MLPR Week 1

N*D (# of data point * dimension) **Design matrix**:

with bias, add a column of 1, then bias = last weight

$$\mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}, \quad X = \begin{bmatrix} \mathbf{x}^{(1)\top} \\ \mathbf{x}^{(2)\top} \\ \vdots \\ \mathbf{x}^{(N)\top} \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_D^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_D^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_D^{(N)} \end{bmatrix}.$$

Polynomials:

Not used, since

feature space grows quickly for high-dimensional inputs

Polynomials rapidly take on extreme values as the input x moves away from the origin

Use polynomials: sparse binary features x {0,1}, most are 0

=> no extreme values, x_1*x_2 detects simoteneous features

$$\Phi = \begin{bmatrix} 1 & x^{(1)} & (x^{(1)})^2 & \cdots & (x^{(1)})^{K-1} \\ 1 & x^{(2)} & (x^{(2)})^2 & \cdots & (x^{(2)})^{K-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x^{(N)} & (x^{(N)})^2 & \cdots & (x^{(N)})^{K-1} \end{bmatrix}.$$

$$\boldsymbol{\phi}(\mathbf{x}) = [1 \;\; x_1 \;\; x_2 \;\; x_3 \;\; x_1x_2 \;\; x_1x_3 \;\; x_2x_3 \;\; x_1^2 \;\; \ldots]^\top,$$

$$\boldsymbol{\phi}(\mathbf{x}) = [\phi_1(\mathbf{x}) \;\; \phi_2(\mathbf{x}) \;\; \ldots \;\; \phi_K(\mathbf{x})]^\top$$

add polynomials / **basis function** $f(\mathbf{x}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x})$

RBF:

$$\exp(-(\mathbf{x}-\mathbf{c})^\top(\mathbf{x}-\mathbf{c})/h^2),$$

bell-curve shape centred at c, with ‘bandwidth’ h

logistic-sigmoid: $\sigma(\mathbf{v}^\top\mathbf{x}+b) = \frac{1}{1+\exp(-\mathbf{v}^\top\mathbf{x}-b)}.$

v and b determine the steepness and x axis position

Total square error:

equation & vector form (why square)

$$\sum_{n=1}^{N} \left[y^{(n)} - f(\mathbf{x}^{(n)}; \mathbf{w}, b)\right]^2 = (\mathbf{y}-\mathbf{f})^\top(\mathbf{y}-\mathbf{f}).$$

L2 Regularization: equation & vector form

trade off accuracy

$$E_\lambda(\mathbf{w}; \mathbf{y}, \Phi) = \sum_{n=1}^{N} \left[y^{(n)} - f(\mathbf{x}^{(n)}; \mathbf{w})\right]^2 + \lambda \sum_{k=1}^{K} w_k^2$$

$$= (\mathbf{y}-\Phi\mathbf{w})^\top(\mathbf{y}-\Phi\mathbf{w}) + \lambda \mathbf{w}^\top \mathbf{w}.$$

Matrix trick: use L2 without modification

Φ is N × K, (batch size * dimension), add K rows (appending N)

Ik the K × K identity matrix

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_K \end{bmatrix} \quad \tilde{\Phi} = \begin{bmatrix} \Phi \\ \sqrt{\lambda}\mathbb{I}_K \end{bmatrix},$$

$$\begin{aligned} E(\mathbf{w}; \tilde{\mathbf{y}}, \tilde{\Phi}) &= (\tilde{\mathbf{y}} - \tilde{\Phi}\mathbf{w})^\top (\tilde{\mathbf{y}} - \tilde{\Phi}\mathbf{w}) \\ &= (\mathbf{y} - \Phi\mathbf{w})^\top (\mathbf{y} - \Phi\mathbf{w}) + \lambda \mathbf{w}^\top \mathbf{w} = E_\lambda(\mathbf{w}; \mathbf{y}, \Phi). \end{aligned}$$

Week 2

generalization error: model f(x) on actual distribution p(x,y)

$$\text{Generalization error} = \mathbb{E}_{p(\mathbf{x},y)}[L(y, f(\mathbf{x}))] = \int L(y, f(\mathbf{x}))\, p(\mathbf{x}, y)\, \mathrm{d}\mathbf{x}\, \mathrm{d}y,$$

Monte Carlo estimate:

Assumption (usually wrong): test set contains M samples from distribution

$$\text{Average test error} = \frac{1}{M} \sum_{m=1}^{M} L(y^{(m)}, f(\mathbf{x}^{(m)})), \qquad \mathbf{x}^{(m)}, y^{(m)} \sim p(\mathbf{x}, y).$$

K fold cross validation: for small datasets, but costly

Univariate Gaussians:

PDF: $p(x) = \mathcal{N}(x; 0, 1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}.$

General form: $p(z) = \mathcal{N}(z; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(z-\mu)^2},$

Find “whether distribution is gaussian”:

if 2/3 samples within mean + std

no big outliers

Central Limit Theorem (CLT): if finite mean & variance, result of adding together many random (independent) outcomes is Gaussian

Constrained values: if some values of a sum are impossible (e.g. non-integer), constrained but still Gaussian

Convergence only close to the mean (convergence in distribution): convergence is weak (not guarantee)

Unbiased estimator: mean of XX is the true value

Error bars: plot standard deviation

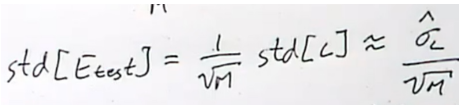
since CLT for large N

fight test set too small

decide if validation/test data is enough

Standard Error of the Mean (SEM): how far the sample mean of the data (test mean) is likely to be from the true population mean (gen mean).

= error bar

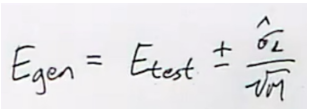


relationship of gen mean & test mean,

if error bar (second term RHS) huge, need bigger validation set

if tiny, need to be sure that this model is going to generalize within similar tiny range if you get more test data from the same distribution

if N very large / small / pretty large: sample mean = normal distributed



report the standard deviation of the models' performances (not a standard error on the mean)

to indicate how much a future fit will typically vary from average performance when something is changed

test model A against model B: find difference in losses on each test case

$$\delta_m = L(y^{(m)}, f(\mathbf{x}^{(m)}; B)) - L(y^{(m)}, f(\mathbf{x}^{(m)}; A)).$$

If the mean of the δ's is several standard errors greater than zero, we would report that A is the better model

Multivariate Gaussians:

vector x, each element sampled from independent Gaussian, PDF =

$$p(\mathbf{x}) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi}} e^{-x_d^2/2} = \frac{1}{(2\pi)^{D/2}} e^{-\frac{1}{2}\sum_d x_d^2}$$
$$= \frac{1}{(2\pi)^{D/2}} e^{-\frac{1}{2}\mathbf{x}^\top \mathbf{x}}.$$

(normalizer, PDF integrates to one)

Covariance,

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \Sigma) = \frac{1}{|\Sigma|^{1/2}(2\pi)^{D/2}} e^{-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{z}-\boldsymbol{\mu})}$$