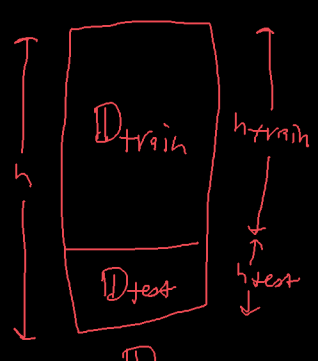


$$\frac{1}{K} = \frac{n_{test}}{n} \Rightarrow K = \frac{n}{n_{test}}$$

$$n_{test} \in \left\{1, 2, \ldots \tfrac{n}{2}, \ldots \tfrac{n}{3}, \ldots \tfrac{n}{5}, \ldots \tfrac{n}{10}, \ldots, n-1\right\}$$

$$\Rightarrow K \in \left\{n, \tfrac{n}{2}, \ldots 2, \ldots 3, \ldots 5, \ldots 10, \ldots \tfrac{n}{n-1}\right\}$$

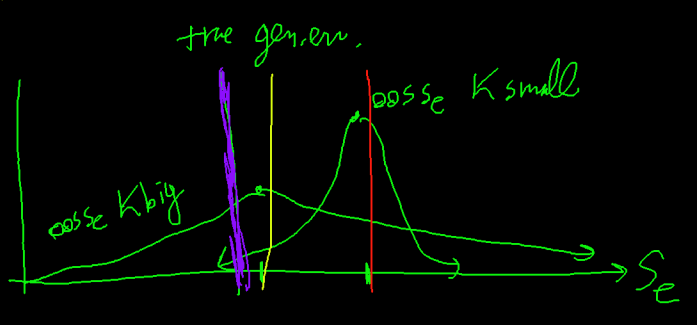
most common choices for K

$\mathbb{D}$ — only a sample of the full distr. $\langle \vec{X}, Y \rangle$

What's the tradeoff?

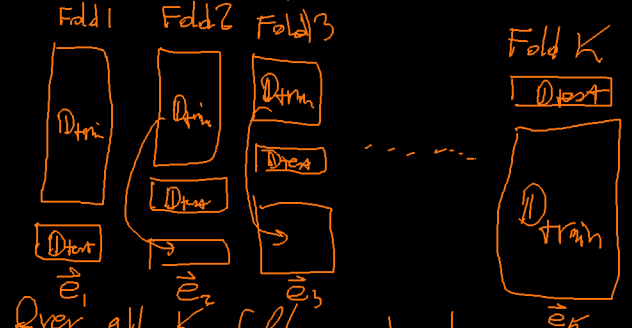
We want honest validation. We use $\mathbb{D}_{test}$ to estimate "generalization error" AKA model performance in the future.

If n_{test} is small, the estimate is very variable but... n_{train} is big thus... my expected estimate is closer to the true performe of g_{final}.



Cross Validation (CV): do many train-test splits.

K-Fold CV → Another (Later)



Over all K folds, each observation is used once.

Let $\vec{e}_{cv} = \begin{bmatrix} \vec{e}_1 \\ \vec{e}_2 \\ \vdots \\ \vec{e}_K \end{bmatrix}$ has length n. Now, compute metrics on $\vec{e}_{cv}$.

If $K = n$ (n folds) AKA "leave-one-out cross validation (LOOCV)"

Consider computing metrics on $\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_K \Rightarrow s_{e_1}, s_{e_2}, \ldots, s_{e_K}$

$\bar{s}_e = \text{Avg}(s_e\text{'s})$, $s_{s_e} = \text{stdev}(s_e\text{'s})$ $CI_{\text{valid}, 95\%} = [\bar{s}_e \pm 2 s_{s_e}]$

For this to be valid, $s_e \overset{iid}{\sim} N(\,,\,)$. Is this true? Independent? NO

K large → normality from CLT.

Because $\vec{e}$'s are dependent since they are functions of g's which are dependent via sharing $\mathbb{D}_{train}$'s.

Computing a valid CI for generalization error under general conditions is NOT possible.

Given $\mathbb{D}$, you have many choices of models because there are many $\mathcal{A}$'s and many $\mathcal{H}$'s. $g = \mathcal{A}(\mathbb{D}, \mathcal{H})$

Let's say you have a choice of a finite # of models, M:

How do we choose between $g_1, g_2, \ldots, g_M$? e.g. $p_{raw} = 1$. $\mathcal{A} = OLS$, $y = \mathbb{R}$

$g_1 = b_0 + b_1 x$

$g_2 = b_0 + b_1 x + b_2 x^2$

$g_3 = b_0 + b_1 \ln(x)$

$g_4 = b_0 + b_1 \mathbb{1}_{x \in [0,1]} + b_2 \mathbb{1}_{x \in [1,2]} + \ldots$

⋮

This is the fundamental problem of "Model Selection".

In this class, we will provide one idea. It is not the only idea.

g_1 is fit with $\mathcal{A}_1, \mathcal{H}_1$ on $\mathbb{D}_{train}$ and s_{e_1} is computed on $\mathbb{D}_{test}$

g_2 // // // $\mathcal{A}_2, \mathcal{H}_2$ // // // s_{e_2} // // //

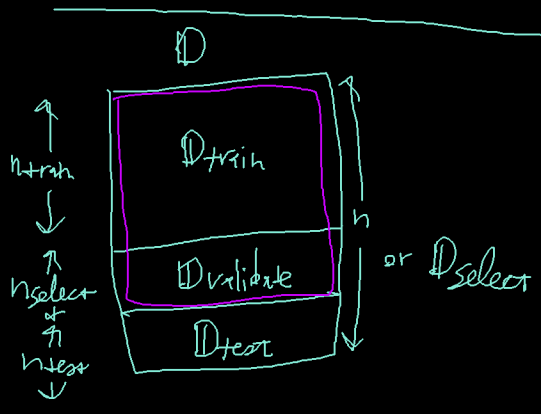
⋮

g_m // // // $\mathcal{A}_m, \mathcal{H}_m$ // // // s_{e_m} // // //

Then select g_{m_*} which has lowest s_e.

Problems: (1) s_e has high variance. So pick K reasonable.

(2) We no longer have honest validation. s_e is not honest because it could be overfit since $\mathbb{D}_{test}$ was used many times.



Procedure

(1) Fit g_1 on $\mathbb{D}_{train}$, compute s_{e_1} on $\mathbb{D}_{select}$

(2) Fit g_2 on // // // s_{e_2} // // //

⋮

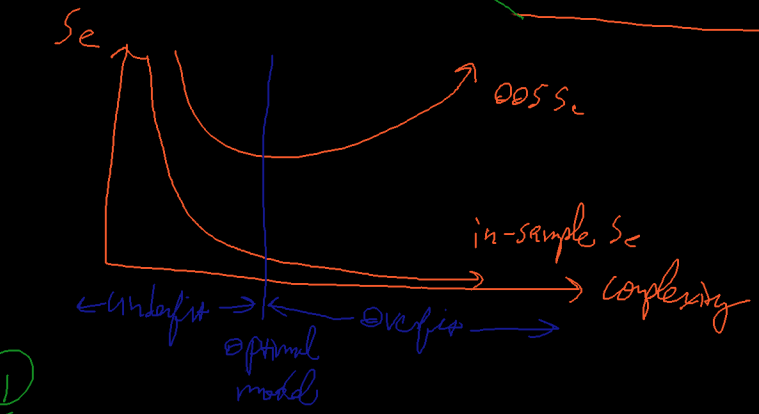
(M) Fit g_M ..... s_{e_M} // // //

(M+1) Final $m_* = \text{argmin}\{s_{e_j}\}$

(M+2) Fit g_{m_*} on $\mathbb{D}_{train} \cup \mathbb{D}_{select}$ and compute $s_{e_{m_*}}$ on $\mathbb{D}_{test}$.

(M+3) Fit g_{final} on $\mathbb{D}$.

Uses of this procedure



(1) Consider models $g_1, g_2, \ldots g_M$ with increasing complexity.

This is sometimes called stepwise modeling.