generated randomly iid from $\mathbb{P}$ joint distribution P. - PREDICTION ERROR
- Probability $P(f(X) \neq Y)$ classifier makes mistake is the expectation of the indicator that the classifier makes a mistake $E[\mathbb{1}(f(X) \neq Y)]$
- This "outer" expectation uses true underlying distribution P to calculate/integrate over
- Tower property of conditional expectation :-
  i) First calculate conditional expectation of indicator given specific values of $X = x$ (pretend we know it); for that specific value what is expectation of indicator that label is not correctly predicted
  ii) Then take expectation of that; "randomness" comes from Y
- $E[\mathbb{1}(f(X) \neq Y) | X]$ is a random variable; classifier f is not random, given $X = x$ $f(X = x)$ is not random: classifier always predicts label for x. But underlying distribution P does not have a deterministic distribution on y given x.
  e.g. same email might 95% of time be spam, but 5% time not spam.

(♯): in this expression, X is fixed at $X = x$ and Y is random; expectations taken over possible outcomes of Y (weighted average of values $\mathbb{1}(f(x) \neq y)$ can take for different values of y, weighted by (conditional) probability of that value being realised)

THEN outer expectation is then taken with X random/allowed to vary; expectations taken over possible outcomes of $\sum_{y \in Y} P(Y = y | X = x) \cdot \mathbb{1}(f(x) \neq y)$ weighted by probability of that outcome being realised.

- nicer (♯♯)

(♯): classifier f assigns one of K possible labels to every point $x \Rightarrow \mathbb{1}(f(x) \neq y) = 0$ for only one value of y
RHS: For label y we assign to x, sum up probabilities of all labels other than that (assignment) i.e. sum probability we get it wrong
In order to minimise, assign label to x to be most probable label according to distribution specified by Nature P.

$\Rightarrow \mathbb{1}(f(x) \neq y) = 0$ for maximum value $P(Y = y | X = x)$

- Then sum up smallest $(K-1)$ probabilities according to distribution $P(Y = y | X = x)$

): If labelled data iid from distribution P from Nature; for every $x \in X$, predict the label to be most probable label conditioned on x, according to that distribution
$\Rightarrow$ minimise probability of error

- However, we do not know distribution P and hence $P(Y=y, X=x) \Rightarrow$ approximation
- Hence no guarantee of optimal classifier

Bayes classifier

- under (X,Y) iid P; the optimal classifier is

$$f^*(x) := \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\ P(Y=y \mid X=x)$$

via Bayes Rule :-

$$P(Y=y \mid X=x) = \frac{P(X=x \mid Y=y)\, P(Y=y)}{P(X=x)}$$

(i) (ii) (iii) (iv)

(i) - posterior probability of label of x, given x

ii) prior probability of that label

iii) likelihood of x given label

iv) normalising constant not dependent on y and hence not relevant for argmax

$$\Rightarrow f^*(x) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\ P(Y=y) \cdot P(X=x \mid Y=y)$$

- $P(Y=y)$ - class prior (a priori prevalence of class)
- $P(X=x \mid Y=y)$ - class conditional distribution of X (given it comes from class Y)
- Both not known $\Rightarrow$ approximate
- X continuous; replace $P(X=x \mid Y=y)$ with class conditional density $p(x \mid Y=y)$

Gaussian class conditional densities

- suppose $\mathcal{X} = \mathbb{R}$ $\mathcal{Y} = \{0,1\}$; distribution of P of (X,Y) :-
  - class prior $P(Y=y) = \pi_y$ $y \in \{0,1\}$
  - class conditional density for class $y \in \{0,1\}$: $p_y(x) = N(x \mid \mu_y, \sigma_y^2)$ → $p_0(x)$, → $p_1(x)$
  - Bayes classifier :-

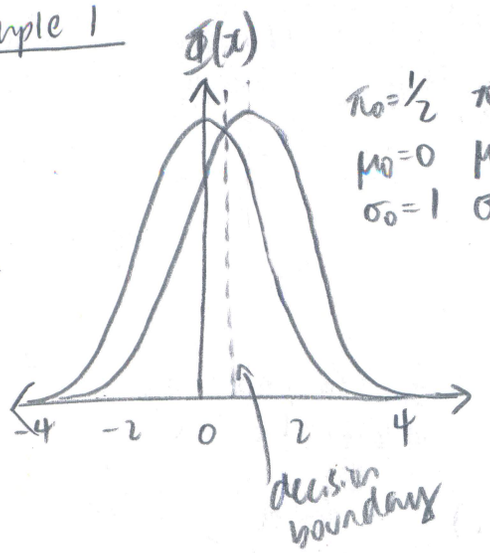
$$f^*(x) = \underset{y \in \{0,1\}}{\operatorname{argmax}}\ P(X=x \mid Y=y) P(Y=y)$$

$$= \begin{cases} 1 \text{ if } \frac{\pi_1}{\sigma_1} \exp\left[-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right] > \frac{\pi_0}{\sigma_0} \exp\left[-\frac{(x-\mu_0)^2}{2\sigma_0^2}\right] \\ 0 \text{ otherwise} \end{cases}$$

(2) $\frac{\pi_1}{\sigma_1}$ and $\frac{\pi_0}{\sigma_0}$ $\sqrt{2\pi}$s cancel out

- generative model : model x and y with distributions
- discriminative (model) : plug x into a distribution on y
  no distributional assumptions on x,
  predict y conditioning on input x
  $p(y \mid x)$

Example 1



$\pi_0 = \frac{1}{2}$ $\pi_1 = \frac{1}{2}$
$\mu_0 = 0$ $\mu_1 = 1$
$\sigma_0 = 1$ $\sigma_1 = 1$

$\frac{1}{2}$ x's from $N(0,1) \to y=0$
$\frac{1}{2}$ x's from $N(1,1) \to y=1$

plug in π_0, π_1, μ_0, μ_1 and σ_0, σ_1 solve inequality for x

Bayes classifier

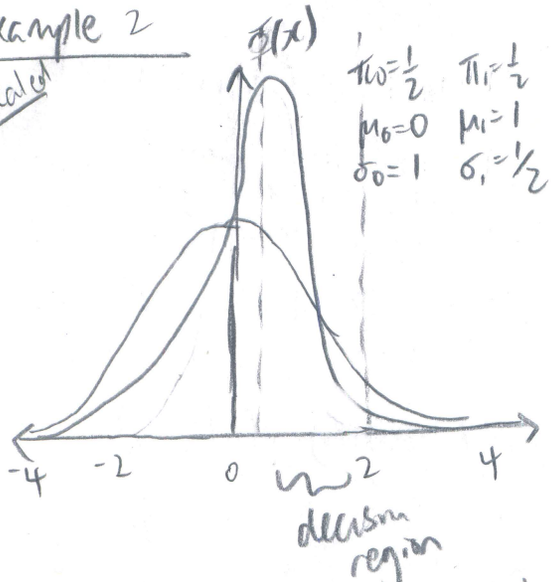
$$f^*(x) = \begin{cases} 1 & \text{if } x > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

B.B - minimising probability of error according to 2 probability distr.

$$\phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

Example 2

rescaled



$\pi_0 = \frac{1}{2}$ $\pi_1 = \frac{1}{2}$
$\mu_0 = 0$ $\mu_1 = 1$
$\sigma_0 = 1$ $\sigma_1 = \frac{1}{2}$

$\frac{1}{2}$ x's from $N(0,1) \to y=0$
— " — from $N(1, \frac{1}{4}) \to y=1$

Bayes classifier

$$f^*(x) = \begin{cases} 1 & \text{if } x \in [0.38, 2.29] \\ 0 & \text{otherwise} \end{cases}$$

Example 3 - multivariate Gaussians

- Data $X = \mathbb{R}^2$ label $Y = \{0,1\}$
- class conditional densities are Gaussians in $\mathbb{R}^2$ with covariance Σ_0 and Σ_1
- see diagrams $\Sigma_0 = \Sigma_1$: Bayes classifier → linear separator
  $\Sigma_0 \neq \Sigma_1$ — " — → quadratic "
- Bayes classifier in general → complicated if class conditional density complicated

Issues with Bayes classifier and plug-in classifiers

- Bayes classifier has smallest prediction error of all classifiers
- issue: we cannot construct Bayes classifier without knowing P
  → what is $P(Y=y|X=x)$ or $P(X=x|Y=y)$ and $P(Y=y)$?
  → only labelled examples drawn (assumed iid) from P
- In general, we do not have class conditional density of each class ⇒ pick and approximate

Plug-in classifiers

- use available data to approximate $P(Y=y)$ and $P(X=x|Y=y)$

- no guarantee of best result amongst all classifiers (eg. RNN etc.)

Example 4 - Gaussian class conditional densities and plug-in

- Data: $\mathcal{X} = \mathbb{R}^d$ $\mathcal{Y} = \{1, \dots, K\}$; estimate Bayes classifier via MLE:-
  - Class priors: MLE estimate of π_y is $\hat{\pi}_y = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(y_i = y)$ - empirical distribution of labels in data set - sum up no. times class y appears in data set; divide by obs.
  - Class conditional density:- choose $p(x|Y=y) = N(x|\mu_y, \Sigma_y)$
    MLE estimate of (μ_Y, Σ_y) is:- (K lots of each estimate $\hat{\mu}_Y$ and $\hat{\Sigma}_Y$)

$$\hat{\mu}_Y = \frac{1}{n_y}\sum_{i=1}^{n} \mathbb{1}(y_i = y)\, x_i \qquad \text{empirical mean of class } y$$

$$\hat{\Sigma}_y = \frac{1}{n_y}\sum_{i=1}^{n} \mathbb{1}(y_i = y)(x_i - \hat{\mu}_y)(x_i - \hat{\mu}_y)^T \qquad \text{empirical covariance of class } y$$

- Limited to class we are learning / approximate class conditional density using Gaussian even though it may not be; for K classes then each class has own MVG from which obsev. generated
- Plug-in classifier:-
- For new x which we want to assign to one of K classes :-

$$\hat{f}(x) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \left[\hat{\pi}_y |\hat{\Sigma}_y|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2}(x - \hat{\mu}_y)^T \hat{\Sigma}_y^{-1} (x - \hat{\mu}_y) \right\} \right]$$

- Pick argmax over y i.e. evaluate fn in the new x for each value of y corresponding to the K classes and predict y to be the max out of these.

Spam filtering example

- Representing emails:-
  Input: x a vector of word counts e.g. ($j \to$ car) $x(j) = 3 \Rightarrow$ car occurs 3 times in email
  (bag of words representation with numbers representing frequencies)
  Output: $\mathcal{Y} = \{-1, +1\}$ Map {email $\to -1$, spam $\to +1$}
  Using Bayes classifier:- $f(x) = \underset{y \in \{-1,+1\}}{\operatorname{argmax}}\ p(x|Y=y)\,p(Y=y)$ (?)

Naive Bayes in context of spam filtering

- Have to define $p(X=x|Y=y)$; MVG not reasonable in this case (why?)
- Simplifying assumption:-
  - Naive Bayes is a Bayes classifier that makes assumption:-

$$p(X=x|Y=y) = \prod_{j=1}^{d} p_j(x(j)|Y=y)$$ - treats dimensions of X as conditionally independent given Y=y

  - In spam context → correlation between words ignored, easier to define distribution
  - Distribution of 'histogram of word counts' given class assignment is independent

Estimation of N.B.

- **Class prior**: Distribution $P(Y=y)$ can be simply estimated from training data:-

$$P(Y=y) = \frac{\#\text{obs. in class } y}{\#\text{observations}}$$

e.g. for Y = spam (no. of spam/total emails)

- **Class-conditional distributions**

Define $P(X=x|Y=y) = \prod_j P_j(x(j)|Y=y) = \prod_j \text{Poisson}(x(j)|\lambda_j^{(y)})$

- Approximate each $\lambda_j^{(y)}$ from the data e.g. ~~MLE~~ - the MLE is:-

$$\lambda_j^{(y)} = \frac{\#\text{unique uses of word } j \text{ in observations from class } y}{\#\text{observations in class } y}$$

- $\text{Poisson}(x(j)|\lambda_j^{(y)})$: Given an email comes from class y, no. of occurences is Poisson distribution with λ_j for class y
  - each word has parameter associated with its Poisson distribution and class depart.
- Condition class distributn in spam email:- (labelled)
  - count how many times word 'car' appears in actual spam email; take all spam, count total occurrences of word car; divide by no of emails in that class (spam class)
  - Do for all words in vocabulary; both classes ⇒ parameter for class specific distribution on word histogram