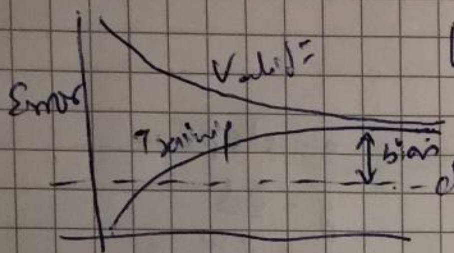
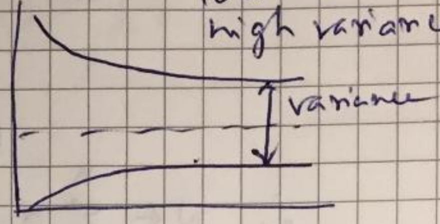
CS4780 Lec 22 Kernels

Combating High Bias — high Training error

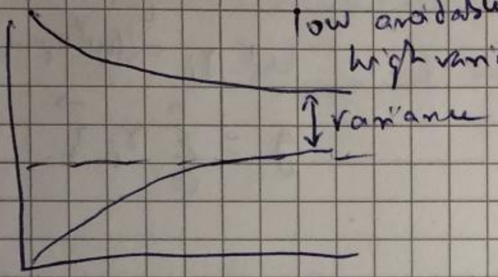


high bias
low var.

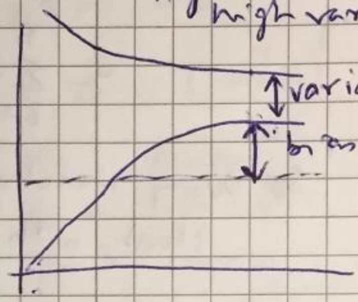
low bias
high variance



low avoidable bias
high variance



high bias (avoidable)
high variance



High bias — add features by modifying them
by adding interaction bet^n them

$$x \qquad \phi(x)$$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_1 x_2 \\ x_2 x_3 \\ x_1 x_3 \\ x_1 x_2 x_3 \end{pmatrix}$$

d

$\in \mathbb{R}^{(2^d)}$ blows up dimensionality

Linear classifiers have bias problem as they can only learn linear boundaries

One could do,

$K_{ij} = x_i^T x_j$ (inner products) precompute & store

$$l(w) = \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

$$\frac{\partial l}{\partial w} = \sum_{i=1}^{n} 2(w^T x_i - y_i)\, x_i$$

let's assume $w = \sum_{i=1}^{n} \alpha_i x_i$ then, $w^T x_j = \sum_{i=1}^{n} \alpha_i \underbrace{x_i^T x_j}_{K_{ij}}$

w a linear comb^n of x's

Proof by Induction:

Initially $w_0 = \vec{0}$ $\quad \alpha_i = 0 \ \forall i$

At any time in updates

$$w = \sum_{i=1}^{n} \alpha_i \vec{x_i}$$

then $w \leftarrow w - s\vec{g}$

s = stepsize

$$g = \frac{\partial L}{\partial w} = \sum_{i=1}^{n} 2(w^T x_i - y_i) x_i$$

Let $\gamma_i = 2(w^T x_i - y_i)$

$$g = \sum_{i=1}^{n} \gamma_i \vec{x_i}$$

$$\therefore \ w \leftarrow \sum_{i=1}^{n} \alpha_i x_i - s \sum_{i=1}^{n} \gamma_i x_i$$

$$= \sum_{i=1}^{n} (\alpha_i - s\gamma_i) x_i$$

So During iteration

start with $\alpha = 0 \ \forall i$

compute γ_i

$\alpha_i \leftarrow \alpha_i - s\gamma_i$

Only need to store n αs and not $\mathbb{R}^d$ w's

Testing: $h(x) = w^T x = \sum_{i=1}^{n} \alpha_i x_i^T x$ $\leftarrow$ no need to compute w's

$$K_{ij} = x_i^T x_j$$

"Inner product" in high dim space

$$k_{ij} = \phi(x)^T \phi(z)$$

$$= 1 + x_1 z_1 + x_2 z_2 + \dots x_d z_d + \dots + x_1 \dots x_d z_1 \dots z_d$$

$$= \prod_{k=1}^{d} (1 + x_k z_k) \rightarrow \text{written as } K(x, z)$$

let's check for $k = 2$

$$(1 + x_1 z_1)(1 + x_2 z_2) = 1 + x_1 z_1 + x_2 z_2 + x_1 x_2 z_1 z_2$$

loss function:

$$l(w) = \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

$$= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \alpha_j \, K(x_i, x_j) - y_i \right)^2$$

gradient w.r.t α

Inner product functions:

Linear kernel function $K(x, z) = x^T z$

Polynomial kernel $K(x, z) = (1 + x^T z)^p$

$p = 1$ linear
$p = 2$ quadratic
$p = 3$ cubic

Radial Basis Funct. kernel $K(x, z) = e^{-\frac{|x - z|^2}{\sigma^2}}$

— universal approximator
Any function can be closely approximated given a few assumptions
Every problem is linearly problem separable, provided one doesn't have 2 identical points.

RBF – takes data set and puts many many gaussian around every single point

Imp:- K has to be +ve semidefinite – iff $q^T K q \geq 0$

$\Leftrightarrow K = z^T z$

factorized

K eigen values are ≥ 0, real, symmetric