Generalized RBF n/ws

The 1-1 correspondence of a training sample $\underline{x}_i$ and an associated Green's function $G(\underline{x}, \underline{x}_i)$ can pose many problems

1) It can be prohibitively expensive n/w when N becomes large

2) For determining the unknown coeffts/weights from the hidden to the o/p layer, we need to solve a linear matrix eqn, requiring matrix inverse operations $\sim O(N^3)$

Require : We need an approx. of the regularized soln.

Idea : Approximate the regularized soln. in a lower dim. space using a suboptimal soln

$$F^*(\underline{x}) = \sum_{i=1}^{m_1} w_i \varphi_i(\underline{x})$$

$\{\varphi_i(\underline{x})\}_{i=1}^{m_1}$ is a new set of basis functions

Observe this detail

$$\varphi_i(\underline{x}) = G(\|\underline{x} - \underline{t}_i\|) \; ; \; i = 1, \ldots, m_1$$

@ centers $t_i = \underline{x}_i$

$$F^*(\underline{x}) = \sum_{i=1}^{m_1} w_i \, G(\|\underline{x} - \underline{t}_i\|)$$

where t_i's have to be determined

Let us formulate the functional

$$\mathcal{E}(F^*) = \sum_{i=1}^{N}\left(d_i - \sum_{j=1}^{m_1} w_j \, G\left(\|\underline{x}_i - \underline{t}_j\|\right)\right)^2 + \lambda \|DF^*\|$$

Can be expanded as $\|\underline{d} - G\,\underline{w}\|$

where $\underline{d} = \begin{bmatrix} d_1 & d_2 & \cdots & d_N \end{bmatrix}^T$

$\underline{w} = \begin{bmatrix} w_1 & \cdots & w_{m_1} \end{bmatrix}^T$

Observe this detail

$$G = \begin{bmatrix} G(\underline{x}_1, \underline{t}_1) & \cdots & G(\underline{x}_1, \underline{t}_{m_1}) \\ \vdots & \ddots & \vdots \\ G(\underline{x}_N, \underline{t}_1) & \cdots & G(\underline{x}_N, \underline{t}_{m_1}) \end{bmatrix}_{N \times m_1}$$

G is of size $N \times m_1$ (rectangular matrix)

Evaluating

$$\|DF^*\|^2 = \langle DF^*, (DF^*)^{\dagger} \rangle_{\mathcal{H}}$$

$$= \sum_{i=1}^{m_1} w_i \; G(\underline{x}, \underline{t}_i), \; \tilde{D}D \sum_{i=1}^{m_1} w_i \; G(\underline{x}, \underline{t}_i)$$

$$= \underline{w}^T G_0 \, \underline{w}$$

(Regularization Constraint)

$$G_0 = \begin{bmatrix} G(\underline{t}_1, \underline{t}_1) & \cdots & G(\underline{t}_1, \underline{t}_{m_1}) \\ \vdots & & \vdots \\ G(\underline{t}_{m_1}, \underline{t}_1) & \cdots & G(\underline{t}_{m_1}, \underline{t}_{m_1}) \end{bmatrix}$$

(Square!)

Home Work

The minimization of $\mathcal{E}(F^*)$ w.r.t $\underline{w}$ yields

$$\left(G^T G + \lambda G_0\right) \underline{w} = G^T \underline{d}$$

If $\lambda \to 0$ (i.e., no regularization)

$$\underline{w} = \left(G^T G\right)^{-1} G^T \underline{d}$$

(min. norm soln / least squares / pseudo inverse soln to the problem when $m_1 < N$)

Weighted norm of data points

$$\left\| \underset{(m_0 \times 1)}{\underline{x}} \right\|_C^2 = (Cx)^T Cx = x^T \underbrace{C^T C}_{\substack{m_0 \times m_0 \text{ norm} \\ \text{weighing matrix}}} x$$

$$F^*(\underline{x}) = \sum_{i=1}^{m_1} w_i \, G\left(\| \underline{x} - \underline{t}_i \|_C \right)$$

weighted norm

For the Gaussian case,

$$G\left(\|\underline{x}-\underline{t}_i\|_C\right) = \exp\left(-(\underline{x}-\underline{t}_i)^T \underbrace{\Sigma^{-1}}_{\text{Covariance matrix}} (\underline{x}-\underline{t}_i)\right)$$

$$\Sigma^{-1} \triangleq C^T C$$

XOR Problem Revisited

We will consider RBF n/w as a special case of the Green's n/w.

Consider the pair of Gaussian functions

$$G(\|\underline{x} - \underline{t}_i\|) = \exp\left(-\|\underline{x} - \underline{t}_i\|^2\right)$$

$i = 1, 2$

Let us choose centers @ $\underline{t}_1$ and $\underline{t}_2$

$$\underline{t}_1 = [1 \quad 1]^T \qquad \underline{t}_2 = [0 \quad 0]^T$$

At the moment, forget optimization over $\underline{t}_1, \underline{t}_2$

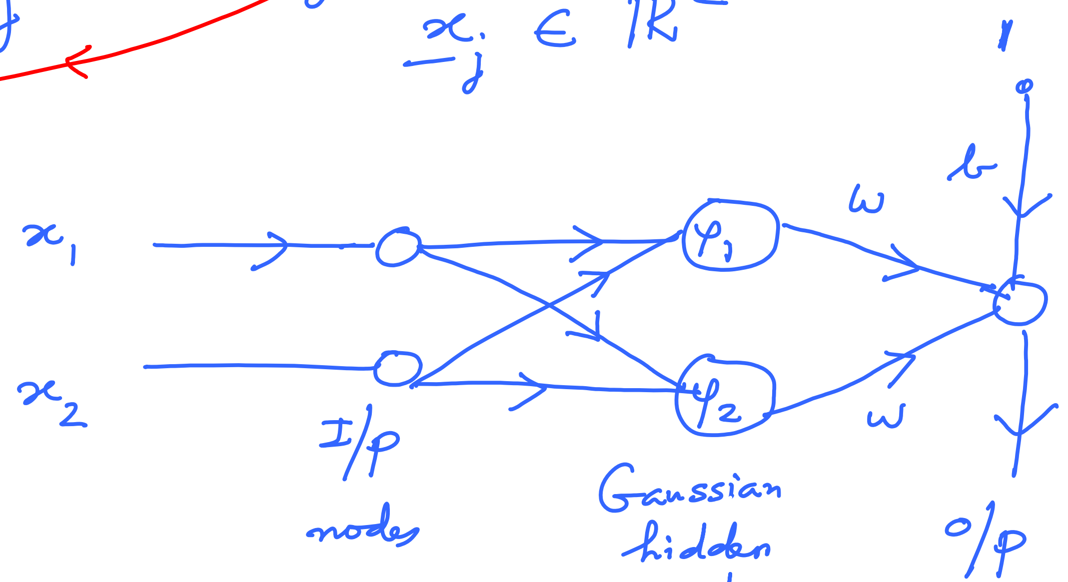
$$y(\underline{x}) = \sum_{i=1}^{2} w\, G(\|\underline{x} - \underline{t}_i\|) + b$$

$i = 1, 2$

$$y(\underline{x}_j) = d_j \qquad j = 1, 2, 3, 4$$

$\underline{x}_j \in \mathbb{R}^2$

$\underline{x}_j$	d_j
(1, 1)	0
(0, 1)	1
(0, 0)	0
(1, 0)	1



Soln to the XOR problem

$$G = \left[\begin{array}{cc:c} 1 & 0.1353 & 1 \\ 0.367 & 0.3678 & 1 \\ 0.1353 & 1 & 1 \\ 0.3678 & 0.3678 & 1 \end{array}\right]_{4\times 3}$$

$$4\times 2$$

4 data points

2 centers $\underline{t}_1$, $\underline{t}_2$

$$\underline{d} = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}^T_{4\times 1}$$

$$\underline{w} = \begin{bmatrix} w & w & b \end{bmatrix}^T_{3\times 1}$$

Solve:

$$\underline{w} = \left(G^T_{3\times 4}\, G_{4\times 3}\right)^{-1} G^T \underline{d}$$

(without regularization)

Plugging G, d into $\underline{w}$ from numerical computations done earlier

$$\underline{w} = \begin{bmatrix} -2.5 \\ -2.5 \\ +2.84 \end{bmatrix}_{3\times 1}$$

Structural risk minimization

Suppose we have the non linear regression model

$$d = f(\underline{x}) + \varepsilon$$

$f(\cdot)$ is unknown

Consider the ensembled-averaged cost

$$J_{act}(f) = \underbrace{E_{x,d}}_{\text{joint expectation}}\left(\frac{1}{2}\left(d - f(\underline{x})\right)^2\right)$$

$$\hat{f}^* = E(d \mid \underline{x}) \text{ minimizes } J_{act}(\hat{f})$$

$\hat{f}^*$ requires the knowledge of joint pdf of $\underline{x}$ and d

Suppose we bring in a neural network and make a first approximation

$$f(\underline{x}) \cong F(\underline{x}; \underline{w})$$

$$J(\underline{w}) = E_{x,d}\left[\frac{1}{2}\left(d - \underbrace{F(\underline{x}; \underline{w})}_{\text{from a neural network}}\right)^2\right]$$

Let $\underline{\hat{w}}^* = \underset{\underline{w}}{\arg\min}\ J(\underline{w})$

$$\boxed{J(\hat{\underline{w}}^*) \geq J_{act}(\hat{f}^*)} \quad \text{———} \quad (1)$$

This is the 1st level of approximation

Consider the time averaged energy function

$$\mathcal{E}_{av}(N; \underline{w}) = \frac{1}{2N} \sum_{i=1}^{N} \left(d(i) - F(\underline{x}(i); \underline{w})\right)^2$$

The minimizer of $\mathcal{E}_{av}(N; \underline{w})$ is $\hat{\underline{w}}_N$

$$\hat{\underline{w}}_N = \underset{\underline{w}}{\arg\min}\ \mathcal{E}_{av}(N; \underline{w})$$

$$J(\underline{\hat{w}}_N) \geq J(\underline{\hat{w}}^*) \geq J_{act}(\hat{f}^*)$$

time averaged cost opt. | over $E(\cdot)$ | conditional mean

Excess error:

$$J(\underline{\hat{w}}_N) - J_{act}(\hat{f}^*) = \underbrace{J(\underline{\hat{w}}_N) - J(\underline{\hat{w}}^*)}_{\text{depends on the size of the data } N} + \underbrace{J(\underline{\hat{w}}^*) - J_{act}(\hat{f}^*)}_{\text{app. error}}$$

(NN model)

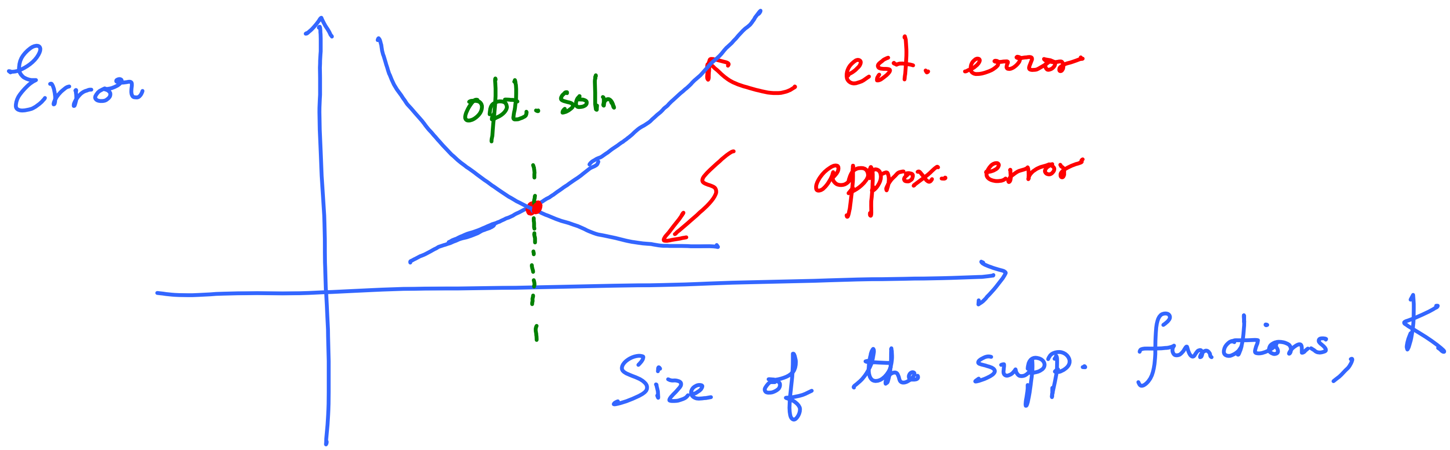
For e.g., for a single hidden layer MLP,
the capacity of the learning machine
is governed by the size of the hidden layer

Consider a family of nested approximating functions

$$\mathcal{F}_k = \left\{ F(\underline{x}; \underline{w}) \quad \underline{w} \in \mathcal{W}_k \right\}$$

such that $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \cdots \subset \mathcal{F}_K$

$\mathcal{F}_k$ is a measure of the machine capacity



1) Before opt. is reached, the machine capacity is too small for the details within the data

2) After opt. is reached, the machine capacity is too large for the details within the data

Bias - Variance Dilemma

$y = f(\mathbf{x})$ ← vector

'scalar' in label associations

Consider the functional approximation problem.

We have a data set D and an associated mapping $f : X \to Y$

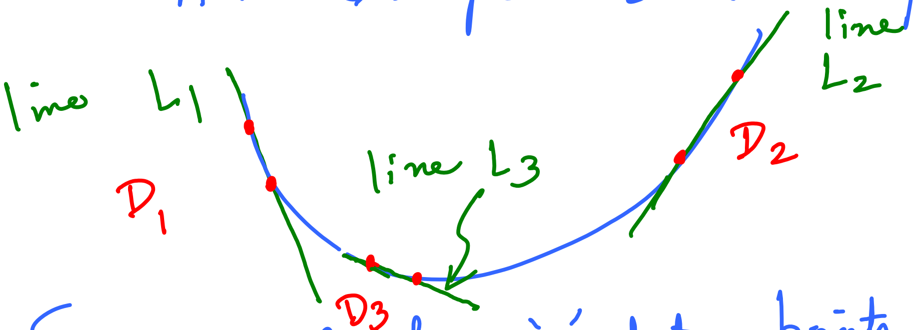
'f' is unknown here!

We need to get a good estimate of 'f' from the data set D., get $g_D(x)$ close to f in some sense

Also, typically, one can have several data sets in the learning example. Given different sets $\{D_i\}_{i=1}^{N}$,

One can arrive at various estimates of 'f'

An example shall help us visualize



Suppose I have '2' data points from a parabola above.
If I need to fit a line i.e., $y = mx + c$ form

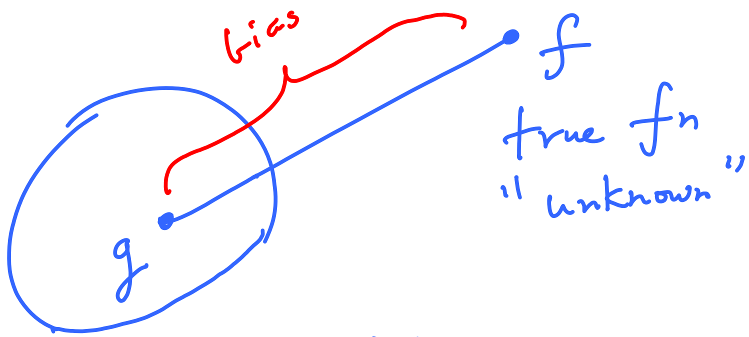
Given D_1, I can get L_1

D_2, —"— L_2

⋮ ⋮

Qn: What if I just want to approximate the parabola by just a scalar? say, c

Intuition

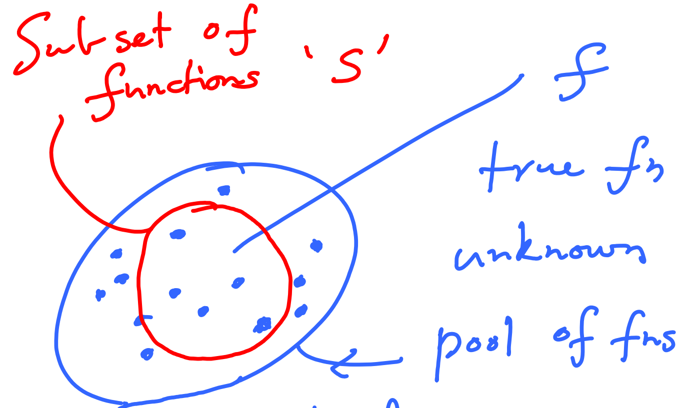


Space of functions having just one function 'g'

$g \approx f$

$\Longrightarrow$ There is bias $g(x) - f(x)$

But no variance since we have just one function



We have a pool of functions $\{g_i\}_{i=1}^{N}$

$\{g_i(x)\}$ agree with f on the training data sets $\{D_i\}$

The $\langle \{g_j\}_{j \in S} \rangle \approx f$

$\Longrightarrow$ Bias is $<<$ (much less)

But the Variance is more since we have > 1 functions

Having seen that there is 'bias' and 'variance' in the error averaged over the data sets corresponding to the choice in the pool of functions available, this gives rise to a trade off in the bias & variance given the generalization problem

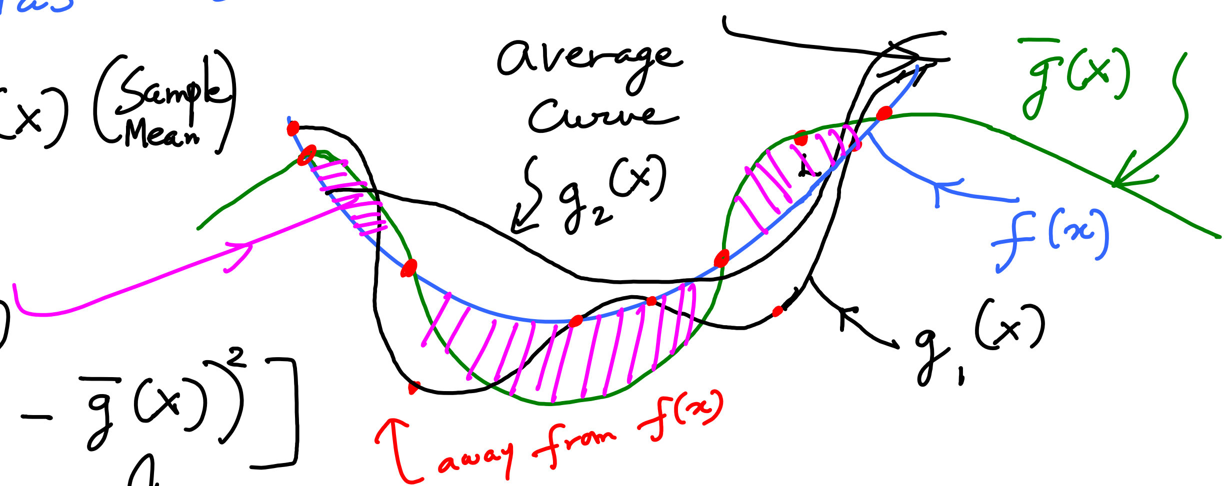
$\Rightarrow$ Bias - Variance dilemma

Role of bias/variance

Suppose we have a higher model complexity (due to fitting noisy samples)

$\Rightarrow$ Bias is less but variance is more

$$\bar{g}(x) = \frac{1}{N}\sum_{i=1}^{N} g_i(x) \quad \text{(Sample Mean)}$$



Higher order curve over fits the data than required

$$\text{bias}(x) = \bar{g}(x) - f(x)$$

$$\text{var}(x) = E_{D|x}\left[\left(g_D(x) - \bar{g}(x)\right)^2\right]$$

Each 'g_D' corresponds to a 'D'

Imagine several such data sets over which we compute our statistics

Let us work out the analysis

$$\text{Let} \quad y = f(X) + \varepsilon \qquad \text{(A)}$$

unknown relationship between X and y

regression noise is a random variable with mean '0' and variance σ^2 & statistically indep. of f and approximating function $g_D(X)$ for a data set D

We are interested in

$$\mathop{E}_{D,X}\left[\underbrace{\left[\left(g_D(X) - y\right)^2\right]}_{\text{function 'D'}}\right] \qquad \text{(B)}$$

over all data sets

Plugging (A) in (B)

$$E_{D,x}\left[\left(g_D(x) - \underbrace{f(x) - \varepsilon}_{y = f(x) + \varepsilon}\right)^2\right]$$

Exchanging expectations 'assuming' it can be done (sums over finite data sets)

$$E_x\left[E_{D|x}\left[\underbrace{\left(g_D(x) - f(x) - \varepsilon\right)^2}\right]\right]$$

Let us expand the terms

$$E_X\left[E_{D|x}\left[g_D^2(x) + f^2(x) + \varepsilon^2 - 2\,g_D(x)\,f(x) - 2\,g_D(x)\,\varepsilon + 2\,f(x)\,\varepsilon\right]\right]$$

$\underline{\text{I}}$

Since expectation is linear, we shall evaluate some of terms towards simplification

$$E_{D|x}\left[f^2(x)\right] = f^2(x)$$

$$E_{D|x}\left[\varepsilon^2\right] = \sigma_x^2$$

$$E_{D|x}\left[g_D(x)\,\varepsilon\right] = 0$$

($\because$ Statistically independent & '0' mean for noise)

$|||^{ly}$ $E_{D|x}\left[f(x)\, \varepsilon \right] = 0$ (Same reason as earlier)

Define $E_{D|x}\left[g_D(x) \right] \triangleq \bar{g}(x)$

Let us simplify (I)

$$E_x \Bigg[\underbrace{E_{D|x}\left(g_D^2(x) \right) - \bar{g}^2(x)}_{\text{Term I}} + \overbrace{\underbrace{\bar{g}^2(x) + f^2(x) - 2\bar{g}(x) f(x)}_{\text{add \& subtract}}}^{\text{Term II}} + \sigma_x^2 \Bigg]$$

$$E_{D|x}\left(g_D^2(x) \right) - \bar{g}^2(x) = Var(x)$$

$$E_x \left[\bar{g}^2(x) - 2\bar{g}(x) f(x) + f^2(x) \right] = E_x \left[\underbrace{\bar{g}(x) - f(x)}_{bias(x)} \right]^2$$

$$= \text{Variance} + \text{bias} + \sigma^2 \leftarrow E_{x,D}(\varepsilon^2) = \sigma^2$$

(Regression Problem)

Estimation of regularization parameter

Consider the N.L. reg. problem

$$d_i = f(\underline{x}_i) + \varepsilon_i \quad ; \quad i = 1, \ldots, N$$

$f(\cdot)$ is unknown

ε_i is drawn from a zero mean white process

with
$$E(\varepsilon_i \varepsilon_k) = \begin{cases} \sigma^2 & i = k \\ 0 & \text{else} \end{cases}$$

GOAL : Recover $f(\underline{x}_i)$ given $\left\{(\underline{x}_i, d_i)\right\}_{i=1}^{N}$

Let $F_\lambda(\underline{x})$ be the regularized estimate of $f(x)$ for some regularization parameter λ

$$\underbrace{\mathcal{E}(F)}_{\text{Tikhonov functional}} = \underbrace{\frac{1}{2}\sum_{i=1}^{N}\left(d_i - F(\underline{x}_i)\right)^2}_{\text{fidelity to data}} + \frac{\lambda}{2}\underbrace{\left\| D F(\underline{x})\right\|^2}_{\text{Smoothness constraint}}$$

Averaged Square error

Let $R(\lambda)$ denote the averaged square error over a given data between $f(\underline{x})$ pertaining to the model and the approximating fn $F_\lambda(\underline{x})$ pertaining to the representation of the soln for some λ over the training data

$$R(\lambda) = \frac{1}{N} \sum_{i=1}^{N} \left(f(\underline{x}_i) - F_\lambda(\underline{x}_i) \right)^2$$

$$F_\lambda(\underline{x}_k) = \sum_{i=1}^{N} a_{ki}(\lambda)\, d_i$$

(Linear combination)

Observe the detail

pertaining to the data point $\underline{x}_k$

$$\underline{F}_\lambda = A(\lambda)\, \underline{d}$$

$$\underline{F}_\lambda = \left[F_\lambda(\underline{x}_1) \quad \cdots \quad F_\lambda(\underline{x}_N) \right]^T$$

$$A(\lambda) = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ \vdots & & \ddots & \vdots \\ a_{N1} & & & a_{NN} \end{bmatrix}$$

influence matrix

$$\underline{d} = \left[d_1 \quad \cdots \quad d_N \right]^T$$

$$R(\lambda) = \frac{1}{N} \left\| \underline{f} - \underline{F}_\lambda \right\|^2$$

($\|\cdot\|$ is L_2-norm)

$$\underline{f} = \left[f(x_1) \quad \cdots \quad f(x_N) \right]^T$$

Simplifying,

$$R(\lambda) = \frac{1}{N} \left\| \underline{f} - A(\lambda)\,\underline{d} \right\|^2 \qquad (1)$$

$$\underline{d} = \underline{f} + \underline{\varepsilon} \qquad (2)$$

$$\underline{\varepsilon} = \left[\varepsilon_1 \quad \cdots \quad \varepsilon_N \right]^T$$

Plug (2) in (1)

$$R(\lambda) = \frac{1}{N} \left\| \underline{f} - A(\lambda)(\underline{f} + \underline{\varepsilon}) \right\|^2$$

$$= \frac{1}{N} \left\| \underline{f} - A(\lambda)\underline{f} - A(\lambda)\underline{\varepsilon} \right\|^2$$

$$= \frac{1}{N} \left\| (I - A(\lambda))\underline{f} - A(\lambda)\underline{\varepsilon} \right\|^2 \qquad 3$$

Let us expand 3

$$R(\lambda) = \overbrace{\frac{1}{N}\left\|(I - A(\lambda))\underline{f}\right\|^2}^{\text{Term 1}} \underbrace{- \frac{2}{N}\varepsilon^T A^T(\lambda)(I - A(\lambda))\underline{f}}_{\text{Middle Term}} + \underbrace{\frac{1}{N}\left\|A(\lambda)\underline{\varepsilon}\right\|^2}_{\text{Term 2}}$$

We need $E(R(\lambda))$ $\quad\left(E(\text{Middle Term}) = 0\right)$

$$E\left(\frac{1}{N}\left\|(I - A(\lambda))\underline{f}\right\|^2\right) = \frac{1}{N}\left\|(I - A(\lambda))\underline{f}\right\|^2$$

Consider $E\left(\|A(\lambda)\,\underline{\varepsilon}\|^2\right)$

$$= E\left[\underline{\varepsilon}^T A^T(\lambda)\; A(\lambda)\,\underline{\varepsilon}\right]$$

$$= tr\left[E\left(\underline{\varepsilon}^T A^T(\lambda)\, A(\lambda)\,\underline{\varepsilon}\right)\right] \quad \left(\because tr(\text{scalar}) = \text{scalar}\right)$$

$$= E\left[tr\left(\underline{\varepsilon}^T A^T(\lambda)\; A(\lambda)\,\underline{\varepsilon}\right)\right] \quad \left(\because \text{exchanging } tr(\cdot) \text{ \& } E(\cdot)\right)$$

$$= E\left[tr\left(A^T(\lambda)\, A(\lambda)\, \underline{\varepsilon}\, \underline{\varepsilon}^T \right) \right] \quad \left(\because tr(AB) = tr(BA) \right)$$

$$= tr\left(A^T(\lambda)\, A(\lambda) \right) E\left[\underline{\varepsilon}\, \underline{\varepsilon}^T \right]$$

$$= \sigma^2\, tr\left(A^T(\lambda)\, A(\lambda) \right)$$

$$E\left(\| A(\lambda)\, \underline{\varepsilon} \|^2 \right) = \sigma^2\, tr\left(A^T(\lambda)\, A(\lambda) \right)$$

$$E(R(\lambda)) = \frac{1}{N} \left\| (I - A(\lambda)) \underline{f} \right\|^2 + \frac{\sigma^2}{N} \operatorname{tr} \left\| A^T(\lambda) A(\lambda) \right\|$$

But we still have a problem!

$E(R(\lambda))$ is still a fn of $f(\cdot)$ which is unknown!

A reasonable estimate of $\underbrace{\hat{R}(\lambda)}_{}$ is given by

$E(R(\lambda))$

$$\underbrace{\hat{R}(\lambda)}_{\text{depends on } \lambda} = \frac{1}{N} \left\| \left(I - A(\lambda) \right) \underline{d} \right\|^2 + \underbrace{\frac{\sigma^2}{N} \operatorname{tr}\left(A^2(\lambda) \right) - \frac{\sigma^2}{N} \operatorname{tr}\left(\left(I - A(\lambda) \right)^2 \right)}_{\text{to make the estimate unbiased}}$$