Lecture 9 : Generalization Gap & Bias-variance decomposition

- We introduced E_{new} ← error on fresh unseen data

- Designing a method with small E_{new} is the central goal in supervised learning

- Cross-validation helps in estimating E_{new}

  (Unless one uses it for tuning hyperparameters)

- E_{new} can be further analyzed even more mathematically

error function

$E(y, \hat{y})$

Recall that

$$E_{train} = \frac{1}{N} \sum_{i=1}^{N} E\left(y^{(i)}, \hat{y}(\underline{x}^{(i)}; \mathcal{T})\right)$$

$$E_{new} = \int E\left(y^{*}, \hat{y}(\underline{x}^{*}; \mathcal{T})\right) \, p(\underline{x}^{*}, y^{*}) \, d\underline{x}^{*} \, dy^{*}$$

multi-dimensional integral

- Note the values of E_{train} and E_{new} were calculated while keeping the training set $\mathcal{T}$ fixed

- We now emphasize the fact that E_{train} and E_{new} are functions of the training set
    - As the training set $\mathcal{T}$ changes, the values of E_{train} and E_{new} changes too!

$$E_{train}(\mathcal{T}) = \frac{1}{N} \sum_{i=1}^{N} E\left(y^{(i)}, \hat{y}(\underline{x}^{(i)}; \mathcal{T})\right)$$

function of training data

$$E_{new}(\mathcal{T}) = \quad E\left(y^*, \hat{y}(\underline{x}^*; \mathcal{T})\right) \; p(\underline{x}^*, y^*) \; d\underline{x}^* \, dy^*$$

- We now emphasize the fact that E_{train} and E_{new} are functions of the training set
    - As the training set $\mathcal{T}$ changes, the values of E_{train} and E_{new} changes too!

$$E_{train}(\mathcal{T}) = \frac{1}{N} \sum_{i=1}^{N} E\left(y^{(i)}, \hat{y}(\underline{x}^{(i)}; \mathcal{T})\right)$$

function of training data

$$E_{new}(\mathcal{T}) = E\left(y^{*}, \hat{y}(\underline{x}^{*}; \mathcal{T})\right) \; p(\underline{x}^{*}, y^{*}) \; d\underline{x}^{*} \, dy^{*}$$

- So we introduce another level of abstraction
    - Training-data averaged versions of E_{new} and E_{train}

$$\bar{E}_{new} = \mathbb{E}_{\mathcal{T}}\left[E_{new}(\mathcal{T})\right]$$

$$\bar{E}_{train} = \mathbb{E}_{\mathcal{T}}\left[E_{train}(\mathcal{T})\right]$$

- $\mathbb{E}_{\mathcal{T}}$ denotes the expected value w.r.t. the training set $\mathcal{T} = \{\underline{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$
- We assume $\mathcal{T}$ consists of independent samples from $p(\underline{x}, y)$

$\bar{E}_{new}$ is the average E_{new} if we were to train the model multiple times on on different training datasets of size N. (Same goes for E_{train})

- $\bar{E}_{new} = \mathbb{E}_{\mathcal{T}}[E_{new}(\mathcal{T})]$

  $\bar{E}_{train} = \mathbb{E}_{\mathcal{T}}[E_{train}(\mathcal{T})]$

  - $\mathbb{E}_{\mathcal{T}}$ denotes the expected value w.r.t. the training set $\mathcal{T} = \{\underline{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$
  - We assume $\mathcal{T}$ consists of independent samples from $p(\underline{x}, y)$

$\bar{E}_{new}$ is the average E_{new} if we were to train the model multiple times on on different training datasets of size N.

- Why introduce these average quantities $\bar{E}_{new}$ and $\bar{E}_{train}$?

  - It is easier to reason about the average behavior $\bar{E}_{new}$ and $\bar{E}_{train}$ than about the errors E_{new} and E_{train} obtained when the model is trained on one specific training dataset $\mathcal{T}$
  - Insights from $\bar{E}_{new}$ are useful, even though we most often care about E_{new} in the end (as the training data is usually fixed)

- We have already seen that $E_{new} \not\approx E_{train}$
    - Usually, $E_{train} < E_{new}$ and $\boxed{\bar{E}_{train} < \bar{E}_{new}}$

      $\Downarrow$

      On an average, a method usually performs worse on new, unseen data than on training data

- A method generalizes well if it performs well on unseen data after training

- We call the difference between $\bar{E}_{new}$ and $\bar{E}_{train}$ as the generalization gap

  (more correctly as the expected generalization gap)

  $$\text{Generalization gap} = \bar{E}_{new} - \bar{E}_{train}$$

  is the difference between the expected performance on new unseen data and the expected performance on training data

- So $\bar{E}_{new}$ can be decomposed as:

  $$\boxed{\bar{E}_{new} = \bar{E}_{train} + \text{generalization gap}}$$

What affects the Generalization Gap?

- One can in some sense say, the more a method adapts to training data, the larger the generalization gap

- Vapnik-Chervonenkis (VC) dimension is a theoretical framework which assesses how much a method adapts to training data
  - The framework provides probabilistic bounds on generalization gap
  - However, the bounds are quite conservative
  - Therefore, the method is not very useful in practice

- Instead, we will only use vague terms $\underbrace{\text{"Model Complexity"}}_{\text{ability of a method to adapt to patterns in training data}}$ (or Model flexibility)
  - High complexity models — e.g. very deep decision trees, KNN with small 'K'
  - Low complexity models — e.g. linear regression, logistic regression

– Instead, we will only use vague terms "$\underbrace{\text{Model Complexity}}_{\text{ability of a method to adapt to patterns in training data}}$" (or Model flexibility)

- High complexity models — e.g. very deep decision trees, KNN with small 'K'
- Low complexity models — e.g. $\underbrace{\text{linear regression, logistic regression}}_{\text{basic parametric methods}}$

- For parametric models, the model complexity is related to the number of parameters, but is also affected by regularization techniques

$$\text{Generalization gap} = \bar{E}_{new} - \bar{E}_{train}$$

Model complexity ↑

$\bar{E}_{train}$ ↓

Generalization gap ↑

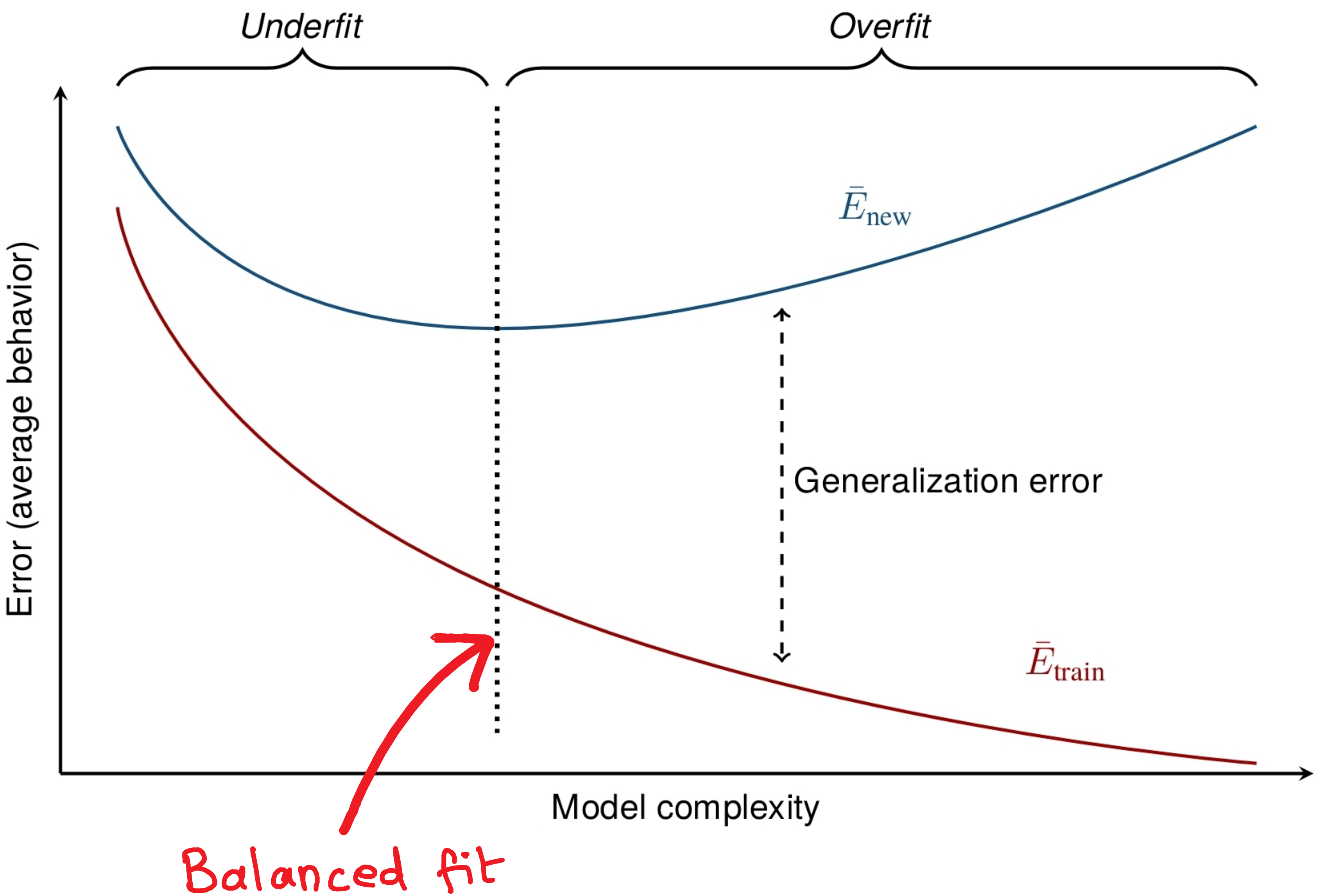
Model complexity ↓

$\bar{E}_{train}$ ↑

Generalization gap ↓

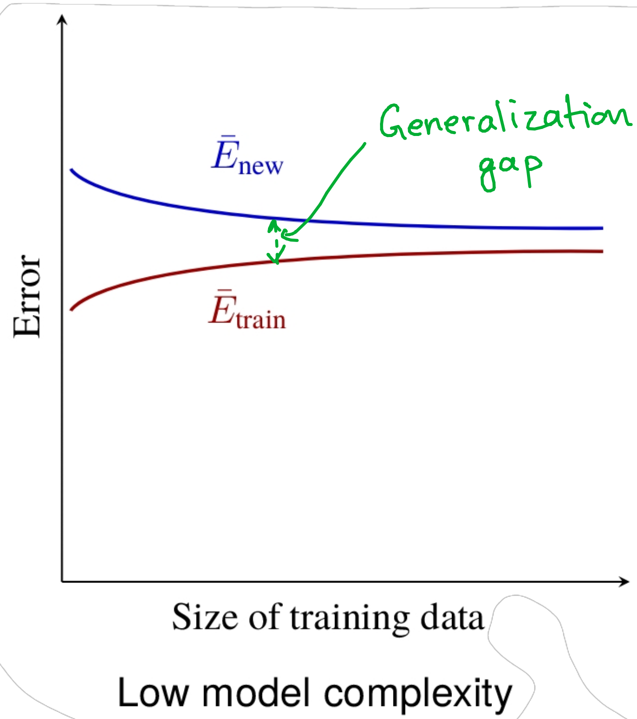
E_{new} usually attains a minimum at some intermediate complexity

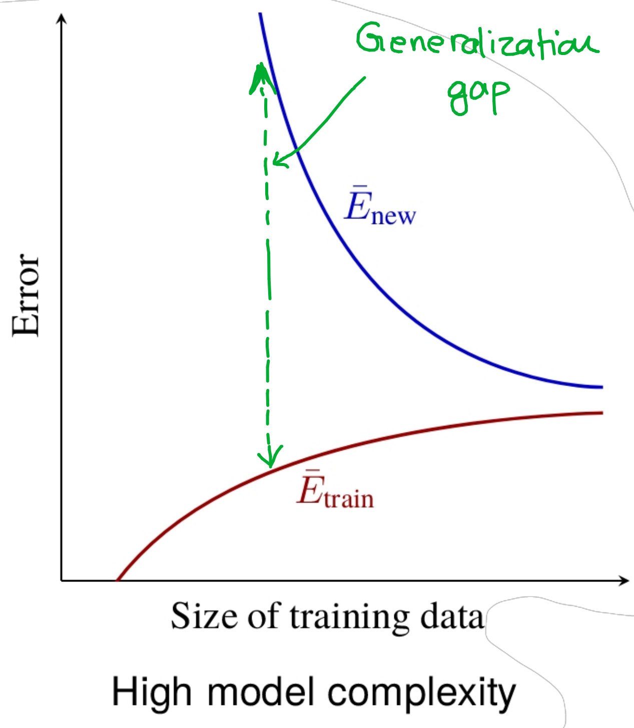
Behavior of $\bar{E}_{train}$ and $\bar{E}_{new}$ with model complexity



When we use cross-validation to select hyperparameters, we search for balanced fit

- Another important aspect is the size of training dataset N
- In general, the more the training data, the smaller the generalization gap
- $\bar{E}_{train}$ typically increases as N increases
    - since most models are unable to fit all training data points well, more so if there are too many of them

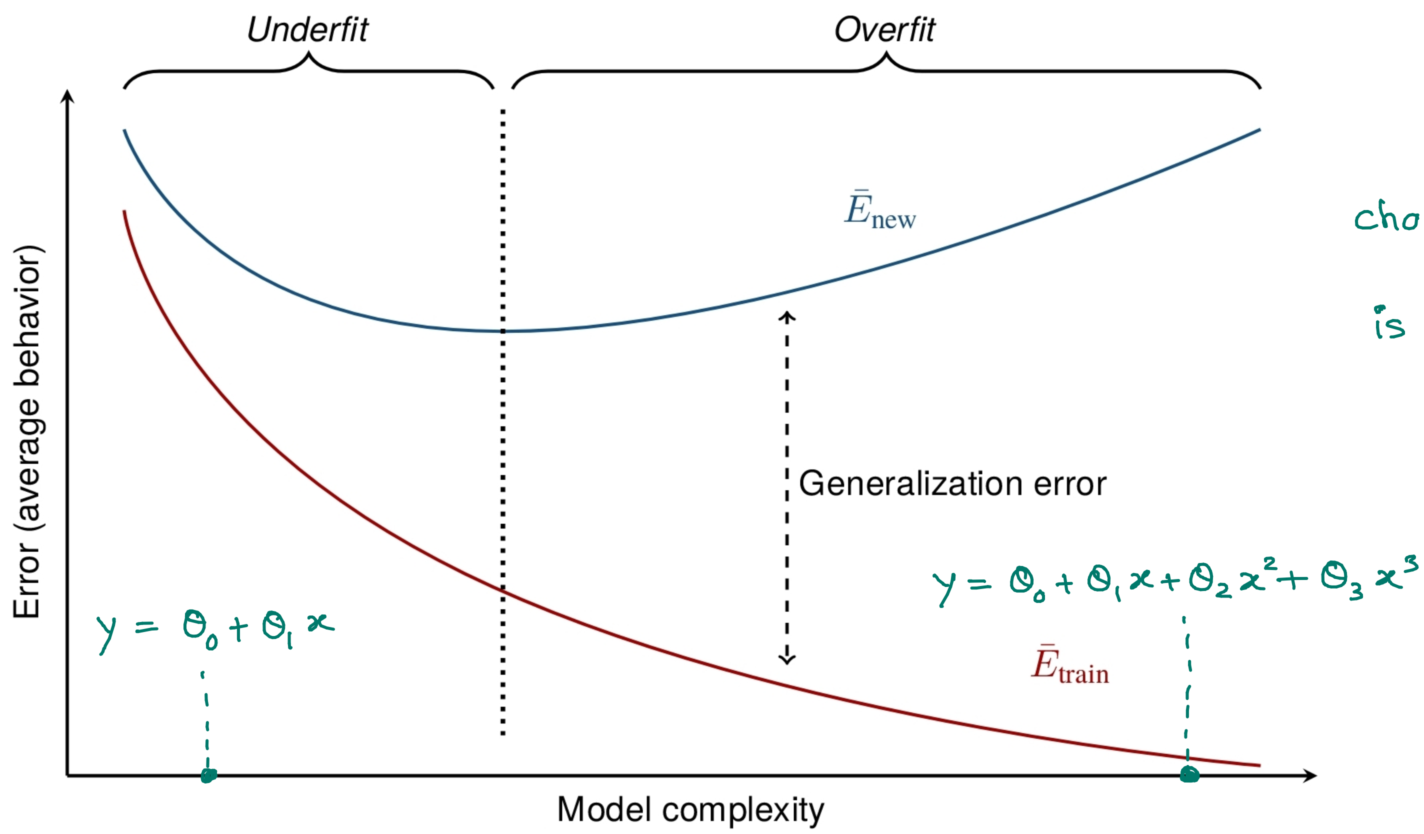




- For large enough data, a more complex model will attain a smaller $\bar{E}_{new}$ than a simpler model would

Shortcomings of using Model Complexity

- The figure below is relevant only when there is a single hyperparameter to choose



Say a hyperparameter to choose in polynomial regression is the order of polynomial

- However, when there are multiple hyperparameters (or multiple methods) to choose from, the above one-dimensional model complexity will not work well

- Lets take an example of jointly choosing the degree of polynomial regression and the regularization parameter
    - higher degree of polynomial $\Rightarrow$ more flexibility / complexity
    - more regularization $\Rightarrow$ less flexibility / complexity

- Example of a simulated problem: Sample data points from $p(x, y)$

Data Generation
- $N = 10$ data points
- Input $x \sim \text{UniformDist}(-5, 10)$
- $y = \min(0.1\, x^2, 3) + \epsilon$
- $\epsilon \sim \text{NormalDist}(0, 1)$

$$p(x, y) = \underbrace{p(y|x)}_{\text{Then sample } y \text{ given } x}\ \underbrace{p(x)}_{\text{First sample } x}$$

Now fit the input-output data $\{x^{(i)}, y^{(i)}\}_{i=1}^{10}$ using

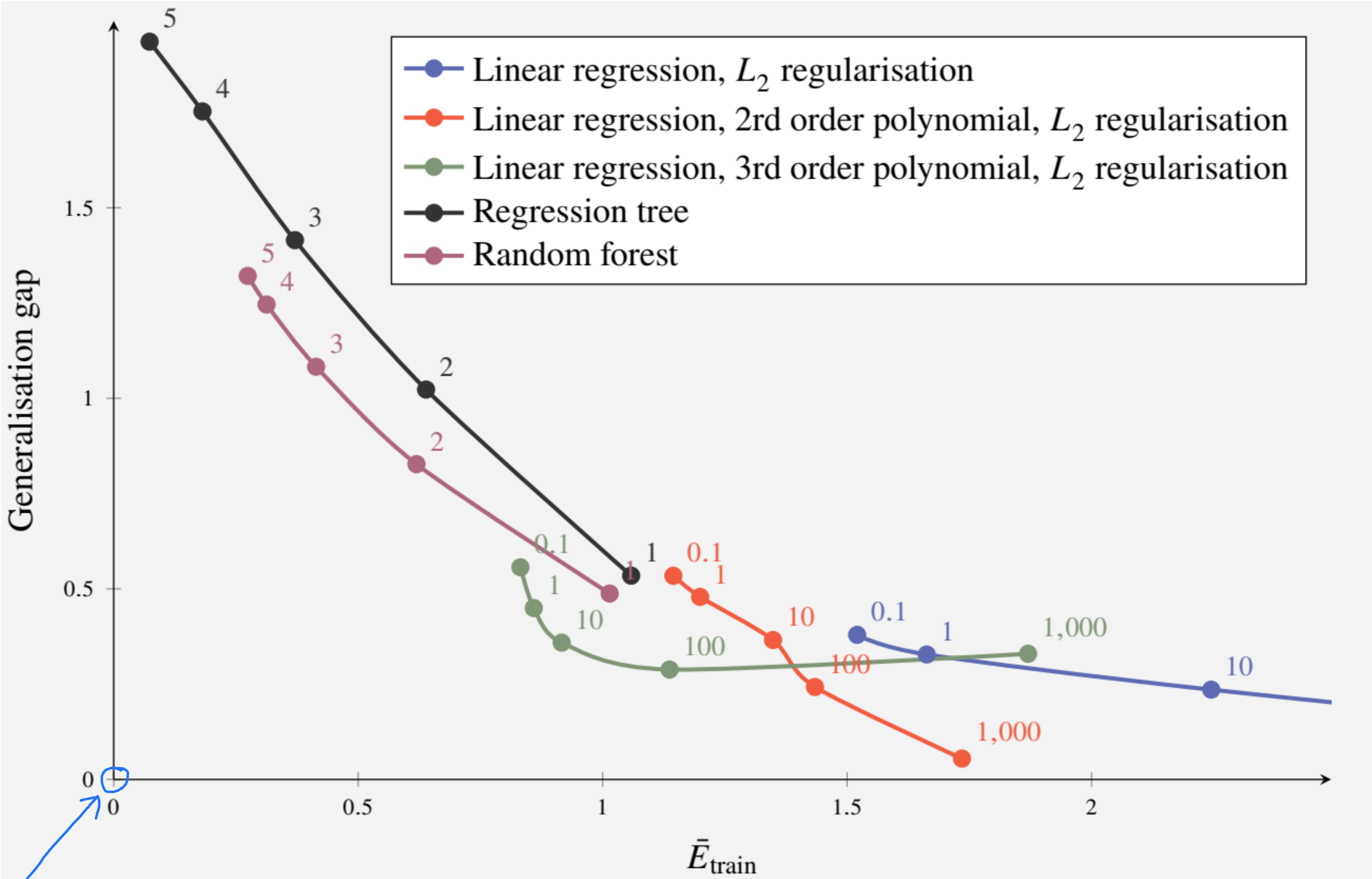
(a) Linear regression with L_2-regularization

(b) Linear regression with a quadratic polynomial and L_2-regularization

(c) Linear regression with a cubic polynomial and L_2-regularization

(d) Regression Tree, (e) A random forest with 10 regression trees

- For each of the methods, we try different values of hyperparameters (regularization parameter λ and tree depth) and compute $\bar{E}_{train}$ and generalization gap



In practice, we only have limited data and we cannot generate these plots!

Lowest $\bar{E}_{new}$ (= $\bar{E}_{train}$ + generalization gap)

Bias - Variance Decomposition of $\bar{E}_{new}$

– We now introduce another decomposition of $\bar{E}_{new}$

$$\boxed{\bar{E}_{new} = (\text{Bias})^2 + \text{Variance} + \text{irreducible noise}}$$

– Concept of BIAS and VARIANCE

- Consider an example: $z_0 \leftarrow$ true location of an object

  $z \leftarrow$ noisy GPS measurements of the location

  treated as a random variable

  it has some mean: $\mathbb{E}[z]$ → μ_z

  it has some variance: $\mathbb{E}[(z-\mu_z)^2]$

μ_z

- Bias describes the systematic error in the measurements z (possible offset)

$$\boxed{\text{BIAS}: \quad \mu_z - z_0}$$

- Variance describes how much the measurements vary (amount of noise in GPS measurements)

$$\boxed{\text{VARIANCE}: \quad \mathbb{E}[(z-\mu_z)^2] = \mathbb{E}[z^2] - \mu_z^2}$$

- Bias : $\mu_z - z_0$

  Variance : $\mathbb{E}[z^2] - \mu_z^2$

- Squared error between measurement and true value : $(z - z_0)^2$

- Expected squared error : $\underbrace{\mathbb{E}\left[(z - z_0)^2\right]}$

  (Averaged)

$$\Downarrow$$

$$= \mathbb{E}\left[\left((z - \mu_z) + (\mu_z - z_0)\right)^2\right]$$

$$= \mathbb{E}\left[(z-\mu_z)^2\right] + \mathbb{E}\left[(\overset{\text{constant}}{\mu_z} - \overset{\text{constant}}{z_0})^2\right] + 2\,\mathbb{E}\left[(z-\mu_z)(\overset{\text{constant}}{\mu_z} - \overset{\text{constant}}{z_0})\right]$$

$$= \underbrace{\mathbb{E}\left[(z-\mu_z)^2\right]}_{\text{variance}} + \underbrace{(\mu_z - z_0)^2}_{\text{bias}} + 2\,(\mu_z - z_0)\underbrace{\left(\mathbb{E}[z] - \mu_z\right)}_{0}$$

- In other words, the averaged squared error between z and z_0 is the sum of the squared bias and variance

- To obtain a small expected squared error, we have to consider both ↗ bias ↘ variance