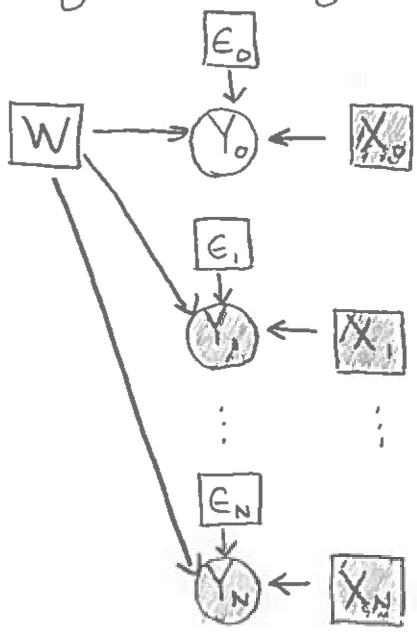
Linear Regression: MAP

① Recall "ordinary linear regression":



where: $P_\epsilon(\epsilon_n) \sim \text{Normal}(0, \sigma^2) \qquad \forall n \in \{0, \dots, N\}$

$y_n \leftarrow w^T x_n + \epsilon_n$

② Also recall that one way to estimate the value of the unobserved response variable Y_0 is through maximum a posteriori (MAP) estimation:

(a) compute $\hat{w} = \underset{w}{\operatorname{argmax}}\ P(w) \prod_{n=1}^{N} P(y_n \mid w, x_n)$

(b) compute $\hat{y}_0 = \underset{y_0}{\operatorname{argmax}}\ P(y_0 \mid \hat{w}, x_0)$

In the MLE approach, we assume that all weight vectors are equally likely (without further evidence), so we treat $P(w)$ as a constant and drop it from the equation.

Linear Regression: MAP

③ But maybe we do have an opinion about which weight vectors are more likely prior to observing any evidence (this is called a prior probability or an apriori belief).

First off, why would we have such an opinion?

④ Consider if we actually wanted to predict someone's cholesterol accurately on the basis of lifestyle factors. We don't know what might be relevant, so we throw a lot of evidence variables into the mix:

X (evidence vars)						Y (response var)
$X[1]$ (offset)	$X[2]$ (age)	$X[3]$ (weight)	$X[4]$ (smoking freq)	...	$X[10000]$ (gumchewing freq)	(cholesterol)

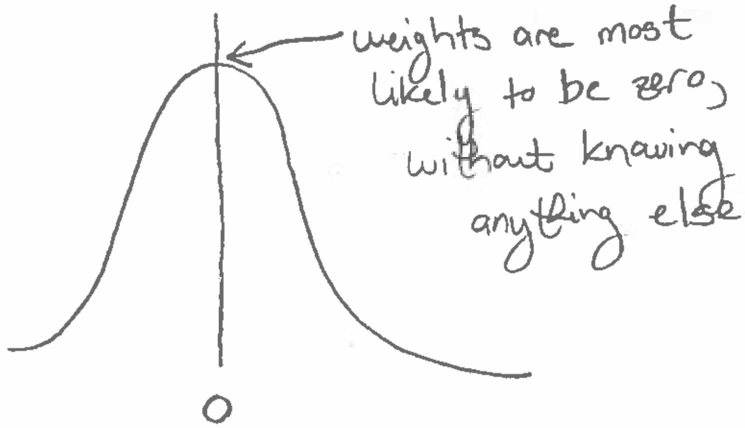
⑤ Most of these evidence vars probably don't have any impact on cholesterol, so we expect that a good weight vector $W = \begin{bmatrix} W[1] \\ \vdots \\ W[10000] \end{bmatrix}$ will contain mostly zeroes, since then $Y = w^T x$ will only be a function of a small subset of the evidence vars.

Linear Regression: MAP

6) So our apriori belief is that weights are most likely to be zero.

How can we express this as a distribution?

One way is to say that for each weight $W[d]$, $P(W[d]) \sim \text{Normal}(0, \sigma^2)$ for some variance σ^2:



7) So let's see if we can simplify 2(a), our point estimate:

$$\hat{w} = \underset{w}{\operatorname{argmax}}\ P(w) \prod_{n=1}^{N} P(y_n \mid w, x_n)$$

$$= \underset{w}{\operatorname{argmax}}\ \log P(w) \prod_{n=1}^{N} P(y_n \mid w, x_n)$$

$$= \underset{w}{\operatorname{argmax}}\ \ell(w)$$

LINEAR REGRESSION: MAP

(8) Continuing:

$$\ell(w) = \log P(w) + \sum_{n=1}^{N} \log P(y_n | w, x_n)$$

$$= \log P(w) + \ell_{MLE}(w)$$

where $\ell_{MLE}(w)$ is the likelihood function for the MLE (see LINEAR REGRESSION: MLE, (4))

---

(9) As with ordinary linear regression, we'll assume the stochastic terms ϵ_n are normally distributed, i.e.

$$P_\epsilon \sim \text{Normal}(0, \sigma^2).$$

We'll also use the prior distribution over weights that we argued for in (6):

$$P(w) \sim \text{Normal}(0, \alpha^2).$$

not necessarily the same variance

These choices give us a type of regression called <u>ridge regression</u>.

Linear Regression: MAP

(10) Continuing to simplify with these choices:

$$\hat{w} = \underset{w}{\operatorname{argmax}}\ \ell(w)$$

$$= \underset{w}{\operatorname{argmax}}\ \log P(w) + \ell_{MLE}(w)$$

$$= \underset{w}{\operatorname{argmax}}\ \log\left(\prod_{d=1}^{D}\left(\frac{1}{2\pi\alpha^2}\right)^{\frac{1}{2}}\exp\left(\frac{-1}{2\alpha^2}w[d]^2\right)\right) + \ell_{MLE}(w)$$

$$= \underset{w}{\operatorname{argmax}}\ \sum_{d=1}^{D}\log\left[\left(\frac{1}{2\pi\alpha^2}\right)^{\frac{1}{2}}\exp\left(\frac{-1}{2\alpha^2}w[d]^2\right)\right] + \ell_{MLE}(w)$$

$$= \underset{w}{\operatorname{argmax}}\ \left(\sum_{d=1}^{D}\frac{-1}{2}\log 2\pi\alpha^2\right) + \left(\sum_{d=1}^{D}\frac{-1}{2\alpha^2}w[d]^2\right) + \ell_{MLE}(w)$$

$$= \underset{w}{\operatorname{argmax}}\ \frac{-D}{2}\log 2\pi\alpha^2 - \frac{1}{2\alpha^2}\sum_{d=1}^{D}w[d]^2 + \ell_{MLE}(w)$$

$$= \underset{w}{\operatorname{argmax}}\ \frac{-\sum_{d=1}^{D}w[d]^2}{2\alpha^2} + \ell_{MLE}(w)$$

$$= \underset{w}{\operatorname{argmax}}\ \frac{-w^T w}{2\alpha^2} + \ell_{MLE}(w)$$

LINEAR REGRESSION: MAP

11) From LINEAR REGRESSION: MLE, 5, we know that:

$$\ell_{MLE}(w) = -\frac{N}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - w^T x_n)^2$$

$$= -\frac{N}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(y - Xw)^T(y - Xw)$$

Plugging this in to what we have so far:

$$\hat{w} = \underset{w}{\operatorname{argmax}} \frac{-w^T w}{2\alpha^2} \cancel{- \frac{N}{2}\log 2\pi\sigma^2} - \frac{1}{2\sigma^2}(y - Xw)^T(y - Xw)$$

$$= \underset{w}{\operatorname{argmax}} -\frac{1}{2\alpha^2} w^T w - \frac{1}{2\sigma^2}(y - Xw)^T(y - Xw)$$

$$= \underset{w}{\operatorname{argmax}} -\frac{\sigma^2}{\alpha^2} w^T w - (y - Xw)^T(y - Xw)$$

$= \ldots$ [See LINEAR REGRESSION: MLE 7]

$$= \underset{w}{\operatorname{argmax}} -\frac{\sigma^2}{\alpha^2} w^T w + 2w^T X^T y - w^T X^T X w$$

$$= \underset{w}{\operatorname{argmax}}\ 2w^T X^T y - w^T X^T X w - \frac{\sigma^2}{\alpha^2} w^T w$$

$$= \underset{w}{\operatorname{argmin}}\ w^T X^T X w - 2w^T X^T y + \frac{\sigma^2}{\alpha^2} w^T w$$

12) So the loss function for ridge regression is:

$$L_{ridge}(w) = w^T X^T X w - 2w^T X^T y + \frac{\sigma^2}{\alpha^2} w^T w$$

Notice that this can be expressed in terms of the loss function for ordinary linear regression $L_{lin}(w) = w^T X^T X w - 2w^T X^T y$:

$$L_{ridge}(w) = L_{lin}(w) + \frac{\sigma^2}{\alpha^2} w^T w$$

13) That means ridge regression's loss function is simply the usual linear regression loss function, plus some constant multiple of the <u>squared "L_2-norm"</u> of the weight vector:

$$w^T w = \sum_{d=1}^{D} w[d]^2 = \|w\|_2^2$$

$$\hat{w} = \underset{w}{\operatorname{argmin}}\ L_{ridge}(w)$$

$$= \underset{w}{\operatorname{argmin}}\ \underbrace{L_{lin}(w)} + \underbrace{K \cdot w^T w}$$

we want the likelihood of the data to be high

but we also want the weights to be close to zero

LINEAR REGRESSION: MAP

(14) Thus ridge regression's loss function is combining two different objectives:

(a) we want the likelihood of the training data to be high $\quad \arg\min_w L_{lin}(w)$

(b) we want the learned weight vector to have a small L_2-norm (i.e. we want the length of the weight vector to be small) $\quad \arg\min_w \|w\|_2^2$

Objective (b) is often called a <u>regularization term</u> and so ridge regression is sometimes known as <u>linear regression with L_2-regularization</u>.

---

(15) So going back to (2), ridge regression estimates the value of the unobserved response variable Y_0 as follows:

(a) compute point estimate $\hat{w} = \arg\max_w P(w) \prod_{n=1}^{N} P(y_n \mid w, x_n)$

$$= \arg\min_w L_{ridge}(w)$$

(b) compute $\hat{y}_0 = \arg\max_{y_0} P(y_0 \mid \hat{w}, x_0)$

$$= \hat{w}^T x_0$$ (see LINEAR REGRESSION: MLE, (3))

(16) Exercise: Adapt LINEAR REGRESSION: MLE (10)-(11) to compute a closed-form expression for $\underset{w}{\operatorname{argmin}}\, L_{ridge}(w)$.

Solution:

$$\nabla L_{ridge}(w) = \frac{\partial}{\partial w}\left(L_{lin}(w) + \frac{\sigma^2}{\alpha^2} w^T w\right)$$

$$= \frac{\partial}{\partial w} L_{lin}(w) + \frac{\sigma^2}{\alpha^2}\frac{\partial}{\partial w} w^T w$$

$$= \frac{\partial}{\partial w} L_{lin}(w) + \frac{\sigma^2}{\alpha^2}\begin{bmatrix} \frac{\partial}{\partial w[1]} w^T w \\ \vdots \\ \frac{\partial}{\partial w[D]} w^T w \end{bmatrix}$$

$$= \frac{\partial}{\partial w} L_{lin}(w) + \frac{\sigma^2}{\alpha^2}\begin{bmatrix} 2w[1] \\ \vdots \\ 2w[D] \end{bmatrix}$$

$$= \frac{\partial}{\partial w} L_{lin}(w) + \frac{2\sigma^2}{\alpha^2} w$$

$$= 2X^TXw - 2X^Ty + \frac{2\sigma^2 w}{\alpha^2}$$

Then $\nabla L_{ridge}(w) = 0$

$$\Rightarrow 2X^TXw - 2X^Ty + \frac{2\sigma^2 w}{\alpha^2} = 0$$

$$\Rightarrow \left(X^TX + \frac{\sigma^2}{\alpha^2} I\right) w = X^Ty$$

$$\Rightarrow w = \left(X^TX + \frac{\sigma^2}{\alpha^2} I\right)^{-1} X^Ty$$

Linear Regression: MAP

17) This gives us the following algorithm for ridge regression:

RidgeRegression$\left(X, y, \text{ratio } \frac{\sigma^2}{\alpha^2}, \text{input } x_0\right)$:

- compute point estimate $\hat{w} = \left(X^T X + \frac{\sigma^2}{\alpha^2} I\right)^{-1} X^T y$
- compute estimate $\hat{y}_0 = \hat{w}^T x_0$
- return $\hat{y}_0$

18) That's what we get if we assume the weights are drawn from a Normal$(0, \alpha^2)$ distribution.

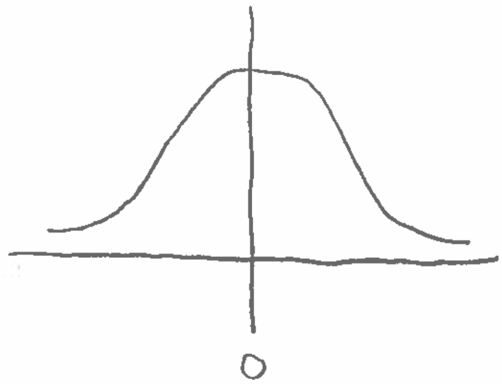
But the whole motivation was that we had a bunch of evidence variables, most of which were completely irrelevant:

X (evidence vars)					Y (response var)
X[1]	X[2]	X[3]	X[4]	X[10000]	
(offset)	(age)	(weight)	(smoking freq)	(gumchewing freq)	(cholesterol)

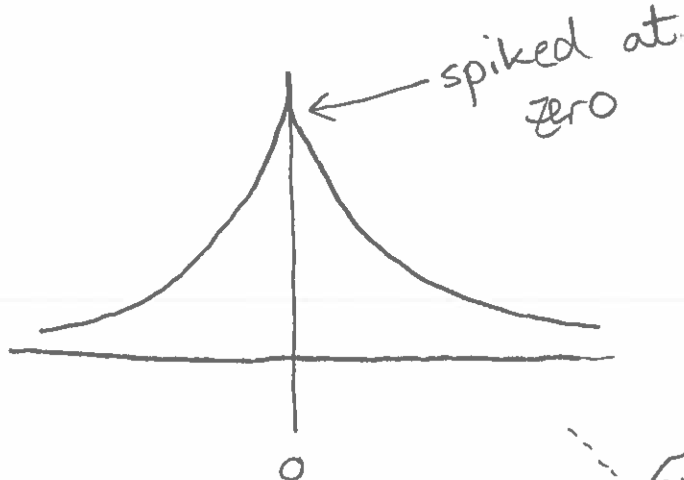
Ideally, we'd want most of the weights to be exactly zero, not just close to zero. That way, the nonzero weights tell us which features are relevant.

LINEAR REGRESSION : MAP

(19) So rather than having a prior distribution that softly favors zero:



we'd like a prior distribution that strongly favors zero:



(20) So let's assume:

$$P_\epsilon \sim \text{Normal}(0, \sigma^2)$$
$$P(W[d]) \sim \cancel{\text{Normal}(0, \sigma^2)}\ \text{Laplace}(0, b)$$

These choices give us a type of regression called <u>lasso</u>.

Linear Regression: MAP

(21) We computed the point estimate $\hat{w}$ for ridge regression in (10). If we use Laplace instead, we get:

$$
\begin{aligned}
\hat{w} &= \underset{w}{\operatorname{argmax}}\ \ell(w) \\
&= \underset{w}{\operatorname{argmax}}\ \log P(w) + \ell_{MLE}(w) \\
&= \underset{w}{\operatorname{argmax}}\ \log \prod_{d=1}^{D} \frac{1}{2b} \exp\left(\frac{-|w[d]|}{b}\right) + \ell_{MLE}(w) \\
&= \underset{w}{\operatorname{argmax}}\ \sum_{d=1}^{D} \log\left(\frac{1}{2b} \exp\left(-\frac{|w[d]|}{b}\right)\right) + \ell_{MLE}(w) \\
&= \underset{w}{\operatorname{argmax}}\left(\sum_{d=1}^{D} \log\frac{1}{2b} + \log\exp\left(-\frac{|w[d]|}{b}\right)\right) + \ell_{MLE}(w) \\
&= \underset{w}{\operatorname{argmax}}\left(\sum_{d=1}^{D} -\log 2b\right) - \left(\sum_{d=1}^{D} \frac{|w[d]|}{b}\right) + \ell_{MLE}(w) \\
&= \underset{w}{\operatorname{argmax}}\ \cancel{-D\log 2b} - \frac{1}{b}\sum_{d=1}^{D} |w[d]| + \ell_{MLE}(w) \\
&= \underset{w}{\operatorname{argmax}}\ -\frac{1}{b}\sum_{d=1}^{D} |w[d]| + \ell_{MLE}(w)
\end{aligned}
$$

LINEAR REGRESSION : MAP

22) From LINEAR REGRESSION: MLE, 5, we know that:

$$\ell_{MLE}(w) = -\frac{N}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(y-Xw)^T(y-Xw)$$

Plugging this in to what we have so far:

$$\hat{w} = \underset{w}{\operatorname{argmax}} -\frac{1}{b}\sum_{d=1}^{D}|w[d]| - \cancel{\frac{N}{2}\log 2\pi\sigma^2} - \frac{1}{2\sigma^2}(y-Xw)^T(y-Xw)$$

$$= \underset{w}{\operatorname{argmax}} -\frac{1}{b}\sum_{d=1}^{D}|w[d]| - \frac{1}{2\sigma^2}(y-Xw)^T(y-Xw)$$

$$= \underset{w}{\operatorname{argmax}} \left(-\frac{2\sigma^2}{b}\sum_{d=1}^{D}|w[d]|\right) - (y-Xw)^T(y-Xw)$$

$= \ldots$ [see LINEAR REGRESSION: MLE 7]

$$= \underset{w}{\operatorname{argmax}} \left(\frac{-2\sigma^2}{b}\sum_{d=1}^{D}|w[d]|\right) + 2w^TX^Ty - w^TX^TXw$$

$$= \underset{w}{\operatorname{argmax}}\ 2w^TX^Ty - w^TX^TXw - \frac{2\sigma^2}{b}\sum_{d=1}^{D}|w[d]|$$

$$= \underset{w}{\operatorname{argmin}}\ w^TX^TXw - 2w^TX^Ty + \frac{2\sigma^2}{b}\sum_{d=1}^{D}|w[d]|$$

(23) So the loss function for lasso regression is:

$$L_{lasso}(w) = w^T X^T X w - 2w^T X^T y + \frac{2\sigma^2}{b} \sum_{d=1}^{D} |w[d]|$$

$$= L_{lin}(w) + \frac{2\sigma^2}{b} \sum_{d=1}^{D} |w[d]|$$

The expression $\sum_{d=1}^{D} |w[d]|$ is called the <u>L_1-norm</u> of weight vector w:

$$\|w\|_1 = \sum_{d=1}^{D} |w[d]|$$

Thus lasso's loss function is simply the usual linear regression loss function, plus some constant multiple of the L_1-norm of the weight vector:

$$\hat{w} = \underset{w}{\operatorname{argmin}}\ L_{lasso}(w)$$

$$= \underset{w}{\operatorname{argmin}}\ \underbrace{L_{lin}(w)}_{\text{we want the likelihood of the data to be high}} + \underbrace{K \cdot \|w\|_1}_{\text{but we also want the weights to be close to zero}}$$

Lasso is sometimes called <u>linear regression with L_1-regularization</u>.

LINEAR REGRESSION: MAP

(24) In summary, lasso regression estimates the value of an unobserved response variable Y_0 as follows:

(a) compute point estimate $\hat{w} = \underset{w}{\operatorname{argmin}}\, L_{lasso}(w)$

(b) compute $\hat{y}_0 = \hat{w}^T x_0$

(25) To compute $\underset{w}{\operatorname{argmin}}\, L_{lasso}(w)$, compute:

$$\nabla L_{lasso}(w) = \frac{\partial}{\partial w}\left(L_{lin}(w) + K\sum_{d=1}^{D}|w[d]|\right)$$

$$= \frac{\partial}{\partial w} L_{lin}(w) + K \frac{\partial}{\partial w}\sum_{d=1}^{D}|w[d]|$$

$$= \frac{\partial}{\partial w} L_{lin}(w) + K \begin{bmatrix} \frac{\partial}{\partial w[1]} \sum_{d=1}^{D}|w[d]| \\ \vdots \\ \frac{\partial}{\partial w[D]} \sum_{d=1}^{D}|w[d]| \end{bmatrix}$$

$$= \frac{\partial}{\partial w} L_{lin}(w) + K \begin{bmatrix} \operatorname{sgn}(w[1]) \\ \vdots \\ \operatorname{sgn}(w[D]) \end{bmatrix}$$

$$= 2X^TXw - 2X^Ty + K \begin{bmatrix} \operatorname{sgn}(w[1]) \\ \vdots \\ \operatorname{sgn}(w[D]) \end{bmatrix}$$

LINEAR REGRESSION: MAP

(26) Now we could try to find the critical points of $L_{lasso}(w)$ by setting $\nabla L_{lasso}(w) = 0$ and solving for w:

$$2X^T X w - 2X^T y + K \begin{bmatrix} \text{sgn}(w[1]) \\ \vdots \\ \text{sgn}(w[D]) \end{bmatrix} = 0$$

$$w = ???$$

but ick.

(27) Fortunately, we have like 6 variants of gradient descent to choose from, since we can compute $\nabla L_{lasso}(w)$.

Thus we have an algorithm for lasso:

LASSOREGRESSION

- Compute point estimate $\hat{w} = \text{GRADDESCENT}(L_{lasso})$
- Compute estimate $\hat{y}_0 = \hat{w}^T x_0$
- return $\hat{y}_0$