DDA5001 Machine Learning

Overfitting (Part I)

Xiao Li

School of Data Science
The Chinese University of Hong Kong, Shenzhen

Recap: Gradient Descent with Momentum

GD:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \nabla \mathcal{L}(\boldsymbol{\theta}_k)$$

A quite popular technique for accelerating gradient descent method is the momentum technique:

$$\boxed{\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \nabla \mathcal{L}(\boldsymbol{\theta}_k) + \beta_k \underbrace{(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1})}_{\text{momemtum}}}$$

An equivalent form:

$$\boxed{\begin{aligned} \boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k - \boldsymbol{m}_k \\ \boldsymbol{m}_k &= \mu_k \nabla \mathcal{L}(\boldsymbol{\theta}_k) + \beta_k \boldsymbol{m}_{k-1} \end{aligned}}$$

- Each iteration takes nearly the same time cost as GD.
- Widely used in practice, notably in Adam algorithm.

Recap: Nesterov's Acceleration

GD:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \nabla \mathcal{L}(\boldsymbol{\theta}_k)$$

Another very useful technique to accelerate GD is Nesterov's accelerated gradient descent (AGD):

$$\boxed{\begin{aligned} \boldsymbol{\theta}_{k+1} &= \boldsymbol{w}_k - \mu_k \nabla \mathcal{L}(\boldsymbol{w}_k) \\ \boldsymbol{w}_k &= \boldsymbol{\theta}_k + \frac{k-1}{k+2}(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}) \end{aligned}}$$

- Each iteration takes nearly the same time cost as GD.
- Widely used in practice.

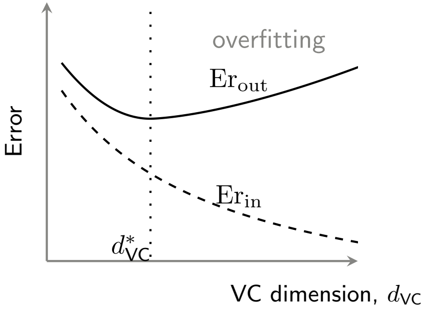
Overfitting

Validation

What is Overfitting?

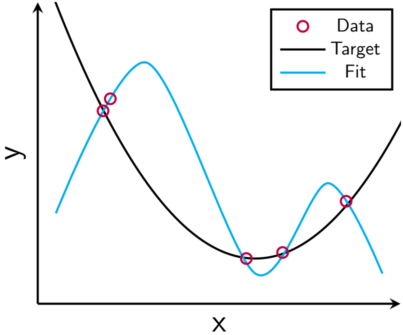
Overfitting

Fitting the data more than is needed



- One possible reason: $\mathcal{H}$ is more complex than is needed.
- It means fitting the observed data well no longer indicates that we will get a small out-of-sample error, and may actually lead to the opposite effect. The main case: Small training error (small $\mathrm{Er}_{\mathrm{in}}$) but bad generalization (large $\mathrm{Er}_{\mathrm{out}}$).

Overfitting: Example



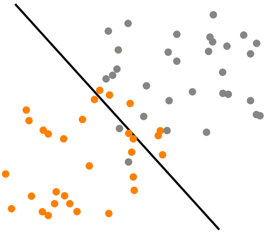
- Quadratic target function g.
- 5 data points with noise.
- 4-th order polynomial hypothesis set $\mathcal{H}$

$$\mathrm{Er}_{\mathrm{in}} = 0, \quad \mathrm{Er}_{\mathrm{out}} \text{ is huge}$$

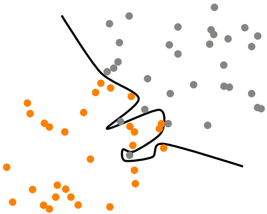
- The model uses its additional degrees of freedom to fit non-desirable pattern in the data (for example, noise), yielding a final learned model that is inferior.

Overfitting: Example

Linear: nearly appropriate



Highly nonlinear: overfit



- We have linearly separable data plus noise, it becomes slightly non-linearly separable.
- Linear classifier

$$\mathrm{Er}_{\mathrm{in}} \neq 0, \quad \mathrm{Er}_{\mathrm{out}} \text{ is appropriate}$$

- Highly nonlinear hypothesis set $\mathcal{H}$

$$\mathrm{Er}_{\mathrm{in}} = 0, \quad \mathrm{Er}_{\mathrm{out}} \text{ is larger}$$

Overfitting: Catalysts

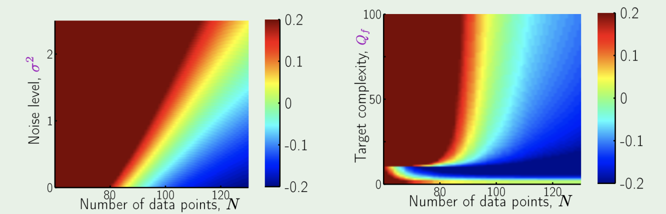


Figure: Color means the overfitting measure $\mathrm{Er_{out}}(f_{10}) - \mathrm{Er_{out}}(f_2)$.

Overfitting: Catalysts

- **Number of training samples** increase, overfitting decreases.
- **Noise** in data increase, overfitting increases.
- **Target model complexity** increases, overfitting increases.

$\rightsquigarrow$ Our HW2 will study the Catalysts for overfitting.

- One important approach to avoid overfitting is validation.
- Another one is regularization.

Overfitting

Validation

Validation as A Cure for Overfitting

From VC generalization analysis, we have

$$\mathrm{Er}_{\mathrm{out}}(f) \leq \mathrm{Er}_{\mathrm{in}}(f) + \text{overfit penalty.}$$

- Generalization error is large in the overfitting regime, hence it becomes the "overfit penalty".
- The ideal case is to minimize $\mathrm{Er}_{\mathrm{out}}$ directly, which is, however, not available.
- Can we turn to estimate the $\mathrm{Er}_{\mathrm{out}}$?

Validation technique tries to estimate the out-of-sample error for eliminating overfitting:

$$\underbrace{\mathrm{Er}_{\mathrm{out}}(f)}_{\text{validation estimates this quantity}} \leq \mathrm{Er}_{\mathrm{in}}(f) + \text{overfit penalty.}$$

Where we have estimated $\mathrm{Er}_{\mathrm{out}}$? Q5 in HW1. Why is it possible? Will justify later in the understanding of validation error.

Applications of Validation

Hyper-parameters: Something that are NOT automatically determined by the learning algorithm.

- Complexity of $\mathcal{H}$.
- Number of iterations in learning algorithm, i.e., stopping criterion.
- Learning rate.
- Regularization parameter λ (in our later lecture).
- $\ldots$

In many cases, we have one (or more) hyper-parameters. The value chosen for hyper-parameters has a significant impact on the algorithm's output.

Model selection:

- The problem of selecting values for hyper-parameters is called model selection.
- The validation is used for model selection.

The Validation Dilemma

- What we have? Training data.
- Can we use the training data to estimate the $\mathrm{Er}_{\mathrm{out}}$ and then select hyper-parameters?
- These hyper-parameters usually control the balance between underfitting and overfitting.
- Thus, it is not appropriate to let the training data influence the selection of hyper-parameters, as this almost always leads to overfitting.

Example: If we let the training data determine the degree in polynomial regression ($\mathcal{H}$ is the set of all polynomial functions with degree less than d), we will just end up choosing the maximum degree and doing interpolation (overfitting).

$\rightsquigarrow$ We need some other data for estimating $\mathrm{Er}_{\mathrm{out}}$.

The Idea of Validation: Split Training Data

The idea

Split the training set to another 'training set' and a validation set.

Validation

After learning $f_{\boldsymbol{\theta}'} \in \mathcal{H}$ based on the new 'training set', use validation set to estimate $\mathrm{Er}_{\mathrm{out}}(f_{\boldsymbol{\theta}'})$.

Why estimate $\mathrm{Er}_{\mathrm{out}}(f_{\boldsymbol{\theta}'})$? We can see if the learned $f_{\boldsymbol{\theta}'} \in \mathcal{H}$ is good or not in terms of (the estimated) out-of-sample error.

Recall the goal of machine learning is to make the out-of-sample error small. If $\mathrm{Er}_{\mathrm{out}}(f_{\boldsymbol{\theta}'})$ is small, it means we are nearly done. If not, it means we have a bad learned model, either underfitting or overfitting.

The Validation Set

Split training data set:

$$\mathcal{S} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$$

$$\mathcal{S}_{\text{val}} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_k, y_k)\} \qquad \mathcal{S}_{\text{train}} = \{(\boldsymbol{x}_{k+1}, y_{k+1}), \ldots, (\boldsymbol{x}_n, y_n)\}$$

Validation set — Training set

- Data set: $\mathcal{S} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, size n.
- Validation set: $\mathcal{S}_{\text{val}} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_k, y_k)\}$, size k.
- New training set: $\mathcal{S}_{\text{train}} = \{(\boldsymbol{x}_{k+1}, y_{k+1}), \ldots, (\boldsymbol{x}_n, y_n)\}$, size $n - k$.

Validation Error

Use the validation set to form an estimate (we use f' to represent $f_{\boldsymbol{\theta}'}$ to ease notation)

$$\mathrm{Er}_{\mathrm{val}}(f') = \frac{1}{k} \sum_{i \in \mathcal{S}_{\mathsf{val}}} e(f'(\boldsymbol{x}_i), y_i)$$

How well $\mathrm{Er}_{\mathrm{val}}(f')$ approximates $\mathrm{Er}_{\mathrm{out}}(f')$?

- In expectation

$$\mathbb{E}\left[\mathrm{Er}_{\mathrm{val}}(f')\right] = \frac{1}{k} \sum_{i \in \mathcal{S}_{\mathsf{val}}} \mathbb{E}\left[e(f'(\boldsymbol{x}_i), y_i)\right] = \mathrm{Er}_{\mathrm{out}}(f')$$

  This implies that the validation error is an unbiased estimation of the out-of-sample error of f'.

However, unbiased estimation is a too weak guarantee. Can we have more?

Validation Error Approximates Out-of-Sample Error

Mimicking the derivation of the generalization result for a single fixed hypothesis, we can utilize the Hoeffding's inequality to show that

$$\mathrm{Er}_{\mathrm{out}}(f') \leq \mathrm{Er}_{\mathrm{val}}(f') + \mathcal{O}\left(\frac{1}{\sqrt{k}}\right).$$

- The key of this result is that we only need to derive the bound for a single fixed hypothesis f', thus the derivation is exactly the same as the fixed f generalization result. No union bounds needed.
- One key feature is that $\mathrm{Er}_{\mathrm{val}}(f')$ is computable.
- This means that once we have enough validation data, i.e., k is large, we get good estimation of $\mathrm{Er}_{\mathrm{out}}(f')$ by $\mathrm{Er}_{\mathrm{val}}(f')$. Then, we will know whether the model f' is good or not.
- This result applies to test data as well, i.e.,

$$\mathrm{Er}_{\mathrm{out}}(f') \leq \mathrm{Er}_{\mathsf{test}}(f') + \mathcal{O}(1/\sqrt{k})$$

  if we have k test data points. The reasoning is exactly the same.

Any remaining issues?

Trade-off in Validation

- Where is k from? Where is f' from?

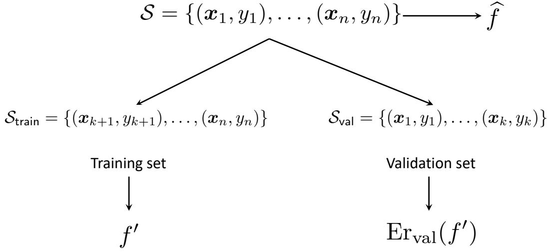
$$\underbrace{n}_{\text{number of data}} = \underbrace{k}_{\text{number of validation data}} + \underbrace{n-k}_{\text{number of new training data}}$$

- $\mathrm{Er}_{\mathrm{val}}(f')$ is $\mathcal{O}\left(1/\sqrt{k}\right)$-close to $\mathrm{Er}_{\mathrm{out}}(f')$, while f' is learned by using $n-k$ new training data.
- Small k: Bad estimate of $\mathrm{Er}_{\mathrm{out}}(f')$.
- Large k: Good estimate of $\mathrm{Er}_{\mathrm{out}}(f')$, but bad f' (since few training samples).

Rule of thumb: $k = n/5$.

Restoring: k is Put Back to n

- Validation is to estimate $\mathrm{Er}_{\mathrm{out}}(f')$ using $\mathrm{Er}_{\mathrm{val}}(f')$, must we output f' as the learned model?
- No. It is mainly used to know how to choose hyper-parameters such as the best possible hypothesis and learning rate.
- Hence, after we have used the validation set to estimate the out-of-sample error, re-train on the whole data set.



By using the previous bound, we have

$$\mathrm{Er}_{\mathrm{out}}(\widehat{f}) \leq \mathrm{Er}_{\mathrm{out}}(f') \leq \mathrm{Er}_{\mathrm{val}}(f') + \mathcal{O}\left(\frac{1}{\sqrt{k}}\right).$$

The gray part is a reasonable guess (not proof) from VC analysis.

Model Selection

Validation for Model Selection

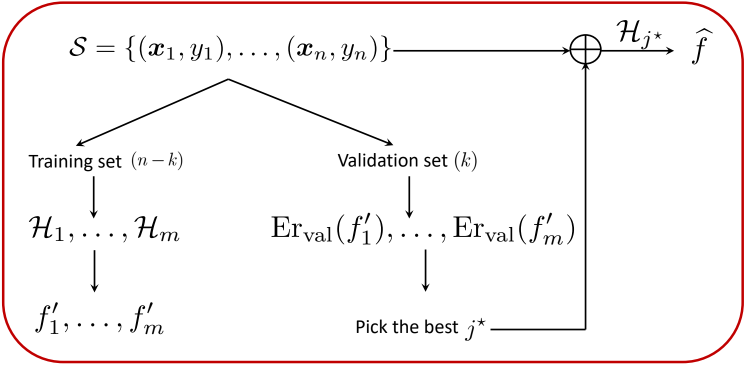
We have discussed how to estimate $\mathrm{Er}_{\mathrm{out}}(f')$ using validation. It can be important to estimate how good our f' will perform on the test data.

- Indeed, validation can be used to guide the learning process systematically, by the way of model selection.
- Recall that model selection is to select hyper-parameters. These could be the choice between a linear model and a nonlinear model, the choice of order of polynomials in a model, the choice of learning rate, or any other choice that affects the learning process.
- Setup: Suppose we have m candidate hypothesis spaces (can also be m different learning rate choices, etc)

  $$\mathcal{H}_1, \ldots, \mathcal{H}_m.$$

  We can use the validation set to estimate the out-of-sample error by using $\mathrm{Er}_{\mathrm{val}}(f'_j)$ for each f'_j learned from these model spaces.
- Selection: Choose $j^\star$ such that $\mathrm{Er}_{\mathrm{val}}(f'_{j^\star}) \leq \mathrm{Er}_{\mathrm{val}}(f'_j)$ for all j.
- Restoring: Train f on the whole set using model space $\mathcal{H}_{j^\star}$, get $\widehat{f}$.

Validation for Model Selection: illustration



Generalization Error of Model Selection

The model selection process gives a new hypothesis space consists of

$$\mathcal{H}_{\text{val}} = \{f_1', f_2', \ldots, f_m'\}.$$

Model selection chooses one from $\mathcal{H}_{\text{val}}$ that achieves the smallest validation error.

- This process is equivalent to learn a model from $\mathcal{H}_{\text{val}}$ using the validation set, where Er_{val} is the "in-sample" error.
- This setting allows us to apply the generalization analysis for finite hypotheses space case, since $\mathcal{H}_{\text{val}}$ only consists of m hypotheses, i.e., $|\mathcal{H}_{\text{val}}| = m$.
- This gives

$$\boxed{\text{Er}_{\text{out}}(\widehat{f}) \leq \text{Er}_{\text{out}}(f'_{j^\star}) \ \leq \text{Er}_{\text{val}}(f'_{j^\star}) + \mathcal{O}\left(\sqrt{\frac{\log m}{k}}\right).}$$

Again, the gray part is a reasonable guess from VC analysis.

Validation vs. Testing

- We call this "validation", but how is it different from "testing"?
- Typically, validation is used to make learning choices, i.e., choosing hyper-parameters to avoid overfitting.

However,

The test data can never influence the training phase in any way.

If it impacts the learning process, i.e., which final $\widehat{f} \in \mathcal{H}$ we choose, then it is no longer a test set,

it becomes a validation set.

Validation Dilemma

What we would like to have

$$\mathrm{Er}_{\mathrm{out}}(\hat{f}) \approx \mathrm{Er}_{\mathrm{out}}(f') \approx \mathrm{Er}_{\mathrm{val}}(f')$$

$$\underset{\text{small } k}{\uparrow} \qquad \underset{\text{large } k}{\uparrow}$$

- All we need to do is set k so that it is simultaneously small and large...

Is it possible? Yes.

$\rightsquigarrow$ Next lecture: Cross validation and regularization.