CS 307 Lecture 18

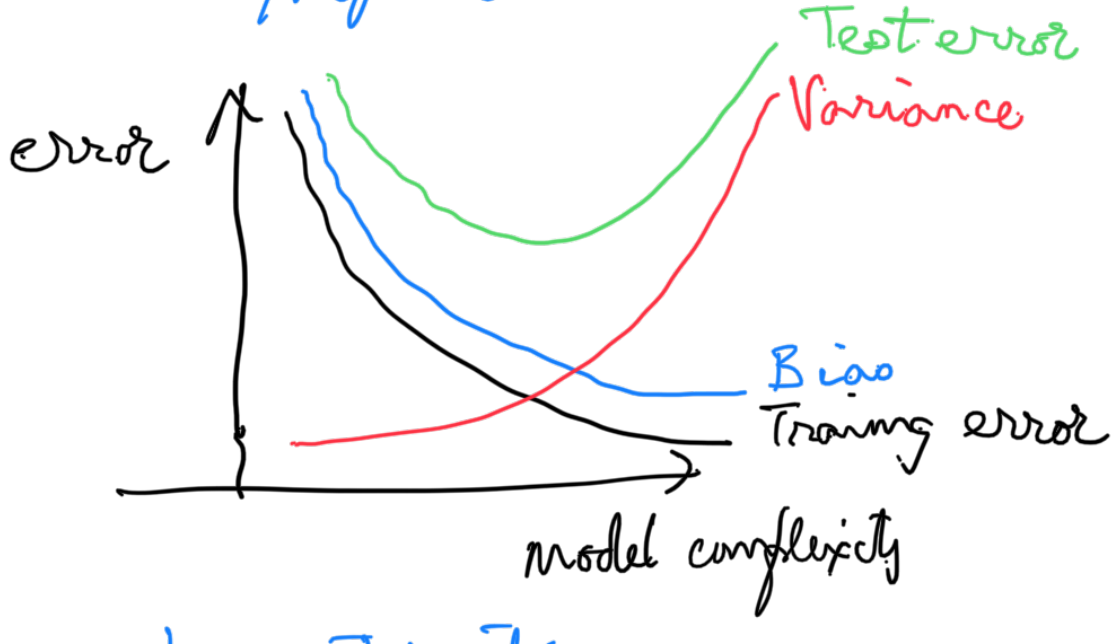
We have $(x, y) \sim D$

- Training set drawn from D
- Goal is to discover a hypothesis h

$$\min \; E_{(x,y)\sim D}\left[\text{error}\left(h(x), y\right)\right]$$

Two reasons why h may do poorly on new examples

- Bias: Hypothesis class is not rich enough to explain the data.
  - High training error } Underfitting
  - High test error
- Variance: The hypothesis is too tightly aligned to the training set.
  - Low training error } Overfitting.
  - High test error



Over parametrization

Regularization: Minimize

$$J_\lambda(\theta) = J(\theta) + \lambda R(\theta)$$

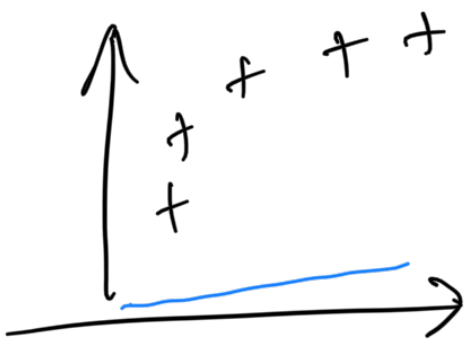
- $J(\theta)$: Old cost fn
- λ: hyper parameter
- $R(\theta)$: Regularization
  - $\frac{1}{2}\|\theta\|_2^2$
  - $\|\theta\|_1$

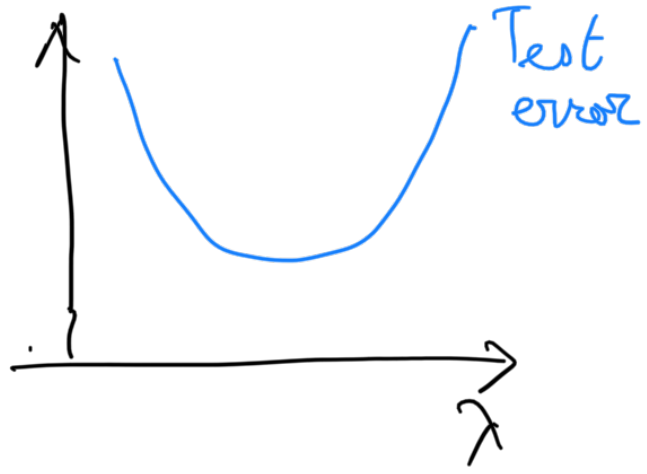
Soft SVM : minimize

$$\frac{\lambda}{2}\|\theta\|_2^2 + \text{Hinge Loss}(\theta, s).$$

Choosing λ:

- When $\lambda = 0$, regularized cost fn is same as old.
- When λ is high, reducing norm of θ is of highest priority.



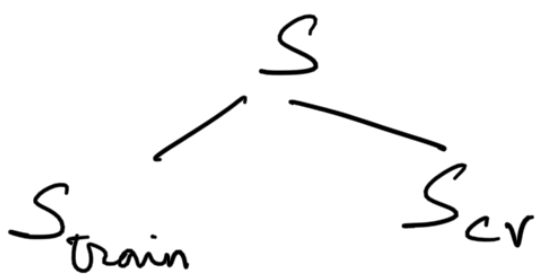


How do we choose hyperparameters, models

Different learning models.

- Dare different values of hyperparameters
- Dare different hypotheses (different degrees of polynomial fns, ...)
- Dare different learning algorithms

Cross Validation : Set of training examples S



For each model M

- Train M on S_{train} ←
- Compute error of M on S_{cv}

Pick the model that has lowest error on S_{cv} → M_*

→ Train M_* on the entire S. ←

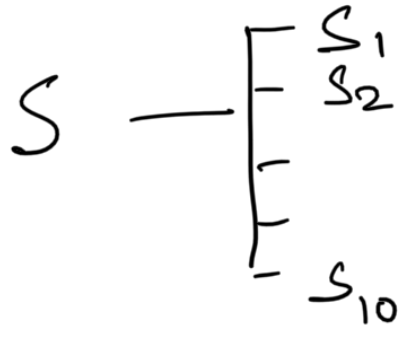
In practice S — S_{train}, S_{cv} } To identify the right Model & find hypo.
S_{test} → Used to estimate test error of hypo.

Typically, S_{train} 70%, S_{cv} 30%

S_{train} 60%, S_{cv} 20%, S_{Test} 20%.

$S \sim$ 1 million examples.

k-fold Cross-Validation ($k \sim 10$)



For each model M

For each $i \in \{1, \ldots k\}$

Train M on $S_1 \cup S_2 \cdots \cup S_{i-1} \cup S_{i+1} \cup \cdots S_k$

Measure error of M on S_i

Average of errors for S_i used as cross validation set.

Pick M_* that has lowest ave error.

+ Used a smaller cross validation set.
- Multiple trainings of a given model.

Leave-One-Out Cross Validation

k-fold cross validation where

$k = n$ (n - size of S)

Feature Selection

Pick a subset of features that are relevant for a problem.

Forward Search.

- Split S into S_{train}, S_{cv}.
- Start $F = \emptyset$
- Repeat

  For each feature $x_i \notin F$

  Train algo $F \cup \{x_i\}$ features

  Compute error of hyp on S_{cv}

  Pick feature x_i that results in lowest error and add to F.