# Machine Learning and Statistical Learning Theory II - Lecture 2 Model Complexity

Daniel Yu

September 11, 2024

# Contents

# 1 Motivation

How to select "good" models? We want models that generalize well to future data (which by definition is unknowable). With sufficient data, we can divide data into:

1. training set
2. validation set – pretesting model before formally on test set
3. test set – quality of final model

What if the data is not enough?
We can use predicted error estimates from **error prediciton methods** based on training data

1. AIC (Akaike information criterion)
2. BIC (Bayesian information criterion)
3. MDL (Minimal Description length)
4. VC (Vapnik Chernovenkis dimension)

How can we approximate/estimate the number of effective parameters from different types of models? We can use **degrees of freedom** and **VC-dimension**

**Definition 1.** Degrees of freedom of a model is number of effective free paramters $\theta_i$ that are being determined by training set $D$. It can be thought of as:

$$df = \text{ num indep vars} - \text{ num of statistics.}$$

Recall SVD: $X = UDV^T$ where $U_{N\times N}$ orthogonal, $V_{d\times d}$ orthogonal, $D_{N\times d}$ diagonal matrix with $\sigma_i$ singular values.

## 1.1 Degrees of Freedom for OLS

$$\begin{aligned}
\vec{y}_{\text{ols}} &= H\vec{y} = X(X^TX)^{-1}X^T\vec{y} \\
&= UDV^T\left(VD^TU^TUDV^T\right)^{-1}UDV^T\vec{y} \\
&= UDV^T\left(VD^2V^T\right)^{-1}UDV^T\vec{y} \\
&= UDV^TV\left(D^2\right)^{-1}V^TUDV^T\vec{y} \\
&= UDD^{-2}DU^T\vec{y} \\
&= UU^T\vec{y} \\
&= \sum_{j=1}^{d}\vec{u}_j\vec{u}_j^T\vec{y}
\end{aligned}$$

Thus, the degrees of freedom: $df(y_{\text{ols}}) = rank(X) = d$

### 1.2 Degrees of Freedom for Ridge Recall

$$\begin{aligned}\vec{y}_{\text{Ridge}} &= H\vec{y} = X(X^TX + \lambda I)^{-1}X^T\vec{y} \\ &= UDV^T\left(VD^TU^TUDV^T + \lambda I\right)^{-1}UDV^T\vec{y} \\ &= UD\left(D^2 + \lambda I\right)^{-1}DU^T\vec{y}.\end{aligned}$$

since $U^T \cdot U = I, V^TV = I$ as $U, T$ are orthogonal matrices. And thus,

$$\vec{y}_{\text{Ridge}} = \sum_{j=1}^{d} \vec{u}_j \left[\frac{\sigma_j^2}{\sigma_j^2 + \lambda}\vec{u}_j^T\vec{y}\right] = H_\lambda\vec{y}.$$

And the degrees of freedom are:

$$\text{df} = \text{Tr}(H_\lambda) = \sum_{j=1}^{d} \frac{\sigma_j^2}{\sigma_j^2 + \lambda}.$$

When $\lambda > 0$, $\frac{\sigma_j^2}{\sigma_j^2+\lambda} < 1$, so $df < d$. But when $\lambda = 0$, $\frac{\sigma_j^2}{\sigma_j^2+\lambda} = 1$ and thus reduces to $OLS$ and $df = d$. What is happening is that the **regularization term** $\lambda$ introduces penalties on small singular values, **reducing the model's complexity** resulting in few degrees of freedom!

## 2 Degrees of Freedom

Start

**Definition 2.** The degrees of freedom of $\hat{h}$ mathematically is defined as:

$$df(\hat{h}) = \frac{1}{\sigma^2}\sum_{i=1}^{N} Cov(\hat{y}^i, y^i) = \frac{1}{\sigma^2}Tr(Cov(\hat{\vec{y}}, \vec{y})).$$

Generalization of the number of parameters.

### 2.1 Mean Predictor

**Note.** Example
Let's say we have one degree of freedom, a predictor with a single parameter. For example, the mean predictor where 1 we assume $y^i = h(\vec{x}^i) + \varepsilon^i$:

$$\hat{y} = \hat{h}(x) = \frac{1}{N}(y^1 + \ldots + y^N).$$

has degrees of freedom:

$$\begin{aligned} df(\hat{h}) &= \frac{1}{\sigma^2}\sum_{i=1}^{N} Cov(\hat{y}^i, y^i) \\ &= \frac{1}{N\sigma^2}\sum_{i=1}^{N} Cov(y^1 + \ldots + y^N, y^i) \\ &= \frac{1}{N\sigma^2}\sum_{i=1}^{N} Cov(y^i, y^i) \\ &= \frac{1}{N\sigma^2}\sum_{i=1}^{N} \sigma^2 = 1. \end{aligned}$$

like we expected

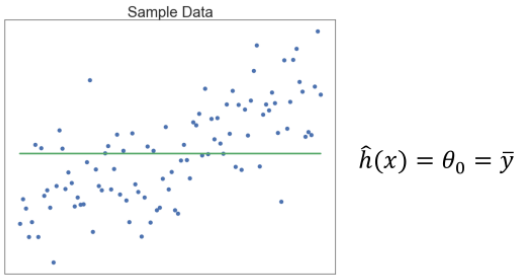


Figure 1: Mean Predictor

TODO OLS and later

## 2.2 Identity Predictor

**Note.** Another example is an **identity** estimator like $\hat{y}^i = f(x^i) = y^i$ has $N$ degrees of freedom:

$$df(\hat{h}) = \frac{1}{\sigma^2}\sum_{i=1}^{N} Cov(\hat{y^i}, y^i) = \frac{1}{\sigma^2}\sum_{i=1}^{N} Cov(y^i, y^i) = \frac{1}{\sigma^2}\sum_{i=1}^{N} \sigma^2 = N.$$

## 2.3 OLS Degrees of Freedom

**Note.** OLS Example
Let $\hat{y} = X\hat{\vec{\theta}} = X(X^TX)^{-1}X^T\vec{y}$ :

$$\begin{aligned} df(\hat{h}) &= \frac{1}{\sigma^2} Tr(Cov(\hat{\vec{y}}, \vec{y})) \\ &= . \end{aligned}$$

TODO finish notes here

## 2.4 Linear Smoother DoF

Note that () is a special function, essentialy acting as a weighted average!

**Definition 3.** A model call linear smoother has forms:

$$\hat{f}(\vec{x}) = \sum_{j=1}^{N} w(\vec{x}, \vec{x}^j) \cdot y^j \tag{1}$$

$$\hat{f}(\vec{x}^i) = \sum_{j=1}^{N} w(\vec{x}^i, \vec{x}^j) \cdot y^j \tag{2}$$

$$\hat{f}(\vec{x}) = \hat{\vec{y}} = W\vec{y} \tag{3}$$

where (1)-(3) are equivalent forms!

TODO Derivation of the degrees of freedom

*Proof.* TODO □

## 2.5 K-Nearest Neighbors DoF

**Remark.** Actually a K-Nearest Neighbors is a **special case** of the linear smoothers.

Recall,

$$\hat{f}(\vec{x}) = \frac{1}{k} \sum_{\vec{x}^j \in N_0} y^j.$$

In fact, this is just equal to $\sum_{j=1}^{N} w\left(\vec{x}, \vec{x}^j\right) \cdot y^j$ where

$$w\left(\vec{x}, \vec{x}^j\right) = \begin{cases} \frac{1}{k} \text{ if } \vec{x}^j \in N_0 \\ 0 \text{ otherwise} \end{cases}.$$

**Remark.** If $k = 1$, then it is the **identity predictor**. If $k = N$ then it is the **mean** predictor!

*Proof.* TODO Expand and rigorize proof

$$df(\hat{\vec{y}} = Tr(W) = \frac{N}{K}.$$

□

# 3 Estimating Degrees of Freedom

While we do have a theoretical formula (in the definition) for the degrees of freedom, in many models, there is no analytic calculation.

We can estimate the degrees of freedom through simulation! I.e. using random resampling (taking one datapoint in $D_1, D_2, \ldots$) in something called the **bootstrap** method.

$$Cov(x,z)_{\text{sample}} = \frac{\sum_{i=1}^{B}(x_i - \overline{x})\,(z_i - \overline{z})}{B-1}.$$

**Note.** Procedure
For $B = 1$ to $B = N$:

1. Draw samples $\left(\vec{x}_b^i, y_b^i\right)$ for $i = 1, \ldots, N$
2. Compute $\hat{\vec{y}}_b = \begin{pmatrix} h(\vec{x}_b^1) \\ \vdots \\ h(\vec{x}_b^N) \end{pmatrix}$ and $\vec{y}_b \begin{pmatrix} y_b^1 \\ \vdots \\ y_b^N \end{pmatrix}$
3. Calculate using the following:

$$\begin{aligned} df(\hat{h}) &= \frac{1}{\sigma^2}\sum_{i=1}^{N} Cov(\hat{y}^i, y^i) \\ &= \frac{1}{\sigma^2}\sum_{i=1}^{N} Cov_{\text{sample}}\left(\hat{y}^i, y^i\right) \\ &= \frac{1}{\sigma^2}\sum_{i=1}^{N}\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{y}_b^i - \overline{y}_b^i\right)\left(y_b^i - \overline{y}_b^i\right). \end{aligned}$$

Note: May also have to estimate variance using the sample in practice!

# 4 Training and Test Error

Let $D = \{\vec{x}^i, y^i\}, i = 1, \ldots, N$ the training data.

**Definition 4.** Sample training Error
Defined as the average error of our $\hat{f}(x)$ on training points $D$:

$$err_D = \frac{1}{N}\sum_{i=1}^{N} L(y^i, \hat{f}(\vec{x}^i)).$$

**Definition 5.** Expected Test Error
Final Goal in theory! Defined as the expected error on future data (impossible) points $(\vec{X}, Y)$ based on current fixed training set $D$:

$$Err_D = E(L(Y, \hat{f}(\vec{X}))).$$

Averaging over all possible $D$ is just called **Err** $E_D(Err_D)$

**Definition 6.** In Sample Prediction Error
A mix of Test Error and the theoretical Expected Test Error. Still not computable, but closer

$$Err_in = \frac{1}{N}\sum_{i=1}^{N} E_{Y_{new}}(L(Y^i_{new}, \hat{f}(\vec{x}^i))).$$

Expected error based on same function $\hat{f}\left(\vec{x}^i\right)$ from $D$ with same $\vec{x}^i$ but newly generated $y^i_{new}$ from the underlying distribution (this is the unrealistic assunption part).

**Definition 7.** Optimism
The difference between test and training errors is known as **optimism** of estimator $\hat{h}$.

$$optimism = Err_{in} - err_D = \frac{1}{N}\sum_{i=1}^{N} E_{Y_{new}}(L(Y^i_{new}, \hat{f}(\vec{x}^i))) - \frac{1}{N}\sum_{i=1}^{N} L(y^i, \hat{f}(\vec{x}^i)).$$

**Theorem 1.** Optimism and Degrees of Freedom
We are averaging over all $y_i$ in $D$:

$$E_y(Optimism) = \frac{2}{N}\sum_{i=1}^{N} Cov(\hat{y}^i, y^i) = \frac{2\sigma^2}{N} df(\hat{f}).$$

Derivation:

$$\begin{aligned} E_y(Err_{in} - err_D) &= E_y(Err_{in} - err_D) \\ &= E_y[\frac{1}{N}\sum_{i=1}^{N} E_{Y_{new}}(L(Y^i_{new}, \hat{f}(\vec{x}^i)))] - E_y[\frac{1}{N}\sum_{i=1}^{N} L(y^i, \hat{f}(\vec{x}^i))] \\ &= \frac{2\sigma^2}{N} df(\hat{f}). \end{aligned}$$

## 4.1 Mean Square Error Example

We can find $Err_{in} = err_D + \frac{2\sigma^2}{N} df(\hat{f}) \approx T$:

$$T = \frac{1}{N}\sum_{i=1}^{N} L(y^i, \hat{f}(\vec{x}^i)) + \frac{2\sigma^2}{N} df(\hat{f}).$$

By optimism, where $L\left(y^i, \hat{y}^i\right) = (y^i - \hat{y}^i)^2$

$$\begin{aligned} E[Err_{in}] &\approx E[T] \\ &= E[\frac{1}{N}\sum_{i=1}^{N} L(y^i, \hat{f}(\vec{x}^i)) + \frac{2\sigma^2}{N} df(\hat{f})] \\ &= E[\frac{1}{N}\sum_{i=1}^{N} L(y^i, \hat{f}(\vec{x}^i))] \\ &= \text{ TBH don't really understand the steps.} \end{aligned}$$

And it is an unbiased estimator!

Thus, if we knew the estimators DoF, then we can use T to approximate test error!

**Definition 8.** The **Cp information criterion** for model selection in linear regression is then just defined as ($df(f_k) = k$ since in lin reg, the dof = num params):

$$\hat{k} = \arg\min_k \frac{1}{N}\sum_{i=1}^{N}(y^i - \hat{y}^i)^2 + \frac{2\sigma^2}{N}k.$$

# 5 AIC Criterion for Log

A test error estimate. Generally, for log-likelihood models.

# 6 VC Dimension

Before: We want estimate for $Err_{in}$ in terms of training set and correction term.

**Remark.** $VC$ gives an upper bound of **full Ex[ected Test Error** i.e.

$$Err = E_{Y,\vec{X}}(L(Y, \hat{f}(\vec{X}))).$$

**Definition 9. indicator** functions are defined as:

$$f(\vec{x}) = \begin{cases} 1 \text{ if } \vec{x} \in A \\ 0 \text{ otherwise} \end{cases}.$$

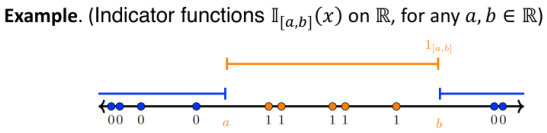


Figure 2: Ex: Indicator Function

**Definition 10.** A family $F$ of indicator functions $f(\vec{x}) = I_A(\vec{x})$ on $\mathbb{R}^d$ shatters a fixed finite collection of points $C$ if the following holds:

For every subset $C_1 \subseteq C$, $\exists$ a function $I_A(\vec{x})$ in family with 1 on $C_1$ and 0 on $C \setminus C_1$.

In terms of estimating model complexity, what this equates to is the indicator function being our model (ex: $y = \theta_1 x_1 + \theta_2 x_2 + \theta_0$) and the family of indicator functions being all possible values for parameters $\theta$. Then the collection of points we attempt to shatter are analogous training set datapoints where 1 represents the points we want classify correctly using model and 0 representing all the points outside. So can the model correctly classify all possible labellings of arbitray number of points and the maximum number possible is $VC$ dimension

TODO can't screenshot tool working


Figure 3: VC Dimension of 2 (can't shatter the right example! Since interval must be continuous)

**Note.**

- What about when indicator functions of circle with radius $r \in \mathbb{R}^+$ centered at $(0, 0)$? VC Dim = 1, can't classify the point further away as being in the circle with the point closer as not
- What about if the circles could have the center vary $(a, b) \forall a, b \in \mathbb{R}$? VC Dim = 3, can't classify 4.

- How about 3-dimensional balls with radius $r \in \mathbb{R}^+$ centered at $\vec{0}$?

**Definition 11.** The family $F$ of indicator functinos has **VC Dimensions** $n$ if $n$ is largest number of points $F$ can always shatter. It can be thought of a way of measuring complexity of a class of functions by measuring how wiggly its members can be.

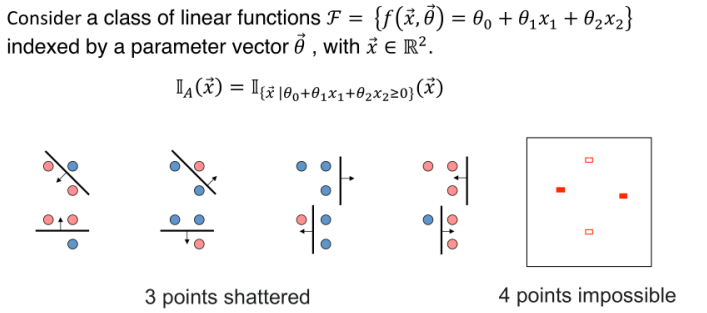


Figure 4: VC dimension Example for Linear functions

**Note.** Thus, the VC dimension of a linear model is as expected: $VC = d + 1$ where $\theta_1, \ldots, \theta_d$

However, in general, the number of paramters is not always approximate of the VC dimensions! For example, consider the classifer:

$$f(x, a) = \lceil \sin(ax) \rceil \text{ i.e. the ceiling.}$$

which has only 1 parameter, but infinite VC dimesion

## 6.1 VC Dimesion Bound

**Definition 12.** Error Bound

$$err_{\text{test}} \leq err + \frac{\varepsilon}{2}\left(1 + \sqrt{1 + 4 \cdot \frac{err}{\varepsilon}}\right).$$

Thus, VC Dimensions give an upper bound for the test error based on training error

**Note.** This is only for binary classification