Lecture 3-P1 Regression

Another type of supervised learning, in which

$$h: \mathbb{R}^d \to \mathbb{R}$$

Predict a numerical quantity, like height, stock value, etc.

Need to define a loss function; most common is

probabilistic story

squared error (SE)

$$\text{Loss}(\text{guess}, \text{actual}) = (\text{guess} - \text{actual})^2$$

Different applications might justify other loss functions.

We will stick with linear hypotheses:

$$h(x; \theta, \theta_0) = \theta^T x + \theta_0$$

Data set: $S_n = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

Mean squared error of hypothesis h on S_n: (MSE)

$$\mathcal{E}_n(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} + \theta_0 - y^{(i)}\right)^2$$

Ordinary least-squares regression (OLS):

find θ, θ_0 to minimize $\mathcal{E}_n(\theta, \theta_0)$

~~Assume we have augmented~~ x's with an extra 1 feature, so no θ_0

Analytic solution

Take derivative of $\mathcal{E}_n$ wrt parameters, set to 0, solve for param values. Can do $\partial \mathcal{E}_n / \partial \theta_j$ for all j and solve system of equations. We will take a more compact (and cool!) matrix view.

Recall:

$$X = \begin{bmatrix} x_1^{(1)} & \cdots & x_1^{(n)} \\ \vdots & & \\ x_d^{(1)} & & x_d^{(n)} \end{bmatrix} \quad d \times n$$

each column is an example

$$Y = \begin{bmatrix} y^{(1)} & \cdots & y^{(n)} \end{bmatrix} \quad 1 \times n$$

Warning many books use the transpose of this as X!!

L3-P2

We will define

$$W = X^T = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & & \\ x_1^{(n)} & & x_d^{(n)} \end{bmatrix} \qquad T = Y^T = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(d)} \end{bmatrix}$$

$n \times d$

each row is an example

$$\mathcal{E} = \frac{1}{n}\underbrace{(W\theta - T)^T}_{1\times n}\underbrace{(W\theta - T)}_{n\times 1}$$

$$\nabla_\theta \mathcal{E} = \frac{2}{n}\underbrace{W^T}_{d\times n}\underbrace{(W\theta - T)}_{n\times 1} = 0$$

$$W^T W\theta = W^T T$$

$$\theta = \underbrace{(W^T W)^{-1}}_{d\times d}\underbrace{W^T}_{d\times n}\underbrace{T}_{n\times 1}$$

Closed form!

Regularization

What if $(W^T W)$ is not invertible?

Do ridge regression:

$$J_{ridge}(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^T x^{(i)} + \theta_0) + \lambda \|\theta\|^2$$

not including θ_0

Larger λ values pressure θ values to be near zero

Don't penalize θ_0, in order to let the offset be governed completely by the data.

Solution is a small bit complicated because θ_0 should have special treatment. Result for $\theta_0 = 0$ is:

We won't worry about it.

$$\theta_{ridge} = (W^T W + \lambda I)^{-1} W^T T$$

$$I \to \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & \\ & & \lambda \end{bmatrix}$$

Becomes invertible as λ increases

Demos: regression with polynomial basis features

$\theta^T x + \theta_0$ is a linear function of x

$\theta^T \varphi(x) + \theta_0$ is a non-linear function of x if φ is a non-linear transform

t(order, lambda)

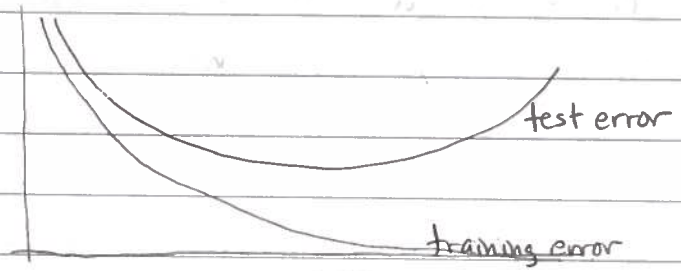
Kinds of error

structural error (bias): cannot represent the desired solution with $\mathcal{H}$: low order, high λ

estimation error (variance): cannot estimate parameters reliably given the available data: high order, low λ

Cartoon of what often happens:

How would you pick λ?



increasing poly order →

decreasing λ →

Gradient descent

Inverting a $d \times d$ matrix $(W^T W)$ is $\sim O(d^3)$ time

Makes analytic solution impractical for large d

Can do gradient descent!

$$J_{ridge}(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^{n} (\theta^T x^{(i)} + \theta_0 - y^{(i)})^2 + \lambda \|\theta\|^2$$

$$\nabla_\theta J = \frac{2}{n} \sum_{i=1}^{n} (\theta^T x^{(i)} + \theta_0 - y^{(i)}) x^{(i)} + 2\lambda\theta$$

$d \times 1$ $\qquad$ $d \times 1$

$$\frac{\partial J}{\partial \theta_0} = \frac{2}{n} \sum_{i=1}^{n} (\theta^T x^{(i)} + \theta_0 - y^{(i)})$$

stochastic gradient descent (SGD)

When gradient is a sum, rather than take one big(ish) step in the direction of the gradient, choose a term of the sum at random and take a small step in that direction.

SGD for regression: ignoring θ_0

$\theta^{(0)} = 0$

for _ = 1 to T

randomly select $i \in \{1, \ldots, n\}$

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k \left((\theta^T x^{(i)} - y^{(i)}) x^{(i)} + \lambda \theta \right)$$

For this to converge to a local optimum as T increases, η has to decrease as a function of time. Rules:

$\sum_{t=1}^{\infty} \eta_t$: infinite $\qquad$ $\sum_{t=1}^{\infty} \eta_t^2$: finite

Legal rule: $\eta_t = 1/t$ but people often play around with it.

Rewrite the sgd step to get insight:

$$\theta^{(k+1)} = (1 - \lambda \eta_k) \theta^{(k)} + \eta_k (y^{(i)} - \theta^T x^{(i)}) x^{(i)}$$

λ pressures θ to be small

error: positive if our prediction is too small

Demos t_gd(order, λ, draw=True, pause=False, step_size=.5)