03_Losses_Regularization_Evaluation

1. Regularization
 1.1. Stabilization
  1.1.1. Background
  1.1.2. Solution
  1.1.3. Geometry
 1.2. General Approach to regularization
 1.3. Sparsity Regularization
  1.3.1. Model
  1.3.2. Geometry
2. Bias-Variance
 2.1. Bias-Variance Decomposition
 2.2. Bias-Variance Tradeoff and Regularization
3. Evaluating Learners
 3.1. Errors
  3.1.1. Estimate True Error
  3.1.2. Cross Validation
 3.2. Learning Curves
 3.3. Feature Curves
 3.4. Curse of Dimensionality
 3.5. Confusion Matrices

1. Regularization

1.1. Stabilization

1.1.1. Background

Take Linear Regression as Example: $\widehat{w} = \left(X^T X\right)^{-1} X^T Y$, if:

- we try to use many dimensions features but only have few observations
- the train data is not "good" (does not present true distribution)

Then some eigenvalues of X may be very small, which means in the inverse, some features will have very large eigenvalues, which means

- the given feature dominates the regression
- new observation of different sample will lead to very large fluctuation on the w, which means, **unstable**

1.1.2. Solution

An idea is: keep the eigenvalues away from 0

$$\widehat{w} = \left(X^T X + \lambda I\right)^{-1} X^T Y$$

- The new w will perform poorer on train data

- But it has high possibility perform better on true data

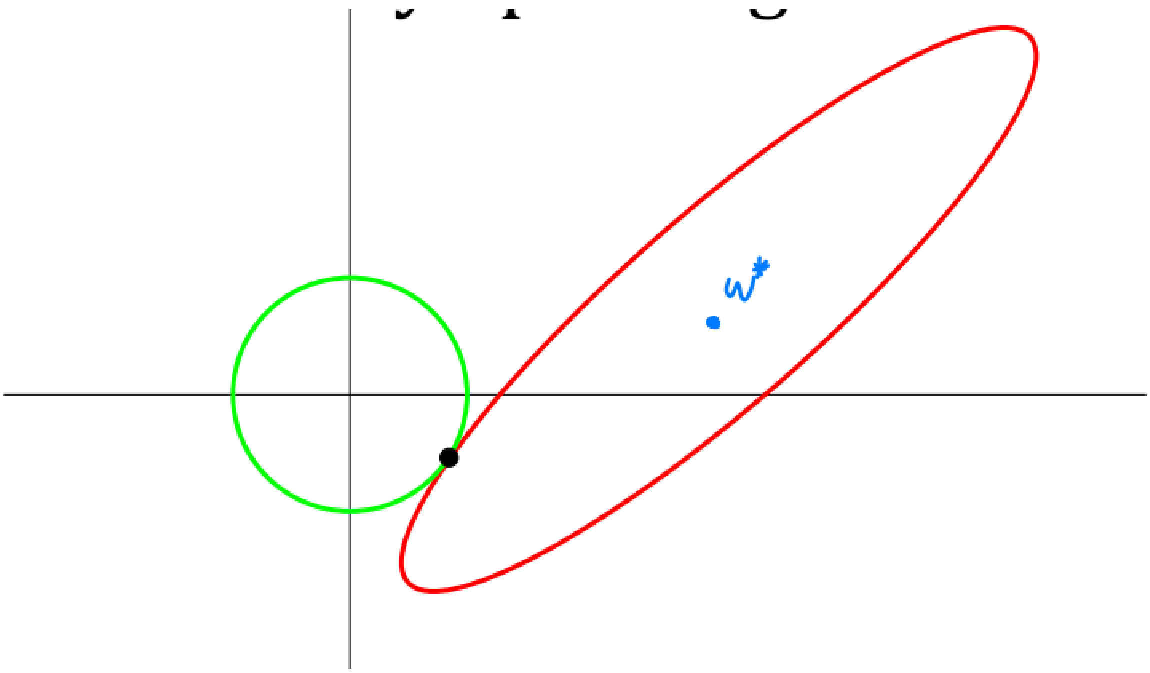
The above equation is equal to:

$$\min_w \sum_{i=1}^{N} \left(x_i^T w - y_i\right)^2 + \lambda \|w\|^2$$

Or the second format:

$$\begin{aligned} &\min_w \sum_{i=1}^{N} \left(f\left(x_i, w\right) - y_i\right)^2 \\ &\text{s.t. } \|w\|^2 \leq \tau \end{aligned}$$

1.1.3. Geometry



1.2. General Approach to regularization

$$\min_w \sum_{i=1}^{N} \ell\left(f\left(x_i, w\right), y_i\right) + R(f) \tag{1}$$

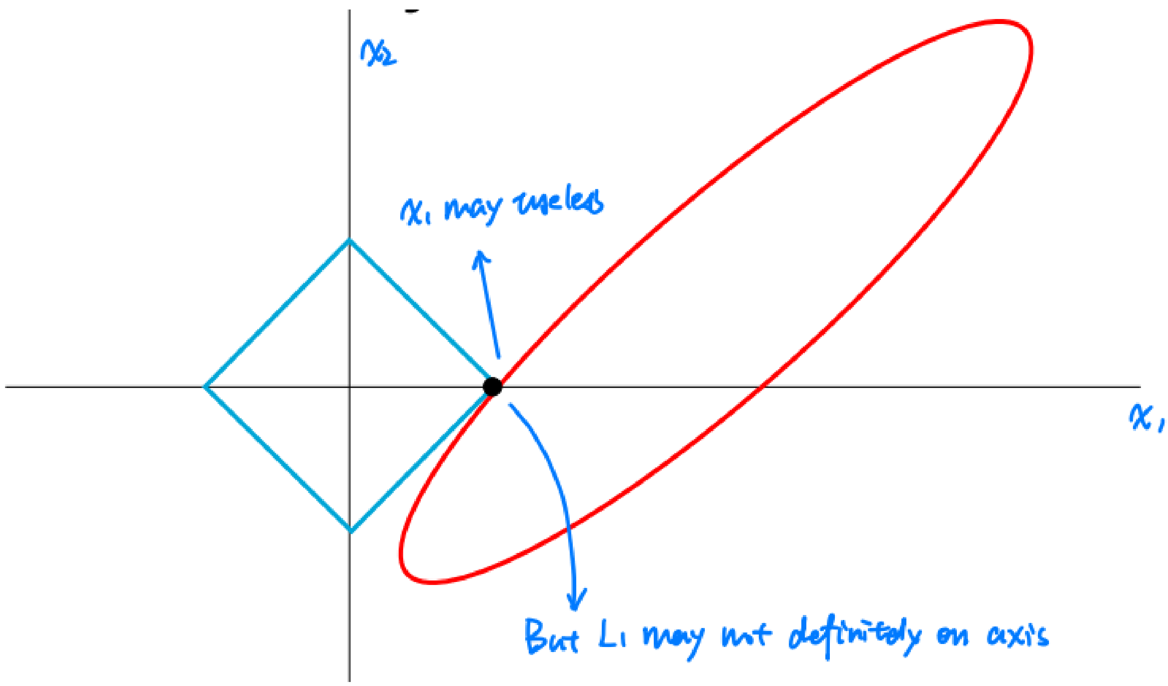
1.3. Sparsity Regularization

- <u>*Sparsity*</u>: make some w_i to zero
  - To some extend means **simplify the model** (lower down the overfitting)

1.3.1. Model

$$\begin{aligned} & \min_w \sum_{i=1}^N \left(f\left(x_i, w\right)-y_i\right)^2 \\ & \text{s.t. } \|w\|_1 \leq \tau \end{aligned} \tag{2}$$

1.3.2. Geometry



2. Bias-Variance

2.1. Bias-Variance Decomposition

Assume we have a given format of classification/regression function: f

- optimal prediction $f^*(x)$ (may not belong to $\mathbb{F}$)
- estimate based on some given data $\hat{f}(x) \in \mathbb{F}$

$$\begin{aligned} & \mathbb{E}\left[\left(f^*-\hat{f}\right)^2\right] \\ = & \mathbb{E}\left[\left(f^*-\mathbb{E} \hat{f}\right)^2\right]+\mathbb{E}\left[(\mathbb{E} \hat{f}-\hat{f})^2\right] \\ = & \text{ bias }^2+\text{ variance } \end{aligned} \tag{3}$$

Understanding

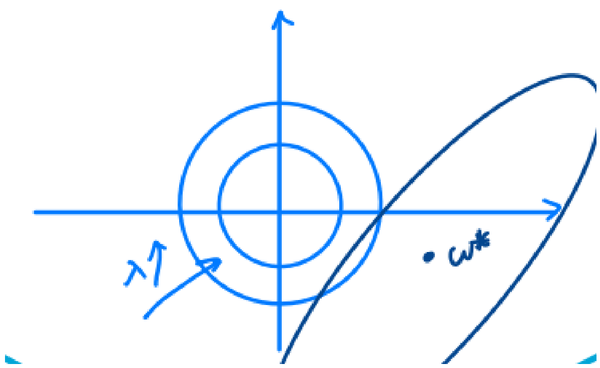
- bias is something like "inherent error" between model assumption and real optimal function

- regression is the sensitivity of the given assumption regarding to dataset

2.2. Bias-Variance Tradeoff and Regularization

larger λ

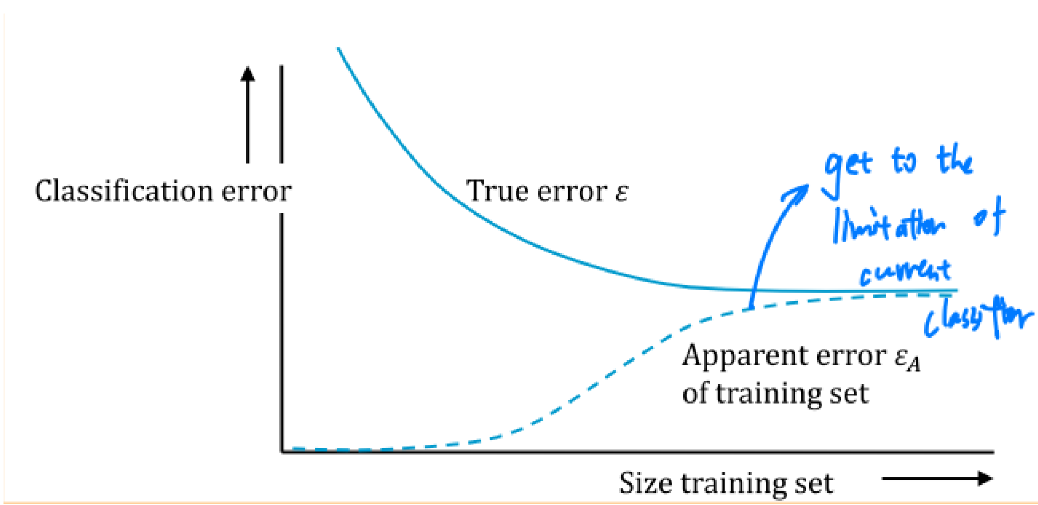
- larger bias: go away from original information from the dataset
- smaller variance: because above, dataset fluctuation is not so important



3. Evaluating Learners

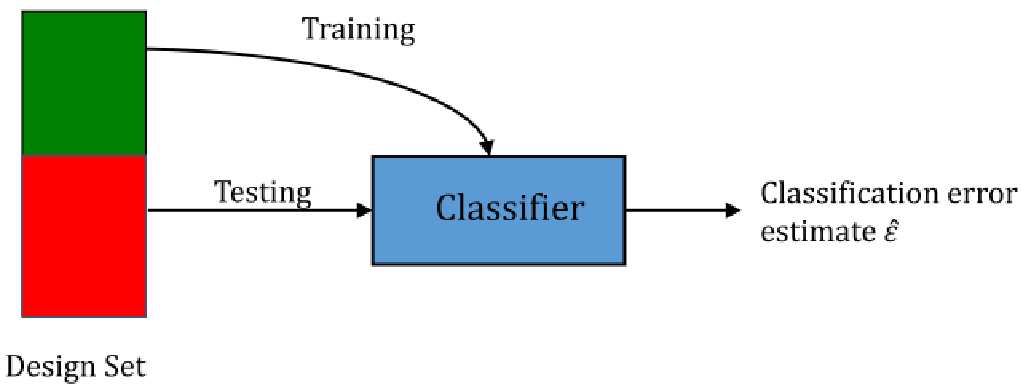
We will use **classifiers** as example

3.1. Errors

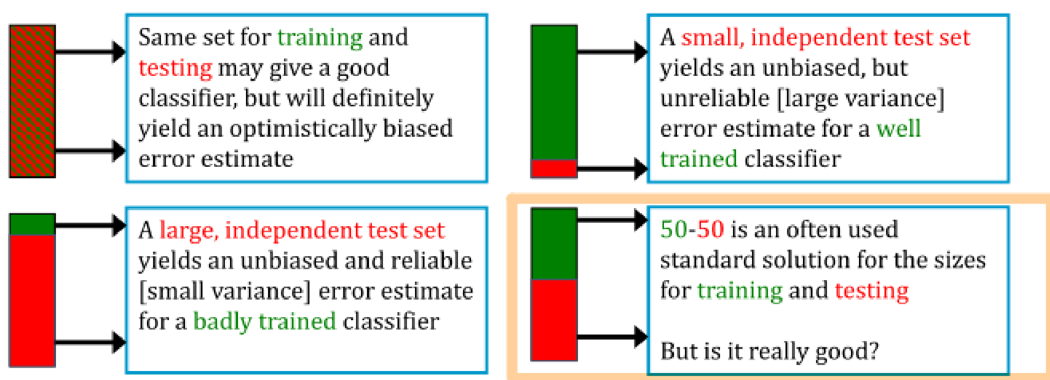


3.1.1. Estimate True Error

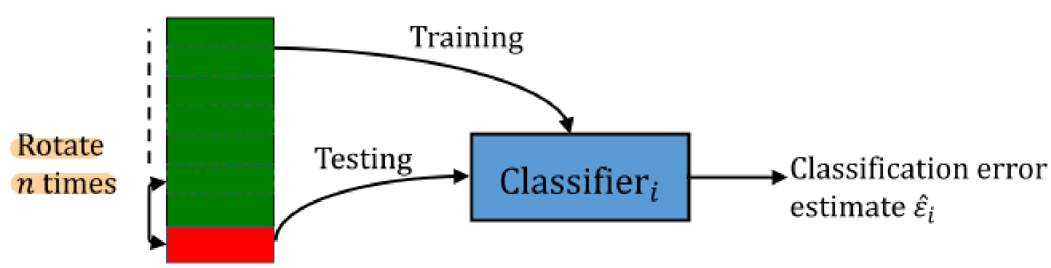
We can estimate *true error* based on test set

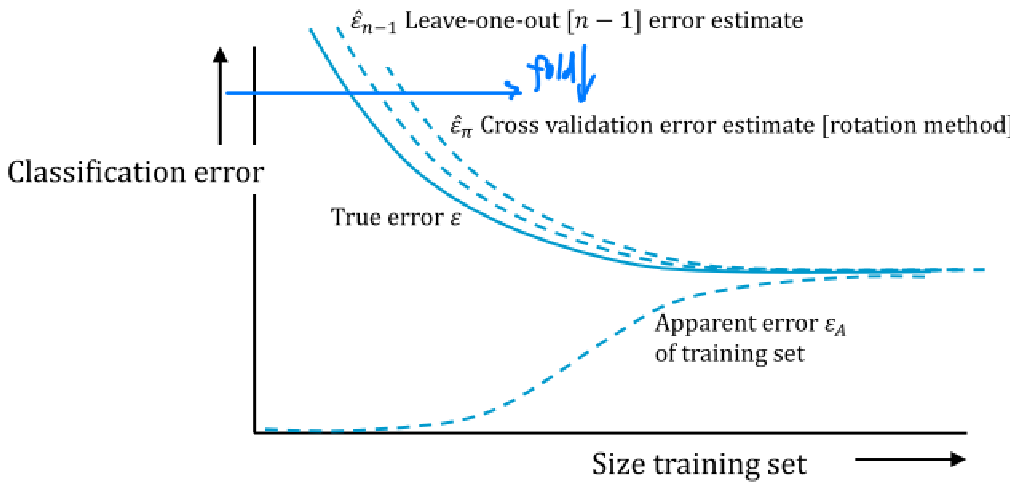


Other training set → other classifier
Other test set → other error estimate

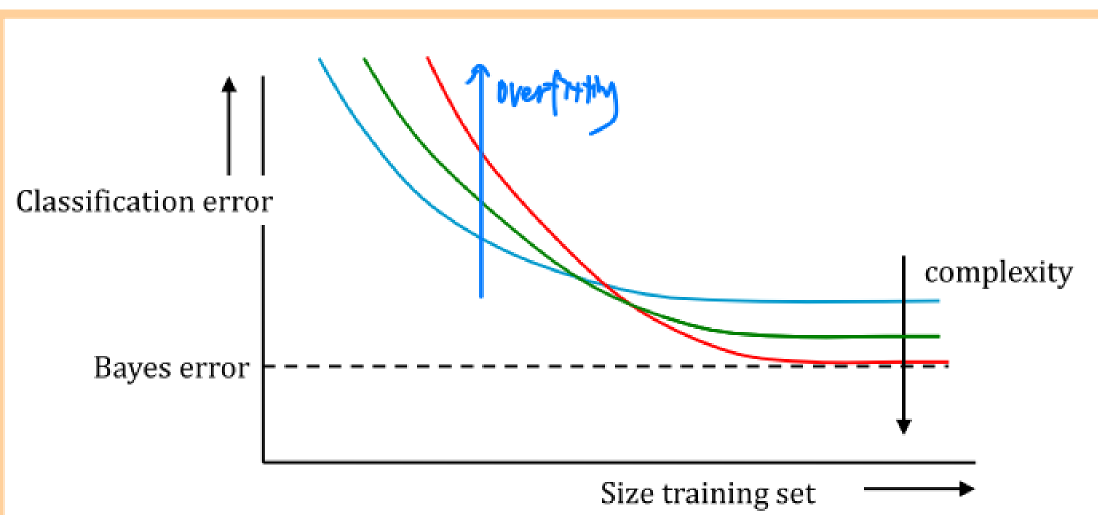


3.1.2. Cross Validation

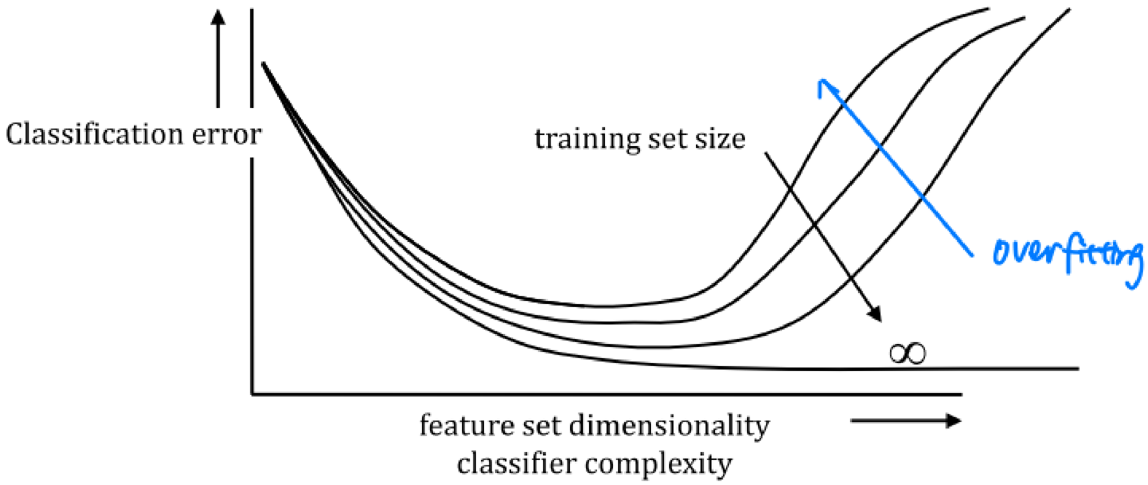




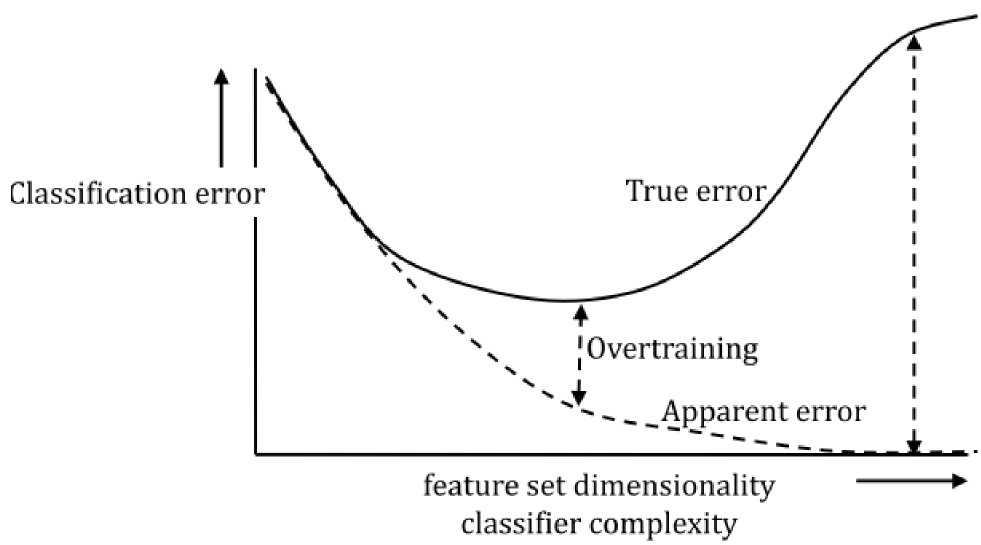
3.2. Learning Curves



3.3. Feature Curves



3.4. Curse of Dimensionality



For a given set, when the complexity is higher, whether the data set is large enough to learn all parameters of the classifier is unknown.

3.5. Confusion Matrices

Provides counts of class-dependent errors : How many object have been classified as A that should have been classified as B?