Weekly Summary

CS60020 : Foundations of Algorithm Design and Machine Learning

April 12, 2025- Thakare Vedant Sharadrao(24BM6JP57)

Contents

1 Vapnik-Chervonenkis (VC) Dimension

The VC dimension is a measure of the capacity (or complexity) of a hypothesis class. It quantifies the maximum number of points that can be *shattered* by the hypothesis set. In other words, it indicates how many data points can be labeled in every possible way (i.e., 2^k possibilities for k points) by some hypothesis in the class.

Definition and Examples

Definition: A set of k points is said to be shattered by a hypothesis class $\mathcal{H}$ if, for every possible binary labeling of the k points, there exists a hypothesis $h \in \mathcal{H}$ that correctly classifies the points. The VC dimension, d_{vc}, is defined as the largest k such that there exists at least one set of k points that can be shattered by $\mathcal{H}$.

Examples:

- **Threshold Functions on the Real Line:** For functions of the form $h(x) = 1$ if $x > t$ and 0 otherwise, the VC dimension is 1.
- **Intervals on the Real Line:** Functions like $h(x) = 1$ if $a < x < b$ yield a VC dimension of 2.
- **Linear Classifiers in $\mathbb{R}^d$:** For perceptrons or linear classifiers, the VC dimension is $d + 1$.

Mathematical Derivation for Linear Classifiers

Consider a set of $d + 1$ points in $\mathbb{R}^d$. We augment each point with a bias term to obtain vectors of dimension $d + 1$ (i.e., each point becomes $[1, x_1, x_2, \ldots, x_d]$). Construct a matrix $X \in \mathbb{R}^{(d+1)\times(d+1)}$ from these points.

For any labeling vector $Y \in \{\pm 1\}^{d+1}$, the classifier is required to satisfy:

$$X \cdot W = Y.$$

Since X is invertible (if the points are in general position), we can solve for:

$$W = X^{-1}Y.$$

Thus, every binary labeling is achievable, and all $2^{(d+1)}$ possibilities can be realized, establishing that the VC dimension d_{vc} is at least $d + 1$.

[Insert Diagram: An illustration showing $d + 1$ points being shattered by a linear classifier and a demonstration that $d + 2$ points cannot be shattered due to linear dependencies.]

Upper Bound on VC Dimension

To show that $d_{\text{vc}} \leq d+1$, assume that there exists a set of $d+2$ points that can be shattered by linear classifiers. Due to the properties of linear dependence in $\mathbb{R}^d$, any set of $d+2$ points must be linearly dependent. This implies that there is at least one labeling configuration that cannot be separated by any hyperplane. Thus, the hypothesis class cannot shatter $d+2$ points.

2 Approximation vs. Generalization Trade-Off

The process of selecting a hypothesis from a hypothesis set $\mathcal{H}$ involves balancing two conflicting goals:

1. **Approximation:** The ability of the model to closely fit the training data and approximate the true underlying function f.
2. **Generalization:** The ability of the model to perform well on unseen data, thereby minimizing the out-of-sample error E_{out}.

The Trade-Off

When $\mathcal{H}$ is too simple, the model is unable to capture the complexities of f, resulting in a high in-sample error E_{in} (high bias). Conversely, when $\mathcal{H}$ is overly complex, the model may fit the training data very well (low E_{in}) but generalize poorly due to overfitting (high variance), leading to a high E_{out}.

A standard bound on the generalization error is expressed as:

$$E_{out}(h) \leq E_{in}(h) + \Omega,$$

where Ω represents a penalty term that increases with the complexity of the hypothesis class.

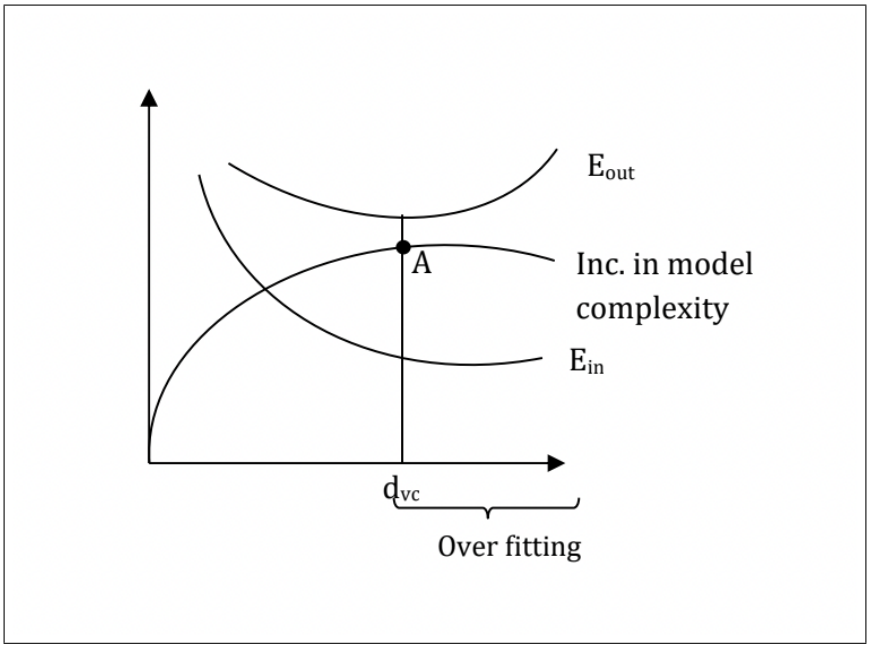

Figure 1: Trade-off between model complexity and generalization error. Point A represents the optimal model complexity where the balance between underfitting (high E_{in}) and overfitting (high E_{out}) is achieved. Increasing complexity beyond this point leads to overfitting.

3 Bias-Variance Decomposition

Understanding the error of a learning algorithm involves decomposing the expected out-of-sample error into bias and variance components, known as the bias-variance trade-off.

Mathematical Formulation

For a given input x, let:

- $g_D(x)$ denote the hypothesis learned from a dataset D.
- $\overline{g}(x)$ be the average of $g_D(x)$ over all possible training datasets.
- $f(x)$ be the true underlying function.

Then, the expected error on unseen data is:

$$E_{out}(g_D) = \mathbb{E}_D \left[\left(g_D(x) - f(x) \right)^2 \right].$$

This error can be decomposed as:

$$E_{out}(g_D) = \underbrace{\mathbb{E}_D \left[\left(g_D(x) - \overline{g}(x) \right)^2 \right]}_{\text{Variance}} + \underbrace{\left(\overline{g}(x) - f(x) \right)^2}_{\text{Bias}^2}.$$

Interpretation

- **Bias:** Measures the error due to erroneous assumptions in the learning algorithm.
- **Variance:** Quantifies the variability of the model prediction for a given data point x due to variations in the training set.

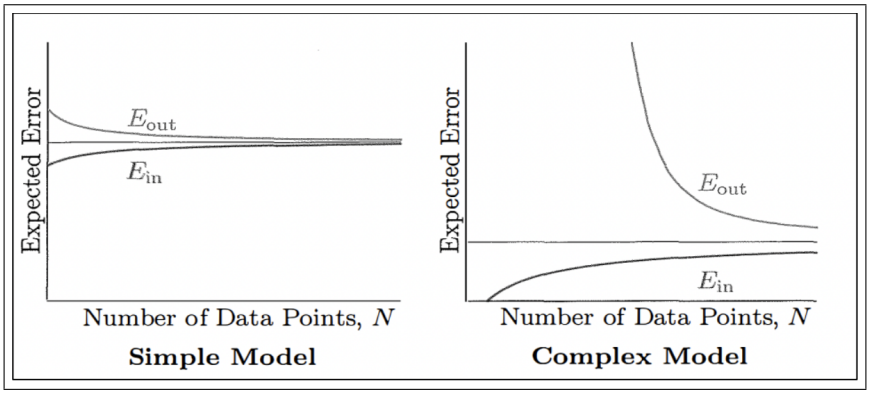

Figure 2: Bias-Variance Trade-off. The diagram illustrates how simple models tend to have high bias and low variance, while complex models exhibit low bias but high variance. The optimal balance minimizes the total expected error E_{out}.

4 Overfitting and Regularization

Overfitting

Overfitting occurs when a model learns not only the underlying patterns in the training data but also the noise, resulting in very low training error E_{in} but high out-of-sample error E_{out}.

There are two primary sources of noise that can lead to overfitting:

- **Stochastic Noise:** Random fluctuations in the data (e.g., measurement errors).
- **Deterministic Noise:** Errors arising from the model's inability to capture the true function f due to an overly complex or overly simplistic hypothesis class.

Regularization Techniques

Regularization adds a penalty to the loss function in order to discourage overly complex models. For example, in regression, the regularized loss function is often:

$$\min_{W} \sum_{i=1}^{N} \left(y_i - h(x_i, W)\right)^2 + \lambda \|W\|^2,$$

where λ is the regularization parameter. This term penalizes large weight values, reducing model variance at the cost of introducing a little bias.

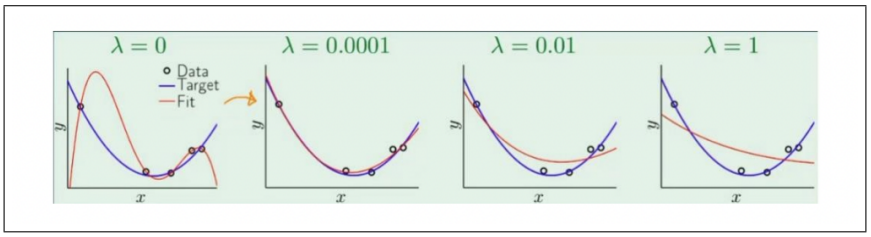

Figure 3: Impact of regularization strength λ on model complexity. As λ increases from 0.0001 to 1, the model becomes simpler and less prone to overfitting, but may underfit if regularization is too strong.

5 Validation and Cross-Validation Techniques

Validation is used to estimate the true generalization error E_{out}, which is not directly observable during training.

Validation Set Approach

The dataset is split into a training set and a validation set. The model is trained on the training set, and its performance is evaluated on the validation set. The validation error is computed as:

$$\text{Validation Error} = \frac{1}{K} \sum_{i=1}^{K} e\big(h(x_i), y_i\big),$$

where $e(h(x_i), y_i)$ is the error measurement for each instance.

K-Fold Cross-Validation

To obtain a more stable estimate of E_{out}, K-Fold Cross-Validation is used. The data is partitioned into K folds. For each fold k:

1. Train the model on $K-1$ folds.
2. Validate on the remaining fold.

The overall cross-validation error is then:

$$E_{CV}(h) = \frac{1}{K} \sum_{k=1}^{K} E_{val}^{(k)}(h).$$

6 Learning Curves and Model Selection

Learning curves plot the training error (E_{in}) and validation error (E_{out}) as a function of the number of training examples.

Observations from Learning Curves

- With a small training set, the model may overfit, yielding very low E_{in} but a high E_{out}.
- As the training set size increases, E_{in} typically increases slightly, while E_{out} decreases, indicating improved generalization.

A typical split is 80% for training and 20% for validation.

7 Model Selection and Hypothesis Space

Selecting the best model involves choosing a hypothesis from the set $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$ that both fits the training data well and generalizes to unseen data.

The Role of the Hypothesis Set

- **Too Simple:** May lead to high bias, as the model cannot capture the complexity of f.
- **Too Complex:** May lead to high variance, as the model fits the noise in the data.

The generalization error bound from VC theory provides:

$$E_{out}(h) \leq E_{in}(h) + \Omega,$$

where Ω depends on the complexity (such as the VC dimension) of $\mathcal{H}$.

8 Regularization and Its Mathematical Foundations

Regularization techniques constrain the hypothesis space to prevent overfitting. A common example is ridge regression (L2 regularization), where the objective function is:

$$\min_{W} \sum_{i=1}^{N} (y_i - h(x_i, W))^2 + \lambda \|W\|^2.$$

Key Concepts

- **Penalty Term:** $\lambda \|W\|^2$ increases with large weights, discouraging an overly flexible model.
- **Trade-Off:** A high λ reduces variance but may increase bias.

9 Detailed Discussion on Overfitting

Overfitting occurs when the model fits the noise in the training data along with the underlying pattern, leading to poor performance on unseen data.

Causes of Overfitting

- **Excessive Model Complexity:** Too many degrees of freedom allow the model to learn the noise.
- **Insufficient Training Data:** A small dataset may not represent the true data distribution, resulting in overfitting.
- **Types of Noise:**
  - **Stochastic Noise:** Random variations (e.g., measurement errors).
  - **Deterministic Noise:** Arising from a mismatch between the true function and the chosen hypothesis class.

Remedies for Overfitting

- **Regularization:** Introduce a penalty for complexity.
- **Cross-Validation:** Use validation sets to estimate E_{out} reliably.
- **Early Stopping:** Halt the training process when the validation error begins to increase.

10 Validation Techniques and Cross-Validation

Hold-Out Validation

The dataset is split into a training set and a validation set. The model is trained on the training set, and the validation error is computed as:

$$\text{Validation Error} = \frac{1}{K} \sum_{i=1}^{K} e\big(h(x_i), y_i\big),$$

which serves as an estimate for E_{out}.

K-Fold Cross-Validation

In K-Fold Cross-Validation, the dataset is divided into K folds. For each fold k:

- Train on $K - 1$ folds.
- Validate on the remaining fold.

The overall error estimate is:

$$E_{CV}(h) = \frac{1}{K} \sum_{k=1}^{K} E_{val}^{(k)}(h).$$

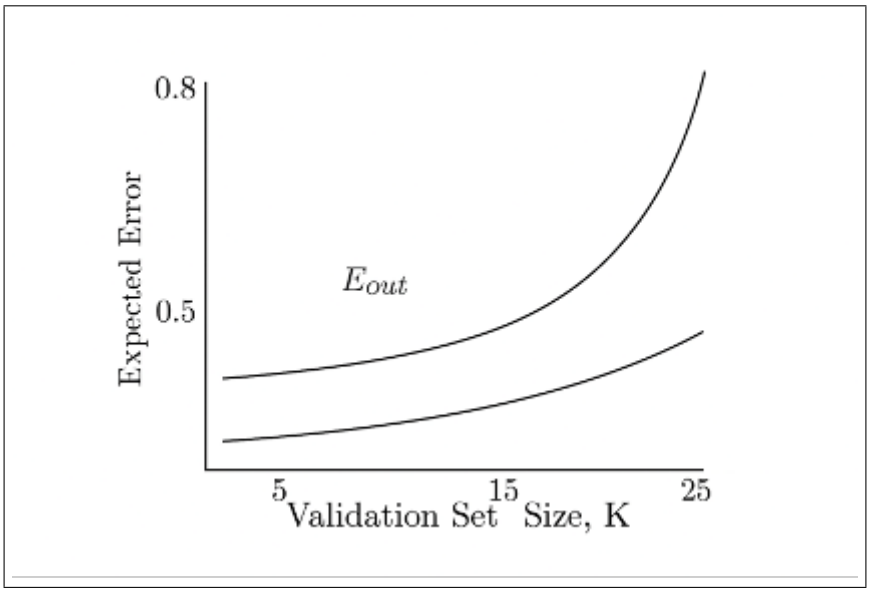

Figure 4: Variation of validation error E_{out} with validation set size K in K-fold cross-validation. As the size increases, the expected validation error may increase due to reduced training data in each fold.

11 Learning Curves

Learning curves graph the training error (E_{in}) and validation error (E_{out}) as a function of the number of training examples.

Observations from Learning Curves

- With a small training set, the model tends to overfit (very low E_{in}, high E_{out}).
- As the training set grows, E_{in} may rise slightly, while E_{out} generally decreases.

A typical split is 80% for training and 20% for validation.

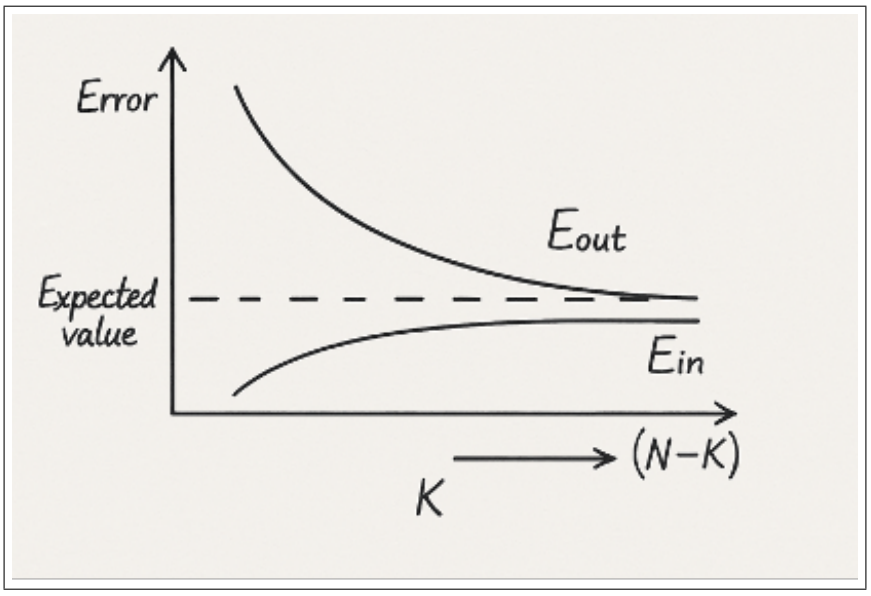

Figure 5: Learning curve illustrating the behavior of training error (E_{in}) and generalization error (E_{out}) as a function of training set size (K). As K increases, E_{in} slightly increases while E_{out} decreases, reflecting improved generalization with more training data.

12 Practical Implications in Model Selection

When selecting models, the following considerations are crucial:

- **Model Complexity:** The hypothesis class must be flexible enough to approximate f but not so complex as to overfit.
- **Validation Performance:** Reliable estimates of E_{out} through cross-validation help in choosing the best model.
- **Regularization:** Helps to control the trade-off between bias and variance, leading to better generalization.

13 Summary of Mathematical Formulations

For quick reference, here are the key equations covered in this document:

1. **VC Dimension for Linear Classifiers:**

$$d_{vc} = d + 1.$$

2. **Perceptron Equation:**

$$X \cdot W = Y \quad \Rightarrow \quad W = X^{-1}Y.$$

3. **Generalization Bound (VC Analysis):**

$$E_{out}(h) \leq E_{in}(h) + \Omega.$$

4. **Bias-Variance Decomposition:**

$$E_{out}(g_D) = \mathbb{E}_D\left[\left(g_D(x) - \overline{g}(x)\right)^2\right] + \left(\overline{g}(x) - f(x)\right)^2.$$

5. **Regularized Error Function (Ridge Regression):**

$$\min_W \sum_{i=1}^{N} \left(y_i - h(x_i, W)\right)^2 + \lambda \|W\|^2.$$

6. **Cross-Validation Error Estimate:**

$$E_{CV}(h) = \frac{1}{K} \sum_{k=1}^{K} E_{val}^{(k)}(h).$$