Ordinary least squares.

$X \in \mathbb{R}^d$, $Y \in \mathbb{R}$, $\phi()$ is a kernel map from $\mathbb{R}^d \to \mathbb{R}^m$, $m \le d$. $\phi = (\phi_0, \phi_1, \ldots \phi_{M-1})$.

$h(x_n) = W^T \phi(x) + w_0$ — Generalized linear model.

$W_0 \to$ bias term

$W \to [w_0 \ldots w_{M-1}]^T$

Eg ϕ:

$\phi_1 = \{1, x, x^2\}$

$\phi_2 = \exp\left\{\frac{(x-\mu_j)^2}{2s^2}\right\}$

Given $D = \{x_1, x_2 \ldots x_n\}$

$Y = [y_1, y_2, \ldots, y_n]$

Construct a design matrix ϕ: $\phi_{nj} = \phi_j(x_n)$.

$$\phi = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \ldots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & & & \vdots \\ \vdots & & & \\ \phi_0(x_n) & \ldots & \ldots & \phi_{M-1}(x_n) \end{bmatrix}_{N \times M}$$

Objective of OLS: find W s.t

$\cancel{L = \| Y - \phi_{N\times M} W_{M\times 1} \|}$

$L = \| Y^T - W^T_{1\times M} \phi^T_{M\times N} \|_2^2$ is minimized.

(1123)

$$W^*_{OLS} = \arg\min \|Y - W^T\phi^T\|_2^2$$

$$= [Y - W^T\phi^T]^T [Y - W^T\phi^T]$$

$$\frac{\partial L}{\partial W} = 0 \Rightarrow W^*_{OLS} = \underbrace{(\phi^T\phi)^{-1}\phi^T}_{\text{Moore-Penrose inverse.}} Y.$$

$$= \phi^+ Y.$$ [Projection of Y on to the space spanned by ϕ].

The ML interpretation of the OLS.

We saw that under the L_2 loss the optimal function is the conditional expectation.

Thus, for any data, the optimal

$$h(x) = E[Y|x].$$

Let us assume the

$$f(y|x) \sim N(h(x), \sigma).$$

Thus, ~~for~~ h(x) can be found using ML estimate of this conditional distribution.

$$h(x) = W^T \phi$$

Now, Likelihood for the above distribution is

$$L = \prod_{i=1}^{n} N(h(x_i), \sigma)$$

$$\log L = \sum_{i=1}^{n} \ln N(y_i | W^T \phi(x_i), \sigma)$$

$$= \frac{n}{2}(\;) - \frac{n}{n}\ln(2\pi) - \frac{1}{\sigma}\sum_{i=1}^{n}(y_i - W^T\phi(x_i))^2$$

Thus, $$W^*_{ML} = \underset{W}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - W^T \phi(x_i))^2$$

$\Rightarrow W^*_{OLS} = W^*_{ML}$ with the Gaussian assuption for the posterior.

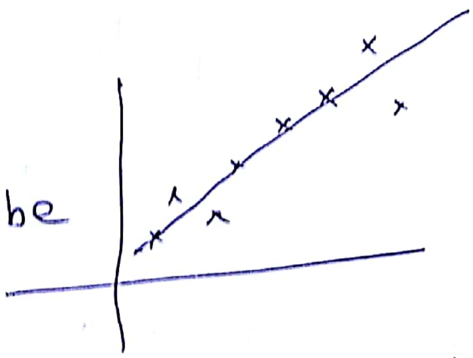
(115).

The idea of regularization.

W_{OLS} can also be viewed as modelling (minimizing) the noise added to the approximator to get to the actual output.

$$y = h(x) + \epsilon \qquad y|x \sim N(h(x), (1)).$$

$$\epsilon \sim N(0, \sigma).$$



Then W_{OLS} or W_{MLE} can be interpretted as finding the parameters of $h(x)$ that would minimize the perturbed & the actual data.

Or thee estimating conditional mean of $Y|x$.

Regularization.

We know that the risk of a classifier

$$R(h) = \underset{P_{XY}}{E}[L]$$

Assuming squared error loss,

$$R(h) = \iint (h(x) - y)^2 f_{xy}(x,y)\, dx\, dy.$$

Consider $\{h(x) - y\}^2 = \{h(x) - E[y|x] + E[y|x] - y\}^2$

$$= \{h(x) - E[y|x]\}^2 + 2\{h(x) - E[y|x]\}\{E[y|x] - y\} + \{E[y|x] - y\}^2.$$

Consider the cross term in the risk integral.

$$2\iint \{h(x) - \overbrace{E[y|x]}^{\text{fn of } x \text{ alone}}\}\{E[y|x] - y\}\, f(x,y)\, dy\, dx$$

$$= \int 2\{h(x) - E[y|x]\} \int (E[y|x] - y) f_{xy}(x,y)\, dy\, dx$$

$$(117). = (\ )[E[y|x] - E[y|x]] = \underline{0}$$

$$\therefore R(h) = \int \underbrace{\{h(x) - E[Y|x]\}^2}_{(1)} f_X(x)\, dx$$

$$+ \int \underbrace{\{E[Y|x] - Y\}^2}_{(2)} f_X(x)\, dx.$$

Observe that only (1) can be tweaked/learned algorithmically.

Nothing can be done about the term (2).

(2) actually signifies the error (noise) that is in the data - could be feature noise | label noise etc.

Thus, risk can be minimized only upto a factor given by term(2) [usually unknown].

Thus, Let us focus on term (1).

[observe that risk minimizes when term(1) is minimized @ $h(x) = E[Y|x]$ .. another proof].
(118).

Now, Risk term of importance

$$R_1(h) = \int \{h(x) - E[Y|x]\}^2 f_X(x)\, dx.$$

Observe that R_1 is an expectation w.r.t $f_X(x)$

Roughly, it is a measure of the 'error' incurred by the classifier $h(x)$ averaged over all possible datasets sampeled from $f_X(x)$.

In that sense, $R_1(h) = \mathbb{E}_D[\text{error}] = \mathbb{E}_D[h(x) - E(Y|x$
[expected error over all possible choices of datasets]

Now, $R_1 = \int \{h(x) - E[Y|x]\}^2 f_X(x)\, dx$

Consider $\{h(x) - E[Y|x]\} = \{h(x) - \mathbb{E}_D[h(x)] + \mathbb{E}_D[h(x)] - E[Y|x]\}$

$$\mathbb{E}_D = \int h(x) f_X(x)\, dx$$

[Average ~~perform.~~ classifier over all datasets].

Plugging the above to R_1 & evaluating the integral,

$$R_1(h) = \int \overbrace{\left[E_D(h(x)) - \underset{E[y|x]}{E(y|x)}\right]^2 f(x)\,dx}\ - \underline{\underline{\text{Bias}^2}}$$

$$+ \int E_D\left[\left\{h(x) - E_D(h(x))\right\}^2\right] f(x)\,dx.$$

— Variance

Thus, Risk = Bias^2 + variance + noise (irreducible).

Bias — The extent to which the average approximator ($h(x)$) (over all data) differs from the desired output. ($E[y|x]$)

Variance — The extent to which the approximator $h(x)$ is sensitive to the choice of a partiallar dataset.

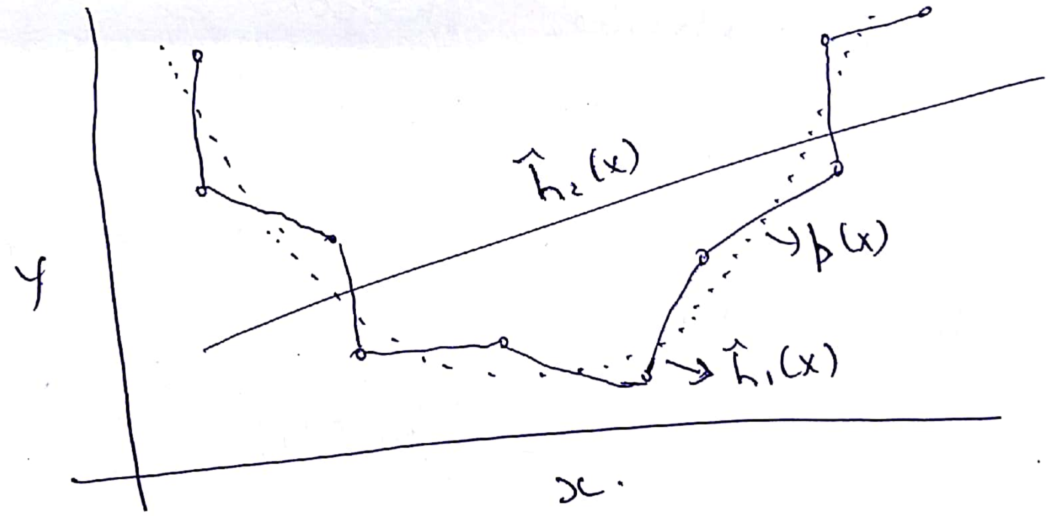
A Good ML algorithm — reduce the error.
Reduce both the bias & variance.

Behaviour of Bias & variance.

Let us consider a regression problem, with y lying on a smooth parabola.

Data is generated by perturbing ~~poi~~ target points with a small noise.

$$Y_i = f(x_i) + \epsilon_i \qquad f(x) = ax^2 + c$$



The optimal mapping (regressor) would be $E[y|x]$ but

let us choose two extreme cases of classifiers (regressors) & see what happens to Bias & variance.

(121).

Scanned by CamScanner

Let $\hat{h}_1(x)$ be a line choosen independently of data.

Consider

$$\text{Variance}(\hat{h}_1(x)) = \mathcal{E}\int \hat{h}_1(x) f(x)\, dx$$

$$= \hat{h}_1(x) \int f(x)\, dx = \hat{h}(x).$$

$$\therefore \quad h(x) = E_D(h(x)) \Rightarrow \underline{\text{Variance} = 0.}$$

However the bias will be typically large. [underfitting]

In the other extreme, choose

$$\hat{h}_2(x_i) = p(x_i) + \epsilon_i$$

In this case bias = 0. [over the observed data]

$$\therefore \quad E_D\left[\hat{h}_2(x)\right] = E_D\left[p(x) + \epsilon\right]$$

$$= E_D\left[p(x)\right] = E[Y|x]$$

[over the data).

However variance :

$$E_D\left[\epsilon^2\right]$$ – variance of the noise which is typically high.

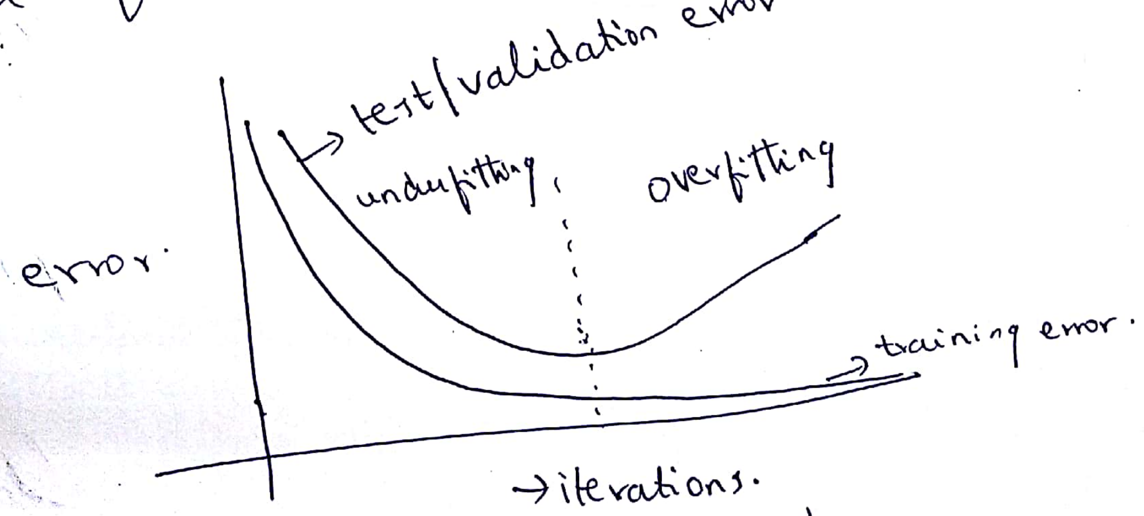
[verify the algebra] [overfitting].

Thus, ~~tht~~ One cannot simultaneously minimize bias & variance together - usually lowering one results in increasing the other.

Thumb rule: h(x) that closely fits the given data (training data) - Low bias | high variance (overfitting).

variance can be lowered by smoothing h(x), but taken too far leads to high bias & low - variance [underfitting].

Question: How to minimize both of them?
Central question in ML.



Typical training graph.

Many possible ways.

1. Given very large dataset, use high-capacity models reducing bias, see 'enough' samples to reduce the variance. [DNN idea].

2. Use prior knowledge on true labels to constrain models [so that bias is not too much]

Bias minds take better decisions !!!

often termed as <u>Regularization</u>.

We'll look at one strategy for <u>Reg</u> in this course : <u>Parameter</u> norm <u>penalties</u>.

Regularized least Squares.

<u>Idea:</u> Real life data is sparse (occupies - a lower dim manifold in the data space).

In such cases, complex models [with higher bias] can fit the data well.

In other words, the error is dominated by the variance component.

Thus, intuitively it is better to trade bias for lower variance.

[Example of fitting a sin ~~with~~ noise by a very high degree polynomial].
Power series of sin → monotonic x.

Lower variance ≡ restricting the value of the parameter space.

One way to achieve that is to add a penalty term on the parameter in the Least square cost.

$$L = \|y - W^T\phi(x)\| + \frac{\lambda}{2} W^T W$$

we want W to be small.

$\lambda \rightarrow$ regularization constant.

$$W^*_{reg} = (\lambda I + \phi^T\phi)^{-1}\phi^T y.$$