Lecture 4

Model Selection and Evaluation

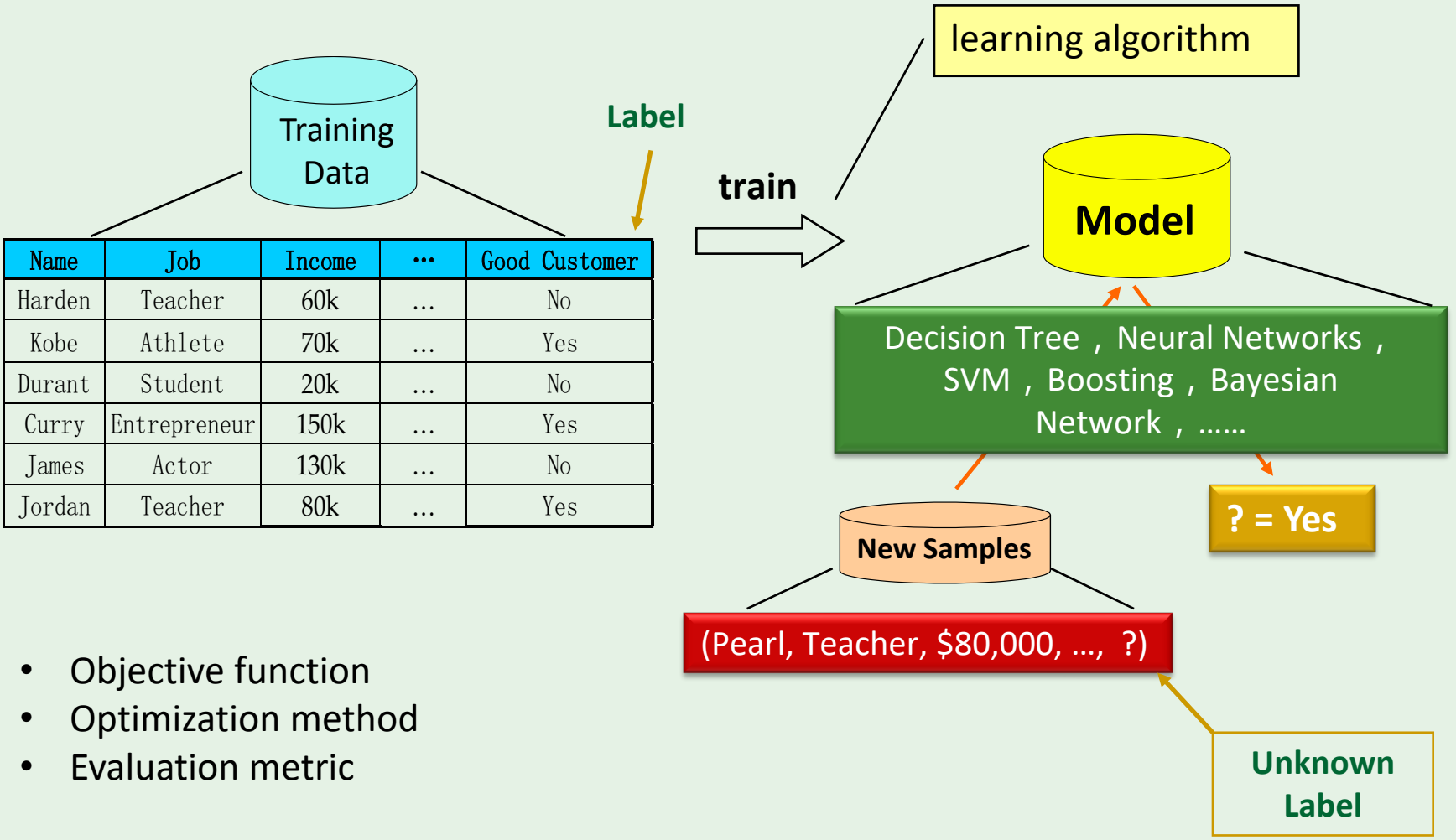
Outline

- Empirical Error and Overfitting
- Evaluation Methods
- Performance Measure
- Bias and Variance
- Further Reading

Types of Machine Learning Algorithm

- Supervised Learning
  - Given: training data + desired outputs (labels)
- Semi-Supervised Learning
  - Given: training data + a few desired outputs
- Unsupervised Learning
  - Given: training data (without desired outputs)
- Reinforcement learning
  - Rewards from sequence of actions

Machine Learning Pipline



Name	Job	Income	…	Good Customer
Harden	Teacher	60k	…	No
Kobe	Athlete	70k	…	Yes
Durant	Student	20k	…	No
Curry	Entrepreneur	150k	…	Yes
James	Actor	130k	…	No
Jordan	Teacher	80k	…	Yes

- Objective function
- Optimization method
- Evaluation metric

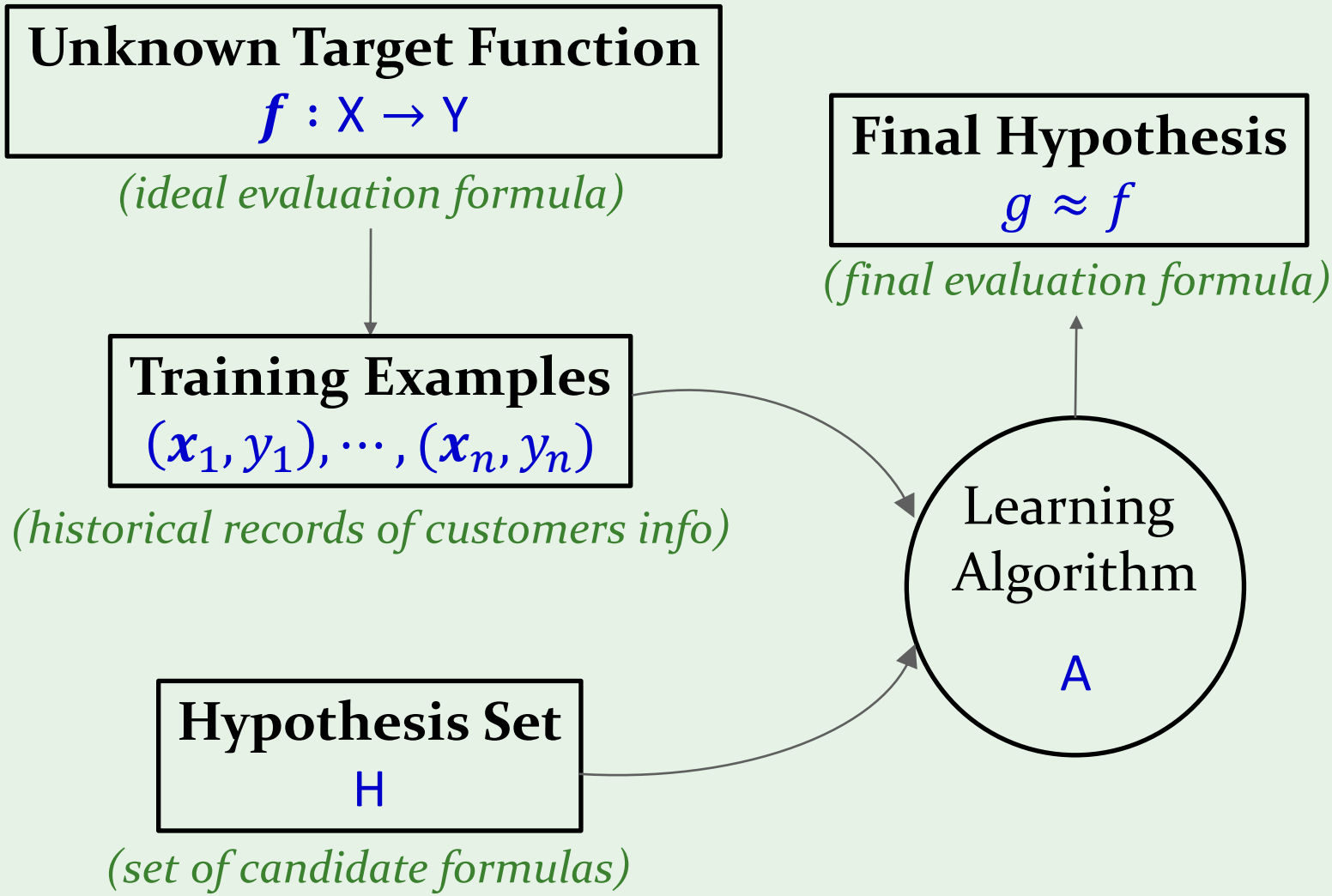
Components of Learning

Formalization

- **Input:** $\boldsymbol{x}$ *(customer info)*
- **Output:** y *(good/bad customer?)*
- **Target function:** $f : \mathcal{X} \to \mathcal{Y}$ *(ideal evaluation formula)*
- **Data:** $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n)$ *(historical records)*

- **Hypothesis:** $g : \mathcal{X} \to \mathcal{Y}$ *(formula to be used)*

Components of Learning



Empirical Error and Overfitting

- Error rate & Error:
  - **Error rate:** proportion of incorrectly classified samples $E = a/m$
  - **Error:** the difference between the output predicted by the learner and the ground-truth output
    - **Training (empirical) error:** on training set
    - **Test error:** on testing set
    - **Generalization error:** the error calculated on the new samples

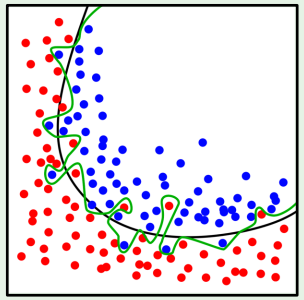
- Since the details of the new samples are unknown during the training phase, we resort to minimizing the empirical error in practice.
- Quite often, we obtain learners that perform well on the training set with a small or even zero empirical error, that is, 100% accuracy. However, are they the learners we need? Unfortunately, such learners are not good in most cases.

Empirical Error and Overfitting

- Overfitting:

  When the learner learns the training examples "too well", it is likely that some peculiarities of the training examples are taken as general properties that all potential samples will have, resulting in a reduction in generalization performance.

  - Regularize the training objective
  - Early stop

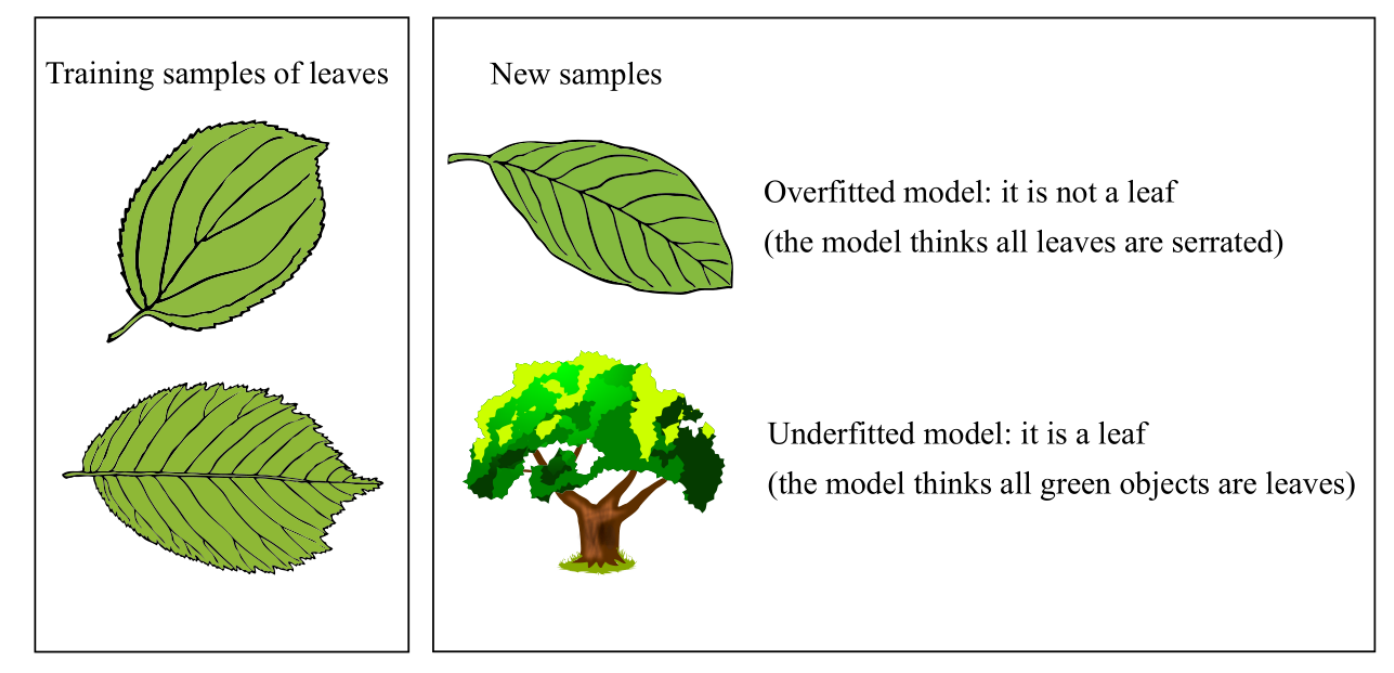


- Underfitting:

  The learner failed to learn the general properties of training examples.

  - Do more branching in decision tree learning
  - Adding more training epochs in neural network learning

Empirical Error and Overfitting



- **Overfitting**: some peculiarities of the training examples are taken as general properties that all potential samples will have.
- **Underfitting**: the learner failed to learn the general properties of training examples.

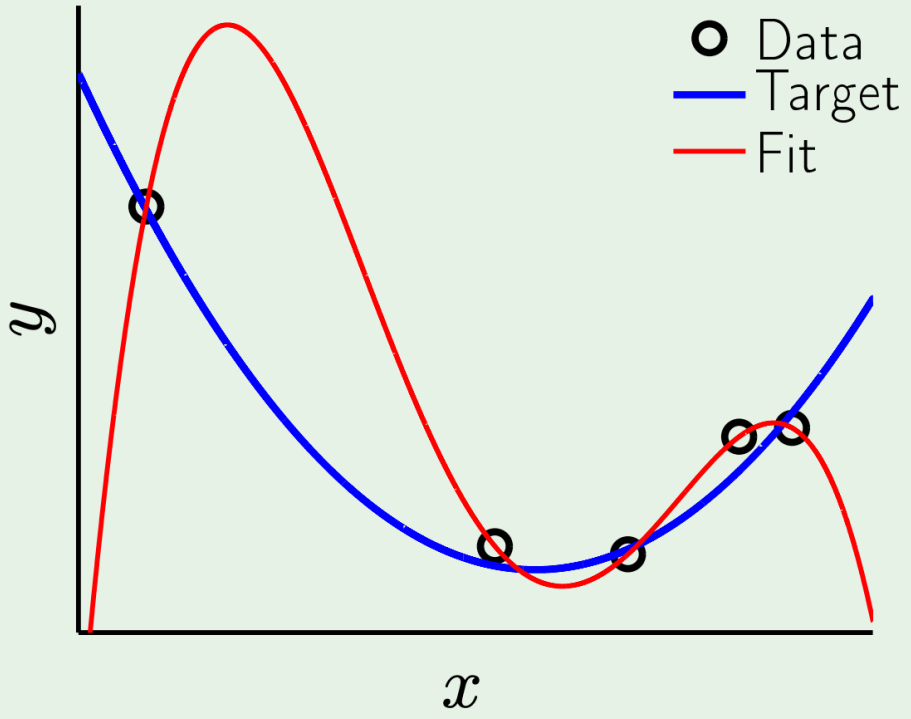
Illustration of over-fitting

Simple target function

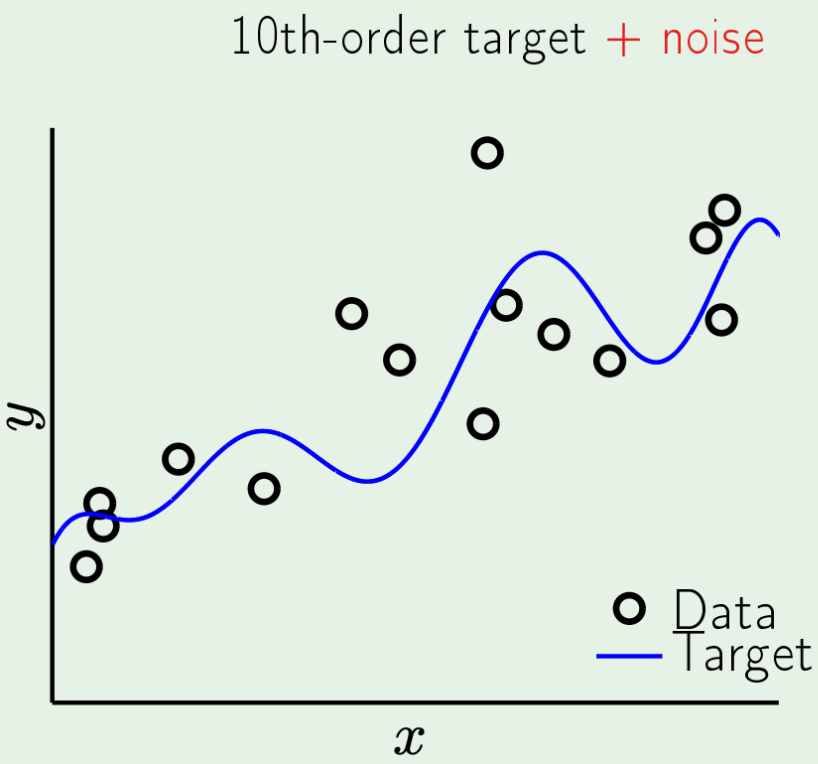
5 data points - noisy

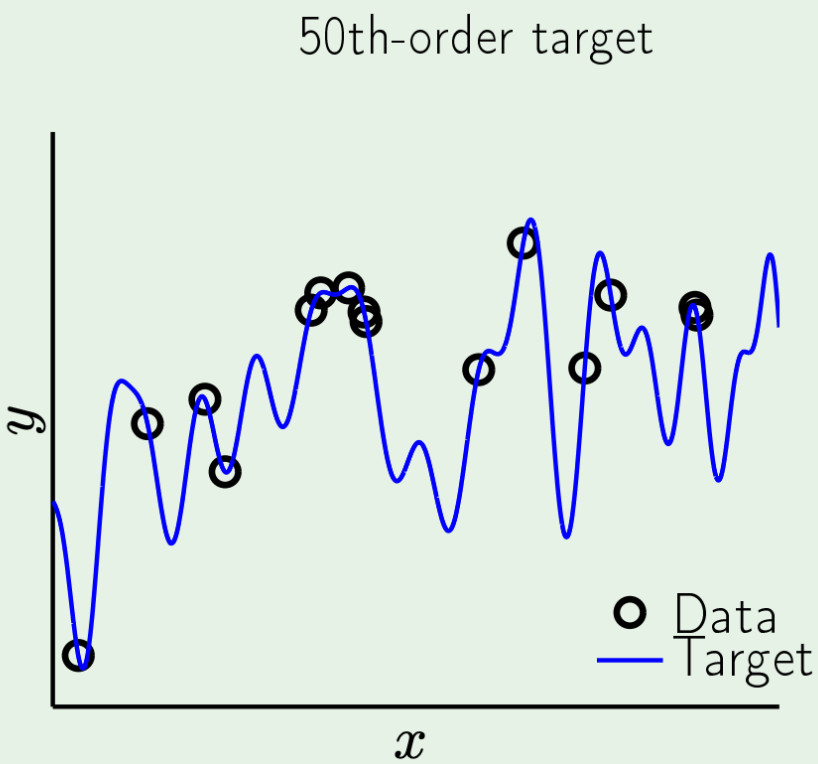
4th-order polynomial fit

$E_{\text{in}} = 0$, E_{out} is huge



Case Study


10th-order target + noise
y
x
Data
Target

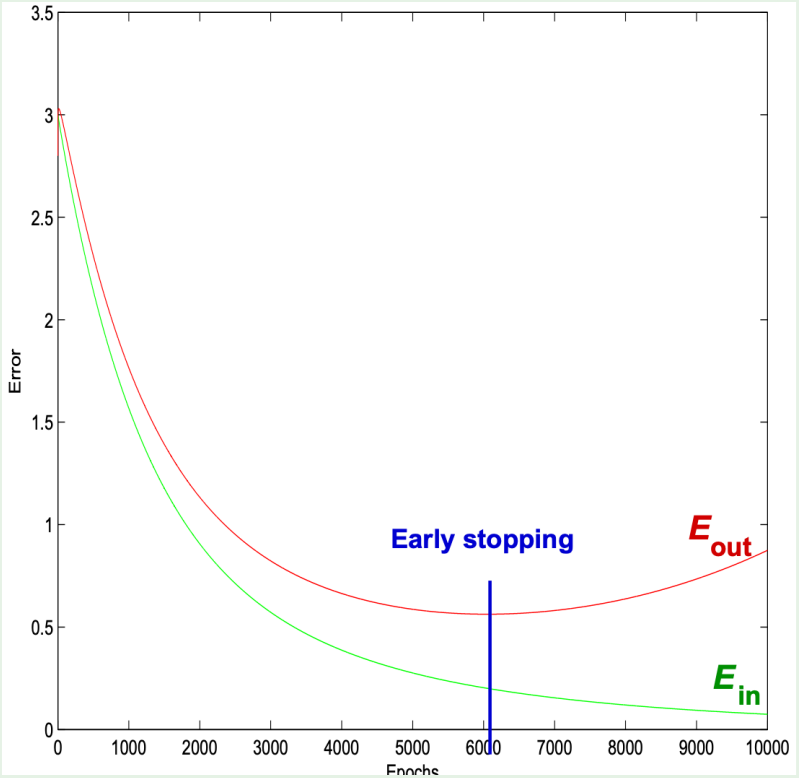

50th-order target
y
x
Data
Target

Over-fitting versus bad generalization

Neural network fitting noisy data

Over-fitting: $E_{\text{in}} \downarrow$ $E_{\text{out}} \uparrow$

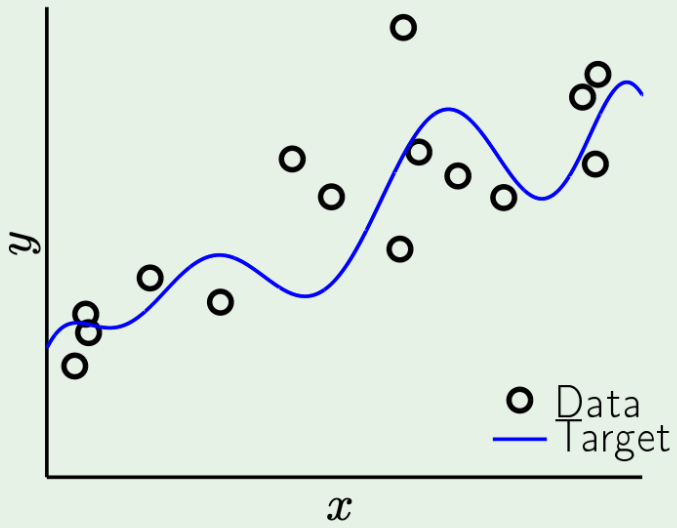
E_{out} : Generalization Error
or Out-of-Sample-Error



Over-fitting: "fitting the data more than is warranted"

A detailed experiment

Impact of **noise level** and **target complexity**



$$y = f(x) + \underbrace{\epsilon(x)}_{\sigma^2} = \underbrace{\sum_{q=0}^{Q_f} \alpha_q x^q}_{\text{normalized}} + \epsilon(x)$$

noise level: σ^2

target complexity: Q_f

data set size: N

Outline

- Empirical Error and Overfitting
- Evaluation Methods
- Performance Measure
- Bias and Variance
- Further Reading

Evaluation Methods

- Here, we only consider the generalization error, but in real-world applications, we often consider more factors such as computational cost, memory cost, and interpretability.

- We assume that the testing samples are independent and identically sampled from the ground-truth sample distribution, and use the test error as an approximation to the generalization error , thus the test set and the training set should be mutually exclusive as much as possible.

Hold out

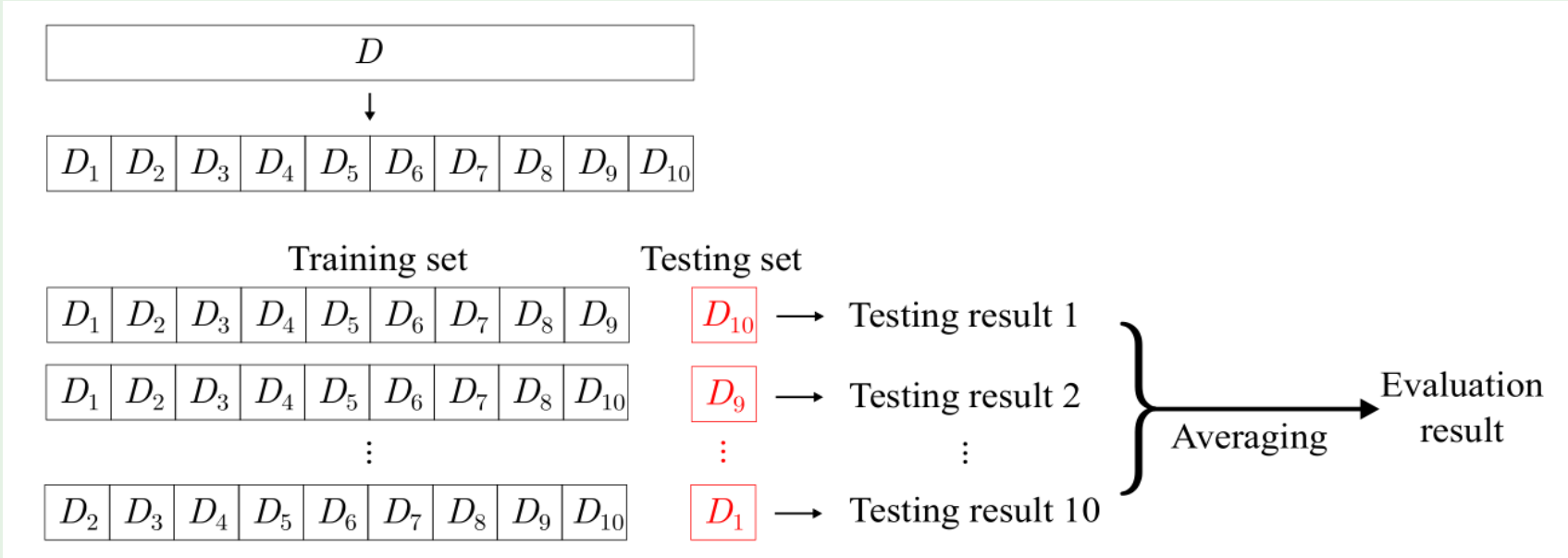
Given the only data set of m samples how can we do both training and testing? The answer is to produce both a training set S and a test set T from the data set D.

- Splits the data set into two disjoint subsets
- The splitting should maintain the original data distribution to avoid introducing additional bias
- We often perform the hold-out testing multiple times, where each trial splits the data randomly, and we use the average error as the final estimation.
- One routine is to use around 2/3 to 4/5 of the examples for training and the rest for testing

Cross-Validation

Cross-validation splits data set D into k disjoint subsets with similar sizes. In each trial of cross-validation, we use the union of $k-1$ subsets as the training set to train a model and then use the remaining subset as the testing set to evaluate the model.

We repeat this process k times and average over k trials to obtain the evaluation result. The most commonly used value of k is 10.



10-fold cross-validation

Leave-One-Out

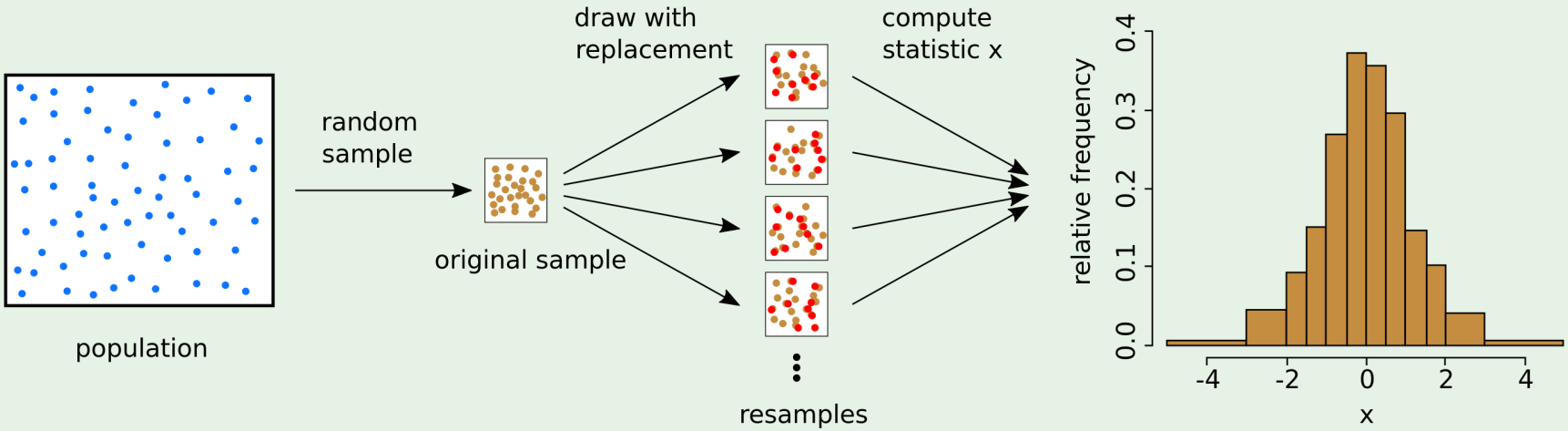
Like hold-out, there are different ways of splitting the data set D into k subsets. To decrease the error introduced by splitting, we often repeat the random splitting p times and average the evaluation results of p times of k-fold cross-validation. For example, a common case is 10-time 10-fold cross-validation.

For a data set D with m samples, a special case of cross-validation is Leave-One-Out (LOO), which lets $k = m$:

- The random splitting does not matter
- The evaluation from LOO is very close to the ideal evaluation
- Computational cost of training m models could be prohibitive for large data sets

Bootstrapping

Given a data set D containing m samples, bootstrapping samples a data set D' by randomly picking one sample from D, copying it to D' , and then placing it back to D so that it still has a chance to be picked next time. Repeating this process m times results in the bootstrap sampling data set D' containing m samples. we can use D' as the training set and $D \backslash D'$ as the testing set.



- Roughly 36.8% of the original samples do not appear in the training data.
- Bootstrapping can create multiple data sets, which can be useful for methods such as ensemble learning
- Bootstrapping is particularly useful when the data set is small

Outline

- Empirical Error and Overfitting
- Evaluation Methods
- Performance Measure
- Bias and Variance
- Further Reading

Performance Measure

- Performance measures can quantify the generalization ability. Different performance measures reflect the varied demands of tasks and produce different evaluation results.

- In classification problems, we are given a data set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$ where y_i is the ground-truth label of the sample $\boldsymbol{x}_i$. To evaluate the performance of a learner f, we compare its prediction $f(x)$ to the ground-truth label y.

- For regression problems, the most commonly used performance measure is the Mean Squared Error (MSE):

$$E(f; D) = \frac{1}{m} \sum_{i=1}^{m} (f(\boldsymbol{x}_i) - y_i)^2$$

Performance Measure

Error rate and **accuracy** are the most commonly used performance measures in classification problems：

- Error rate is the proportion of misclassified samples to all samples
- Accuracy is the proportion of correctly classified samples instead

Error rate

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\left(f\left(\boldsymbol{x}_i\right)\neq y_i\right)$$

Accuracy

$$\begin{aligned}\mathrm{acc}(f;D) &= \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\left(f\left(\boldsymbol{x}_i\right)=y_i\right)\\ &= 1-E(f;D)\ .\end{aligned}$$

Performance Measure

- We often want to know "*What percentage of the retrieved information is of interest to users?*" and "*How much of the information the user interested in is retrieved?*" in applications like information retrieval and web search. For such questions, **precision** and **recall** are better choices.
- In binary classification, there are four combinations of the ground-truth class and the predicted class, namely *true positive*, *false positive*, *true negative*, and *false negative*. The four combinations can be displayed in a confusion matrix.

The confusion matrix of binary classification

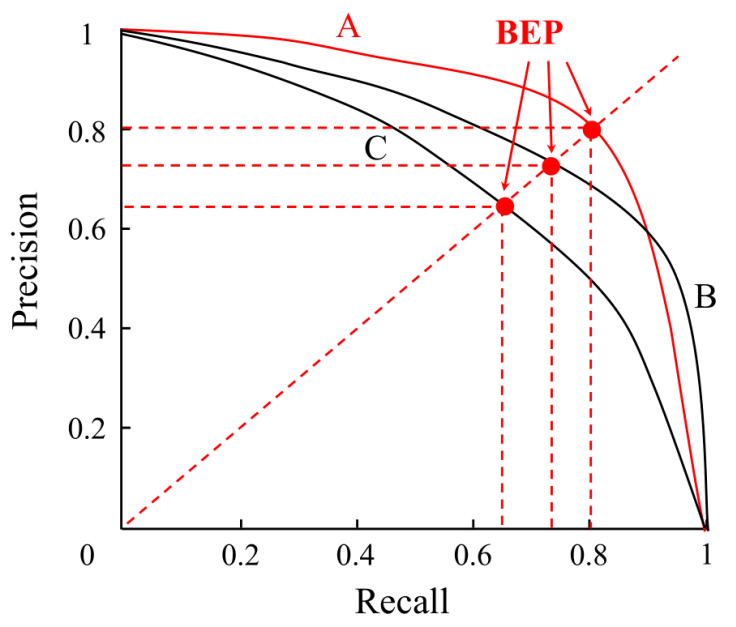
Ground-truth class	Predicted class	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Precision $P = \frac{TP}{TP + FP}$

Recall $R = \frac{TP}{TP + FN}$

Performance Measure

We can use the learner's predictions to sort the samples by how likely they are positive. Starting from the top of the ranking list, we can incrementally label the samples as positive to calculate the precision and recall at each increment. Then, plotting the precisions as y-axis and the recalls as x-axis gives the Precision-Recall Curve (P-R curve).



P-R curve and break-even points

Break-Even Point (BEP) is the value when precision and recall are equal, which can be used to compare performance when the P-R curve intersects.

Performance Measure

BEP could be oversimplified, and a more commonly used alternative is **F1-score**:

Harmonic mean

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{\text{total number of samples} + TP - TN}$$

The general form of F-score is : F_β

$$F_\beta = \frac{(1 + \beta^2) \times P \times R}{(\beta^2 \times P) + R}$$

$\beta = 1$: Standard F1-score

$\beta > 1$: Recall is more important

$\beta < 1$: Precision is more important

The F-score has been widely used in the natural language processing literature, such as in the evaluation of named entity recognition and word segmentation.

Performance Measure

- The ranking quality reflects the learner's "*expected generalization ability*" for different tasks or the generalization ability for "*typical cases*". The *Receiver Operating Characteristics (ROC) curve* follows this idea to measure the generalization ability of learners.

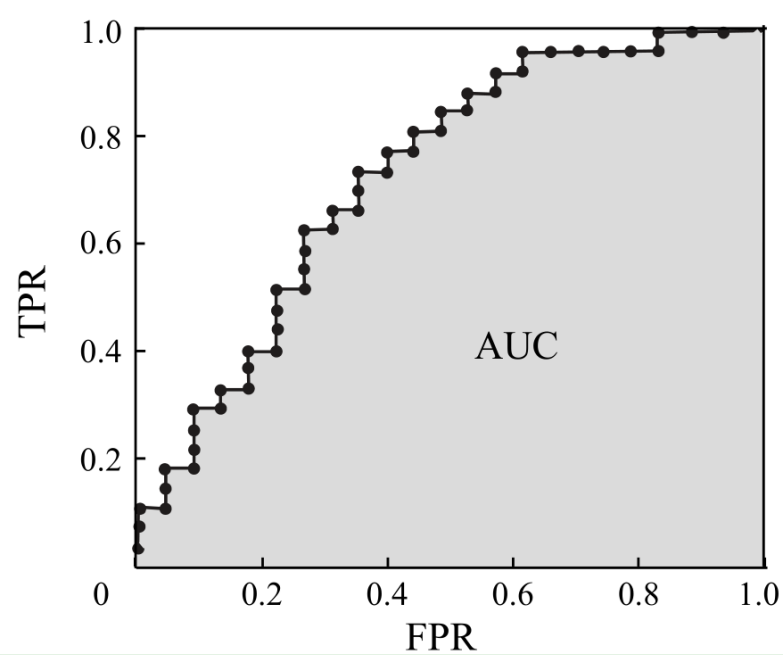
- The plotting process is as follows:

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$

1. given m^+ positive samples and m^- negative samples, we first sort all samples by the learner's predictions, and then set the threshold to maximum, that is, predicting all samples as negative.
2. At this moment, both TPR and FPR are 0, so we mark at coordinate $(0, 0)$.
3. Then, we gradually decrease the threshold to the predicted value of each sample along the sorted list, that is, the samples are classified as positive successively. Let (x, y) denote the previous coordinate, we put a mark at $(x, y + \frac{1}{m^+})$ if the current samples is true positive, and we put a mark at $(x + \frac{1}{m^-}, y)$ if the current samples is false positive.
4. By connecting all adjacent marked points, we have the ROC curve.

Performance Measure

Learner A is better than learner B if A's ROC curve entirely encloses B's ROC curve; However, when there exist intersections, no learner is generally better than the other. One way of comparing intersected ROC curves is to calculate the areas under the ROC curves, that is, Area Under ROC Curve (AUC or AUROC).



ROC curve and AUC with finite samples

Suppose that the ROC curve is obtained by sequentially connecting the points $\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, where $x_1 = 0$ and $x_m = 1$.

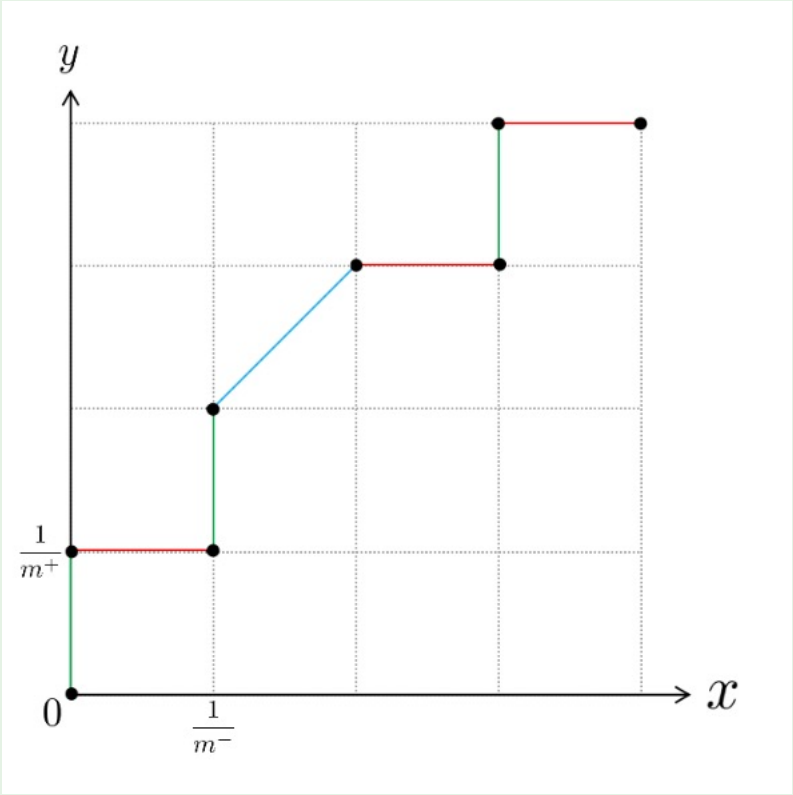
Then, the AUC is estimated as（梯形面积公式）

$$\mathrm{AUC} = \frac{1}{2}\sum_{i=1}^{m-1}(x_{i+1} - x_i)\cdot(y_i + y_{i+1})$$

AUC considers the ranking quality of predictions

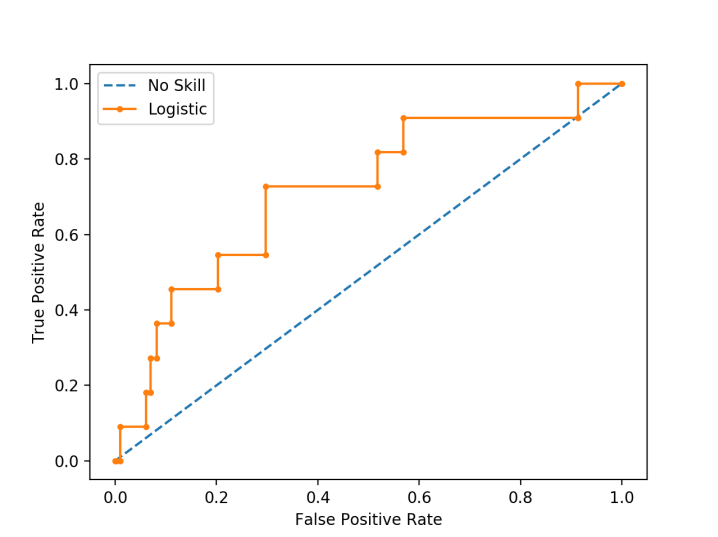
Plot the ROC curve

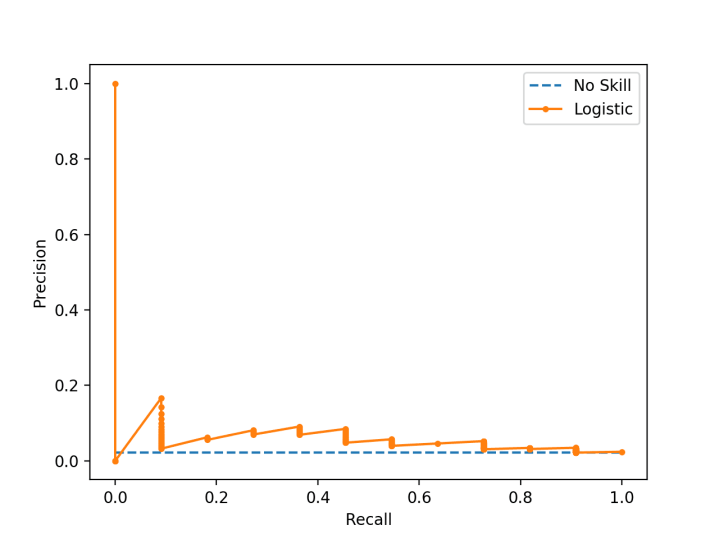
$$(s_1, 0.77, +), (s_2, 0.62, -), (s_3, 0.58, +), (s_4, 0.47, +), (s_5, 0.47, -), (s_6, 0.33, -), (s_7, 0.23, +), (s_8, 0.15, -)$$



When to Use ROC vs. Precision-Recall Curves?

- ROC curves should be used when there are roughly equal numbers of observations for each class.
- Precision-Recall curves should be used when there is a moderate to large class imbalance.





Cost-Sensitive Error Rate

- In some problems, the consequences of making different errors are not the same, thus we need to assign unequal costs to different errors.

- For binary classification problems, we can leverage domain knowledge to design a cost matrix. where $cost_{ij}$ represents the cost of misclassifying a sample of class i as class j. The larger the difference between the costs is, the larger the difference between $cost_{01}$ and $cost_{10}$ will be.

- With unequal costs, however, we no longer minimize the counts but the total cost, the cost-sensitive error rate is defined as：

$$E(f;D;cost)=\frac{1}{m}\left(\sum_{\boldsymbol{x}_i\in D^+}\mathbb{I}\left(f\left(\boldsymbol{x}_i\right)\neq y_i\right)\times cost_{01}+\sum_{\boldsymbol{x}_i\in D^-}\mathbb{I}\left(f\left(\boldsymbol{x}_i\right)\neq y_i\right)\times cost_{10}\right)$$

Sources: [12][13][14][15][16][17][18][19][20] view · talk · edit

		Predicted condition			
	Total population $= P + N$	Predicted Positive (PP)	Predicted Negative (PN)	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	Positive (P)[a]	True positive (TP), hit[b]	False negative (FN), type II error, miss, underestimation[c]	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate $= \frac{FN}{P} = 1 - TPR$
	Negative (N)[d]	False positive (FP), type I error, false alarm, overestimation[e]	True negative (TN), correct rejection[f]	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2\,PPV \times TPR}{PPV + TPR} = \frac{2\,TP}{2\,TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

Outline

- Empirical Error and Overfitting
- Evaluation Methods
- Performance Measure
- Bias and Variance
- Further Reading

Bias and Variance

- Bias measures the difference between the learning algorithm's expected prediction and the ground-truth label; that is, expressing the fitting ability of the learning algorithm;
- Variance measures the change of learning performance caused by changes to the equal-sized training set, that is, expressing the impact of data disturbance on the learning outcome;

The bias-variance decomposition tells us that the generalization performance is jointly determined by the learning algorithm's ability, data sufficiency, and the inherent difficulty of the learning problem.

In order to achieve excellent generalization performance, a small bias is needed by adequately fitting the data, and the variance should also be kept small by minimizing the impact of data disturbance.

Bias and Variance

Bias-Variance analysis decomposes E_{out} into

1. How well H can approximate f
2. How well we can zoom in on a good $h \in H$

Applies to **real-valued targets** and uses **squared error**

$$E_{\text{out}}(g^{(\mathcal{D})}) = \mathbb{E}_{\mathbf{x}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]$$

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[E_{\text{out}}(g^{(\mathcal{D})})\right] &= \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\mathbf{x}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]\right] \\ &= \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]\right]\end{aligned}$$

Bias and Variance – Squared Error

Now let us focus on:

$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]$$

We define the average hypothesis:

$$\bar{g}(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}\left[g^{(\mathcal{D})}(\mathbf{x})\right]$$

Imagine many data sets $\mathtt{D}_1, \mathtt{D}_2, \cdots, \mathtt{D}_K$

$$\bar{g}(\mathrm{x}) \approx \frac{1}{K}\sum_{k=1}^{K} g^{(\mathcal{D}_k)}(\mathrm{x})$$

Bias and Variance - Using $\bar{g}(\mathbf{x})$

$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x})-f(\mathbf{x})\right)^2\right]=\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x})-\bar{g}(\mathbf{x})+\bar{g}(\mathbf{x})-f(\mathbf{x})\right)^2\right]$$

$$=\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x})-\bar{g}(\mathbf{x})\right)^2+\left(\bar{g}(\mathbf{x})-f(\mathbf{x})\right)^2\right.$$

$$\left.+\ 2\ \left(g^{(\mathcal{D})}(\mathbf{x})-\bar{g}(\mathbf{x})\right)\ \left(\bar{g}(\mathbf{x})-f(\mathbf{x})\right)\right]$$

$$=\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x})-\bar{g}(\mathbf{x})\right)^2\right]+\left(\bar{g}(\mathbf{x})-f(\mathbf{x})\right)^2$$

Bias and Variance

$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x})-f(\mathbf{x})\right)^2\right]=\underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x})-\bar{g}(\mathbf{x})\right)^2\right]}_{\text{var}(\mathbf{x})}+\underbrace{\left(\bar{g}(\mathbf{x})-f(\mathbf{x})\right)^2}_{\text{bias}(\mathbf{x})}$$

Therefore, $\mathbb{E}_{\mathcal{D}}\left[E_{\text{out}}(g^{(\mathcal{D})})\right]=\mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x})-f(\mathbf{x})\right)^2\right]\right]$

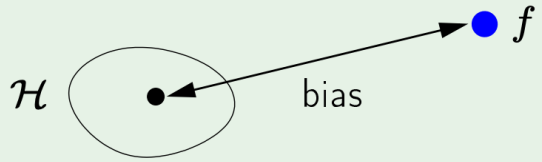
$$=\mathbb{E}_{\mathbf{x}}[\text{bias}(\mathbf{x})+\text{var}(\mathbf{x})]$$

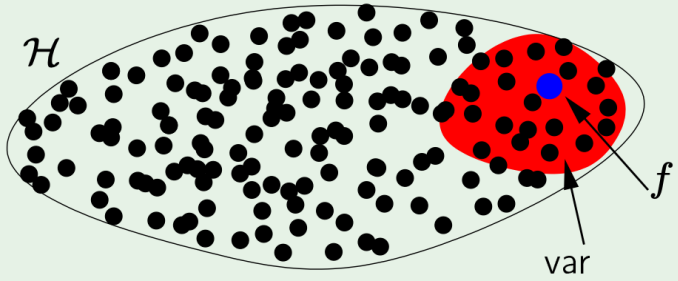
$$=\text{bias}+\text{var}$$

Bias and Variance Tradeoff

$$\text{bias} = \mathbb{E}_{\mathbf{x}}\left[\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]$$

$$\text{var} = \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]\right]$$





$\downarrow$ $\mathcal{H}\uparrow$ $\uparrow$

Example – sine target

f

$f : [-1, 1] \to \mathbb{R} \qquad f(x) = \sin(\pi x)$

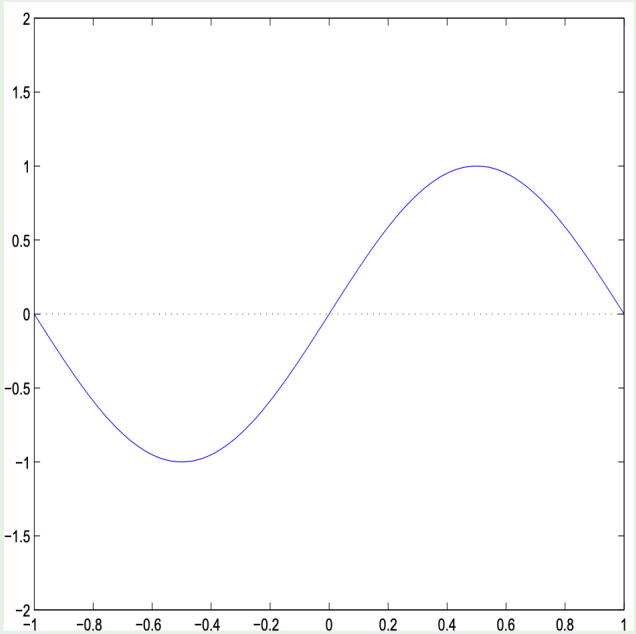
Only two training examples! $N = 2$

Two models used for learning:

$\mathcal{H}_0$: $h(x) = b$

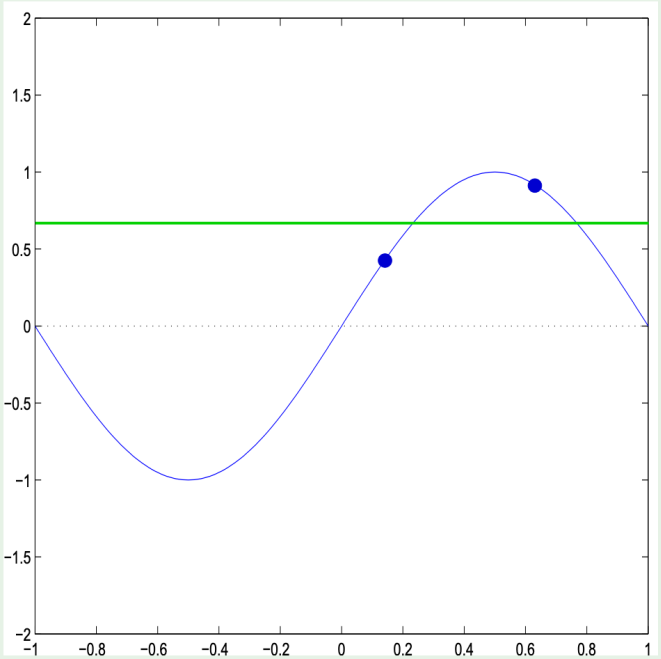
$\mathcal{H}_1$: $h(x) = ax + b$

Which is better, $\mathcal{H}_0$ or $\mathcal{H}_1$?

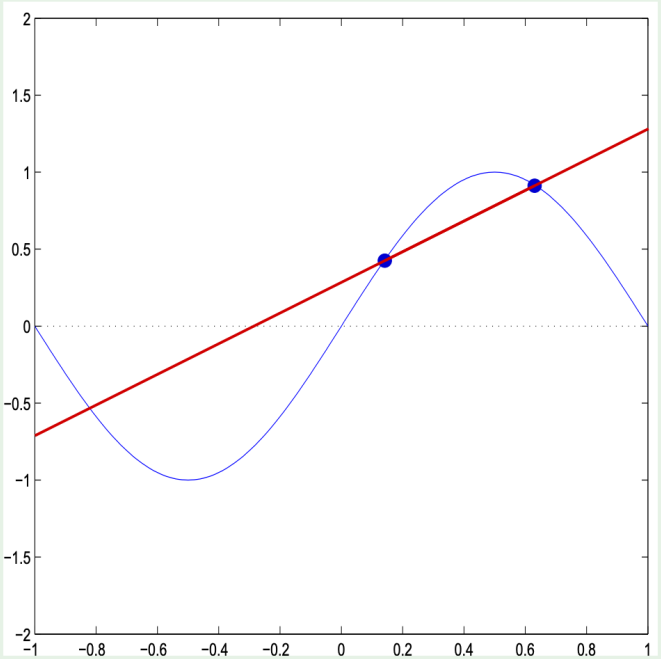


Learning H_0 versus H_1

$\mathcal{H}_0$

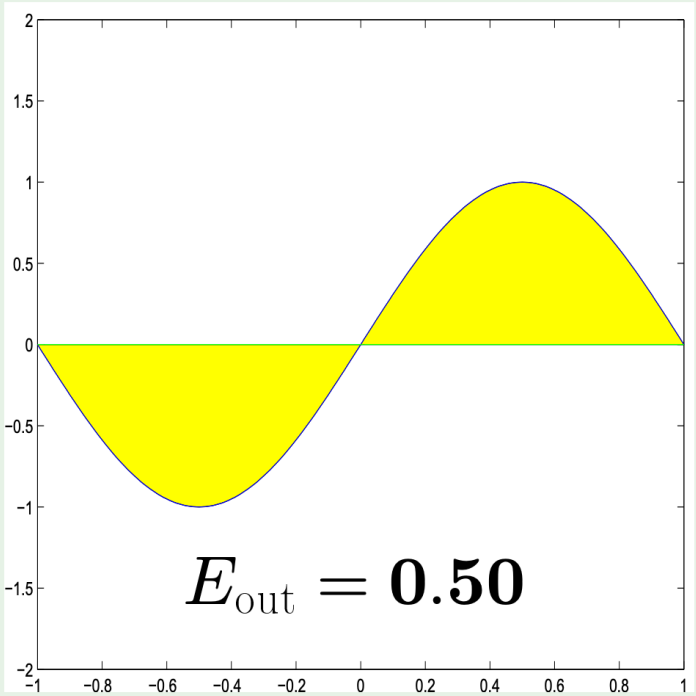


$\mathcal{H}_1$

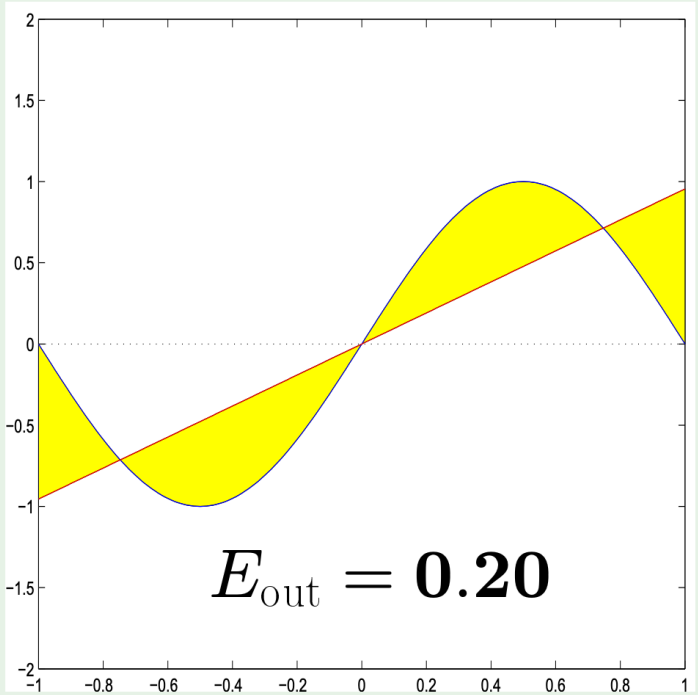


Approximation – H_0 versus H_1

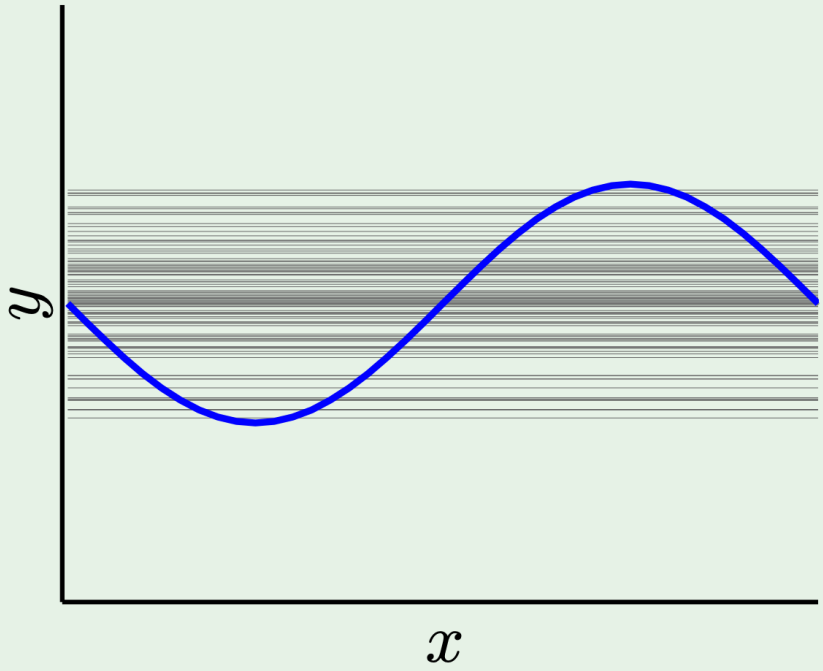
$\mathcal{H}_0$

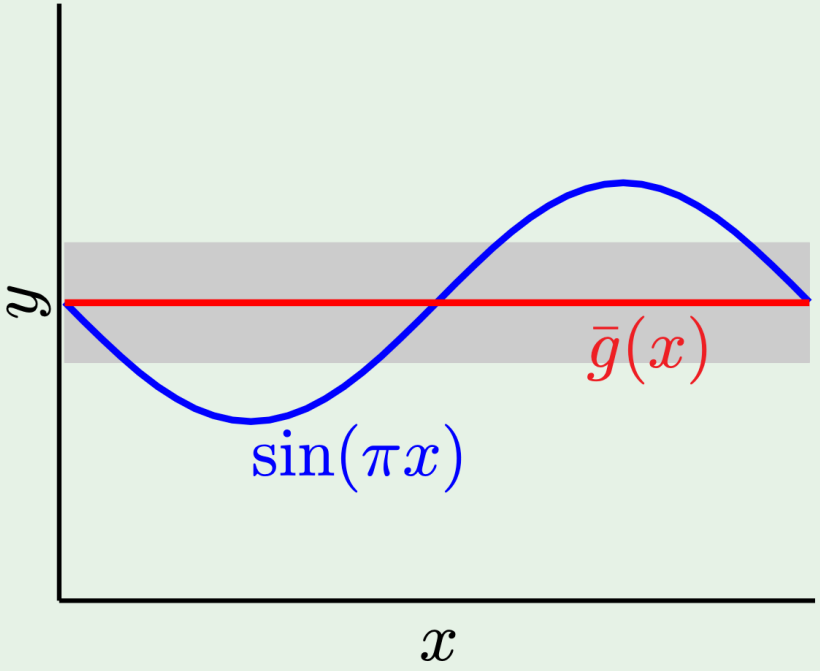


$\mathcal{H}_1$

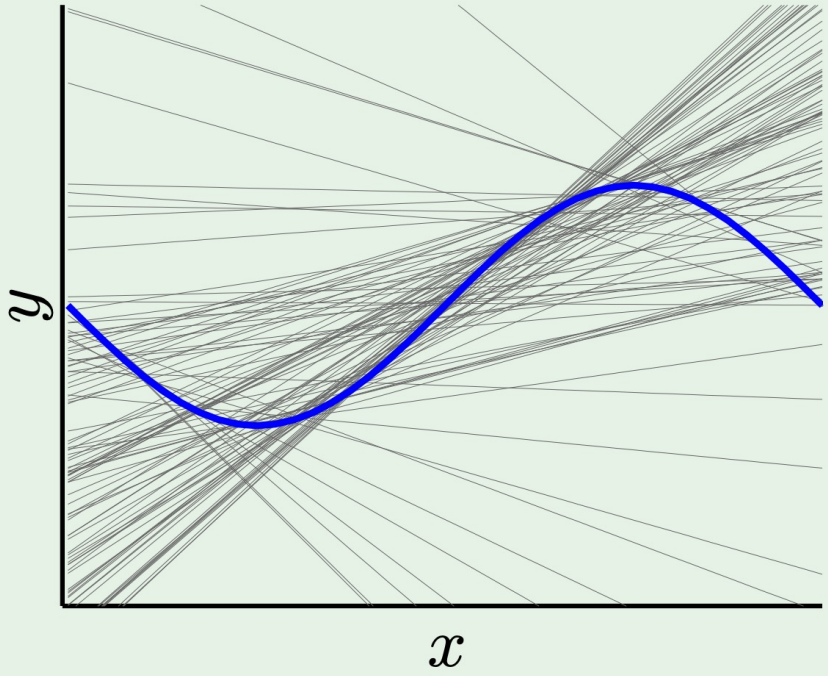


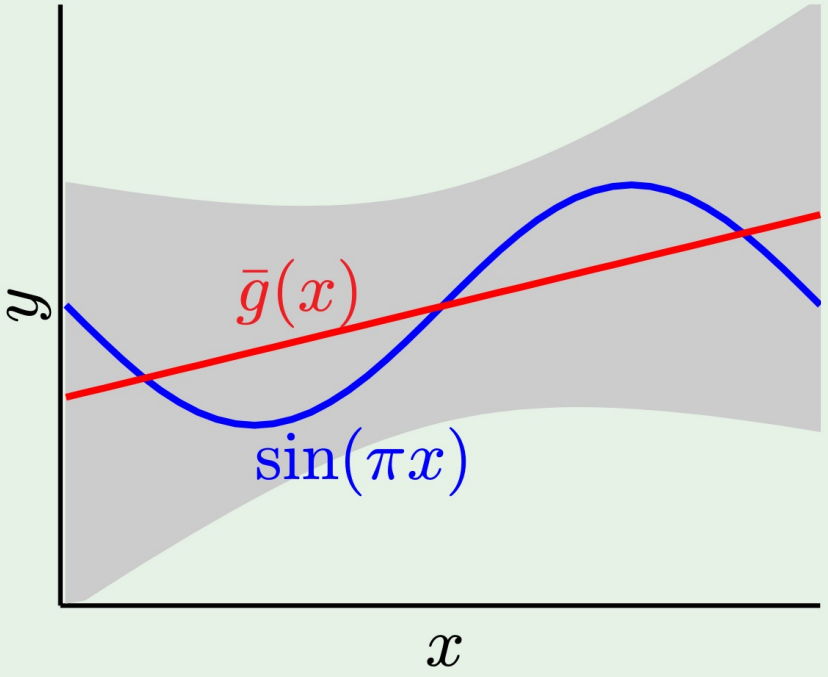
Bias and Variance – H_0



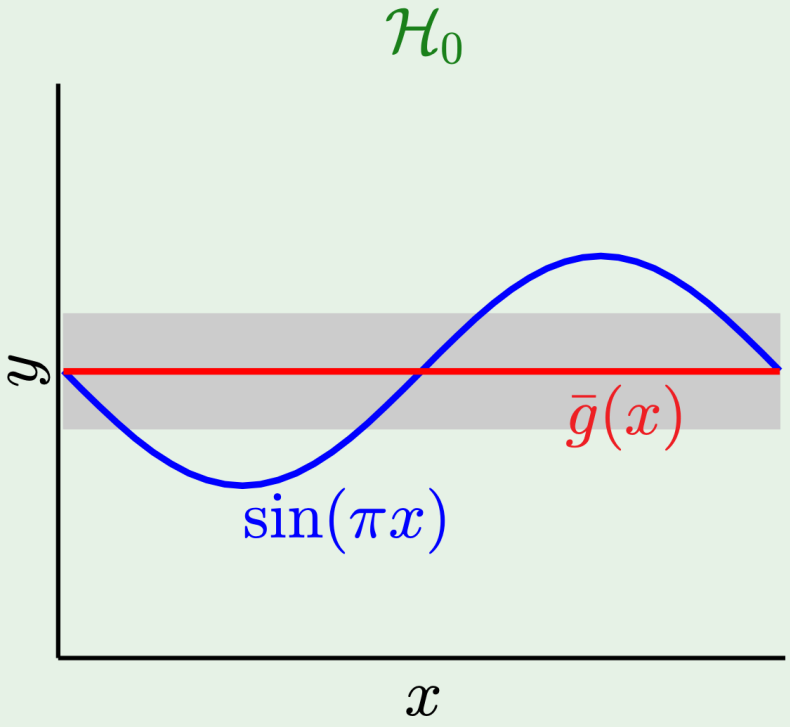


Bias and Variance – H_1

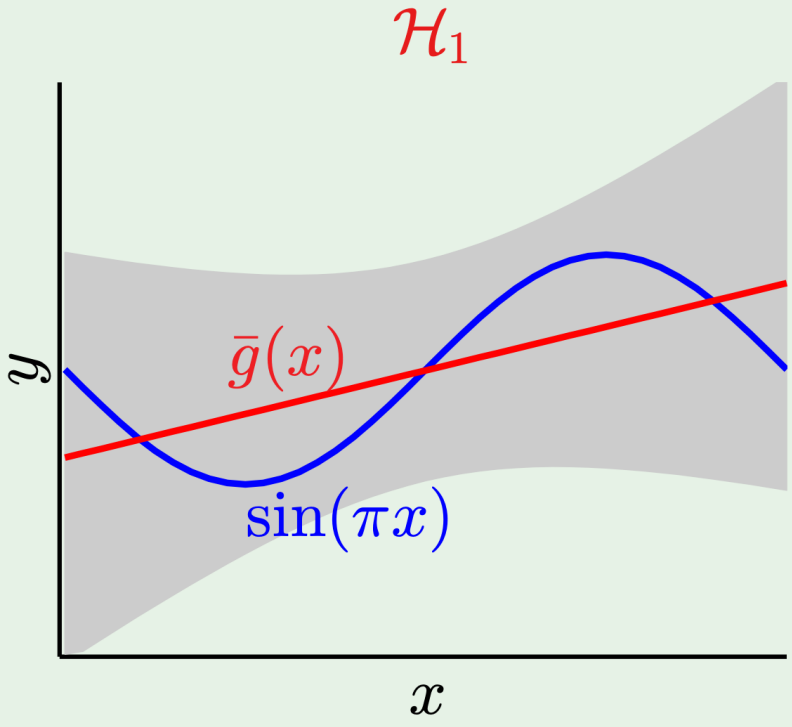




and the winner is . . .



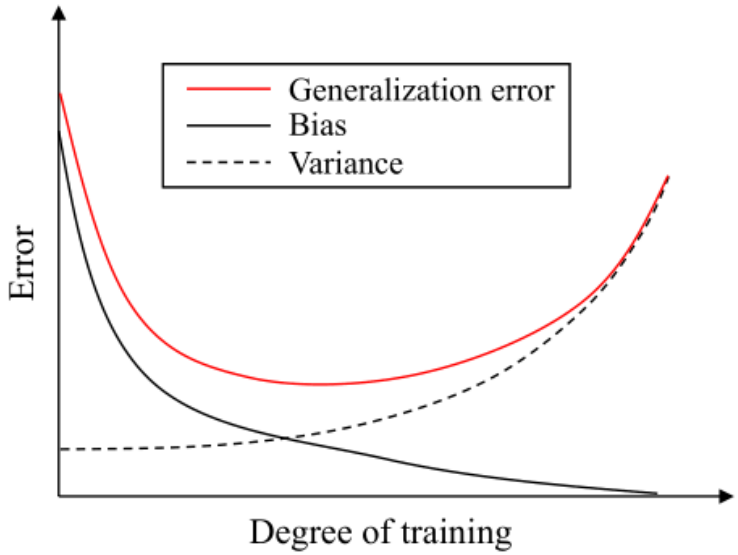
bias $= \mathbf{0.50}$ var $= \mathbf{0.25}$



bias $= \mathbf{0.21}$ var $= \mathbf{1.69}$

Lesson learned

Match the "model complexity" to the data resources, not to the target complexity.



Relationships between generalization error, bias, and variance.

Outline

- Empirical Error and Overfitting
- Evaluation Methods
- Performance Measure
- Bias and Variance
- Further Reading

Further Reading

- Bootstrap sampling has crucial applications in machine learning, and a detailed discussion can be found in *[Efron and Tibshirani, 1993]*.

- ROC curve was introduced to machine learning in the late 1980s*[Spackman, 1989]*, and AUC started to be widely used in the field of machine learning since the middle 1990s *[Bradley,1997]*. *[Hand and Till,2001]* extended the ROC curve from binary classification problems to multiclass classification problems. *[Fawcett,2006]* surveyed the use of the ROC curve.

- *[Drummond and Holte,2006]* invented the cost curve. Cost-sensitive learning *[Elkan,2001;Zhou and Liu,2006]* is a research topic for learning under unequal cost settings.

Further Reading

- *[Dietterich,1998]* pointed out the risk of using the regular k-fold cross-validation method, and proposed the 5 × 2 cross-validation method. *[Demsar, 2006]* discussed the hypothesis testing methods for comparing multiple algorithms.

- *[Geman et al.,1992]* proposed the bias-variance-covariance decomposition for regression problems, which was later shortened as bias-variance decomposition. For classification problems, however, deriving the bias-variance decomposition is difficult since the 0/1 loss function is discontinuous. There exist many empirical methods for estimating bias and variance *[Kong and Dietterich,1995;Kohavi and Wolpert, 1996; Breiman, 1996; Friedman,1997; Domingos,2000]*.

Summary

- How to do model selection?
  - Cross-validation
  - Leave-one-out

- How can we evaluate models?
  - Accuracy
  - Precision, Recall, F1
  - AUC