Week 4: Model selection & validation

Hyperparameters

Ridge Regression with poly. expansion

$$\left(\min_w \frac{1}{2N}\sum_{n=1}^{N}(y_n - x_n^T w) + \lambda\|w\|_2^2\right) \rightarrow \lambda, d$$

- Generally, the data $\sim D$ with D unknown

True error:

$$L_D(f) = \mathbb{E}_D(\ell(y, f(x)))$$

Empirical error:

$$L_S(f) = \frac{1}{|S|}\sum_{(x_n,y_n)\in S} \ell(y_n, f(x_n))$$

- $f_S = A(S)$ (learning alg.)

Train error:

$$L_S(f_S) = \frac{1}{|S|}\sum_{(x_n,y_n)\in S} \ell(y_n, f_S(x_n))$$

Train-Test split:

$S = S_{Train} \cup S_{Test}$, $f_S = A(S_{Train})$

$$L_{S_{Test}}(f_S) = \frac{1}{|S_{Test}|}\sum_{(x_n,y_n)\in S_{Test}} \ell(y_n, f_S(x_n))$$

the true error is:

$$L_D(f) = \mathbb{E}_{(y,x)\sim D}[\ell(y, f(x))]$$

generalization error:

$$|L_D(f) - L_{S_{Test}}(f)|$$

empirical error is unbiased:

$$L_D(f) = \mathbb{E}_{Test\sim D}[L_{S_{Test}}(f)]$$

Variation:

Then loss decreases in $O(1/\sqrt{|S_{Test}|})$

→ More data → More confident bound

$$\mathbb{P}\left[|L_D(f) - L_{S_{Test}}(f)| \geq \sqrt{\frac{(b-a)^2 \ln(2/\delta)}{2|S_{Test}|}}\right] \leq \delta$$

given $\ell(\cdot,\cdot)$ is bounded in $[a,b]$

Chernoff Bound:

$\theta_1 \dots \theta_n$, iid with mean $\mathbb{E}[\theta]$ and range $[a,b]$

Then for any $\varepsilon > 0$,

$$\mathbb{P}\left[\left|\frac{1}{N}\sum_{n=1}^{N}\theta_n - \mathbb{E}[\theta]\right| \geq \varepsilon\right] \leq 2e^{-2N\varepsilon^2/(b-a)^2}$$

Equating $2e^{-2|S_{Test}|\varepsilon^2/(b-a)^2} = \delta$

$$\varepsilon = \sqrt{\frac{(b-a)^2 \ln(2/\delta)}{2|S_{Test}|}}$$

$$\mathbb{P}\left[L_D(f) > L_S(f) + \sqrt{\frac{(b-a)^2 \ln(2/\delta)}{2|S|}}\right] \leq \delta$$

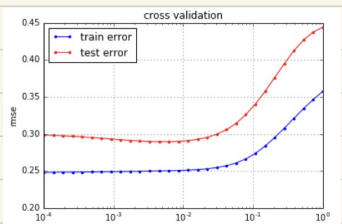
- Probabilistic upper bound on the true risk since both $L_S(f)$ as well as the sum term can be computed with the data.

Model Selection:

- Select hyperparameters of our model.
  (i.e λ in ridge regression)

$S = S_{Train} \cup S_{Test}$ and we suppose $S_{Train} \perp\!\!\!\perp S_{Test} \sim D$

λ_k for $k = 1 \dots K$



- run the learning algorithm K times on S_{Train}.
  → f_{S_{Train}, λ_k} and compute Training error and Test error.
  ($L_{S_{Test}}(f_{S_{Train},\lambda_k})$)

→ We select the best model based on the Test error

- Similarly as before we have:
  The maximum deviation for all K candidates is bounded:

$$\mathbb{P}\left[\max_k |L_D(f_k) - L_{S_{Test}}(f_k)| \geq \sqrt{\frac{(b-a)^2 \ln(2K/\delta)}{2|S_{Test}|}}\right] \leq \delta$$

- error decreases as $O\left(\frac{1}{\sqrt{|S_{Test}|}}\right)$

and as we have K hyperparameters, error only goes up by a small factor $\sim \sqrt{\ln(k)}$

Let $K^* = \arg\min_k L_D(f_k)$
$\hat{K} = \arg\min_k L_{S_{Test}}(f_k)$

$$\mathbb{P}\left[L_D(f_{\hat{K}}) > L_D(f_{K^*}) + 2\sqrt{\frac{(b-a)^2 \ln(2K/\delta)}{2|S_{Test}|}}\right] \leq \delta$$

In words, if we choose the "best" function according to the empirial risk then its true risk is not too far away from the true risk of the optimal choice.

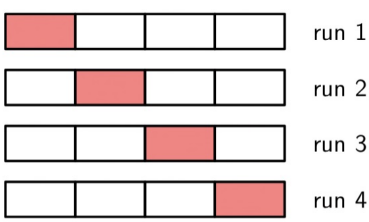
Cross validation :

Splitting the data into 2 subsets: Train and test is not the most efficient way to use the data.

K-fold cross-validation is a popular variant.

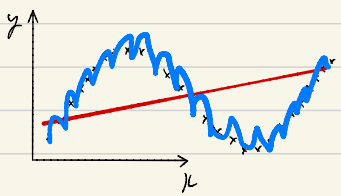
- Randomly partitions into K groups.
- Train K times, each time leaving exactly one of the K groups for testing.
- Average the K results.

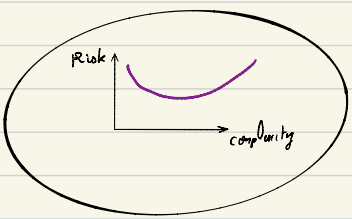
Note: Have used all data for training, and all data for testing, and used each data point the same number of times.



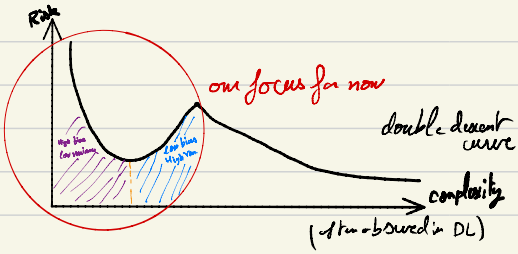
Cross-validation returns an unbiased estimate of the *generalization error* and its variance.

Bias - Variance decomposition





Important note



Data model:

$y = f(x) + \varepsilon$ (true model + noise)

noise: $\varepsilon \sim D_\varepsilon$ iid, $\perp\!\!\!\perp D_x$

$\mathbb{E}[\varepsilon] = 0$

- In general, the model will not be realizable. i.e.: f is not in our model class (cannot recover f exactly)

- $\mathbb{E}_D[(y - f_{S_{Train}}(x))^2]$ $\quad (f_{S_{Train}} = A(S_{Train}))$

- look at one single point x_0: $(y - f_{S_{Train}}(x_0))^2 = (f(x_0) + \varepsilon - f_{S_{Train}}(x_0))^2$

$$\mathbb{E}_{\substack{\varepsilon \sim D_\varepsilon \\ S_{Train} \sim D}}\left[(y - f_{S_{Train}}(x_0))^2\right] = \mathbb{E}\left[(f(x_0) + \varepsilon - f_{S_{Train}}(x_0))^2\right]$$

$$= \mathbb{E}_S\left[(f(x_0) - f_{S_{Train}}(x_0))^2\right] + \mathbb{E}[\varepsilon^2] + \text{Cross term}$$

$\mathbb{E}[\varepsilon^2] = \text{Var}(\varepsilon) + (\mathbb{E}[\varepsilon])^2$, $\mathbb{E}[\varepsilon] = 0$

Cross term: $\mathbb{E}[2(f(x_0) - f_S(x_0))\varepsilon]$, $D \perp\!\!\!\perp D_\varepsilon$

$= 2\,\mathbb{E}[\varepsilon]\,\mathbb{E}[f(x_0) - f_S(x_0)] = 0$

$$\Rightarrow \mathbb{E}\left[(y - f_{S_{Train}}(x_0))^2\right] = \mathbb{E}\left[(f(x_0) - f_S(x_0))^2\right] + \text{Var}(\varepsilon)$$

- $\mathbb{E}\left[(f(x_0) - f_S(x_0))^2\right] = \mathbb{E}_S\left[\left(f(x_0) - \mathbb{E}_{S'}[f_{S'}(x_0)] + \mathbb{E}_{S'}[f_{S'}(x_0)] - f_S(x_0)\right)^2\right]$

$$= \left(f(x_0) - \mathbb{E}_{S'}[f_{S'}(x_0)]\right)^2 + \mathbb{E}_S\left[\left(f_S(x_0) - \mathbb{E}_{S'}[f_{S'}(x_0)]\right)^2\right] + C.T$$

$C.T = \mathbb{E}_S\left[(f(x_0) - \mathbb{E}_{S'}[f_{S'}(x_0)])(\mathbb{E}_{S'}[f_{S'}(x_0)] - f_S(x_0))\right]$

$= (f(x_0) - \mathbb{E}_{S'}[f_{S'}(x_0)])\,\mathbb{E}_S[\mathbb{E}_{S'}[f_{S'}(x_0)] - f_S(x_0)] = 0$

- $\mathbb{E}_{\substack{\varepsilon \sim D_\varepsilon \\ S \sim D}}\left[(y - f_{S_{Train}}(x_0))^2\right] = \text{Var}(\varepsilon) + \underbrace{(f(x_0) - \mathbb{E}_{S_{Tr}}[f_{S_{Tr}}(x_0)])^2}_{\text{Bias}} + \underbrace{\mathbb{E}_{S_{Tr}}\left[(f_{S_{Tr}}(x_0) - \mathbb{E}_{S_{Tr}}[f_{S_{Tr}}(x_0)])^2\right]}_{\text{Variance}}$

$\Rightarrow$ Decomposition in 3 positives: Noise term (noise data), Bias term, Variance Term

Bias term: $f(x_0)$: actual fct. $\mathbb{E}_S[f_{S(x_0)}]$: expectation of the pred. (How precise)

→ If the complexity is high → low bias | if complexity is low → High bias

Variance term: $f_S(x_0)$: pred. $\mathbb{E}_S[f_{S(x_0)}]$: average pred (Expected) (How consistent)

→ If the complexity ↗ Then Variance ↗ | if complexity ↘, Var. ↘