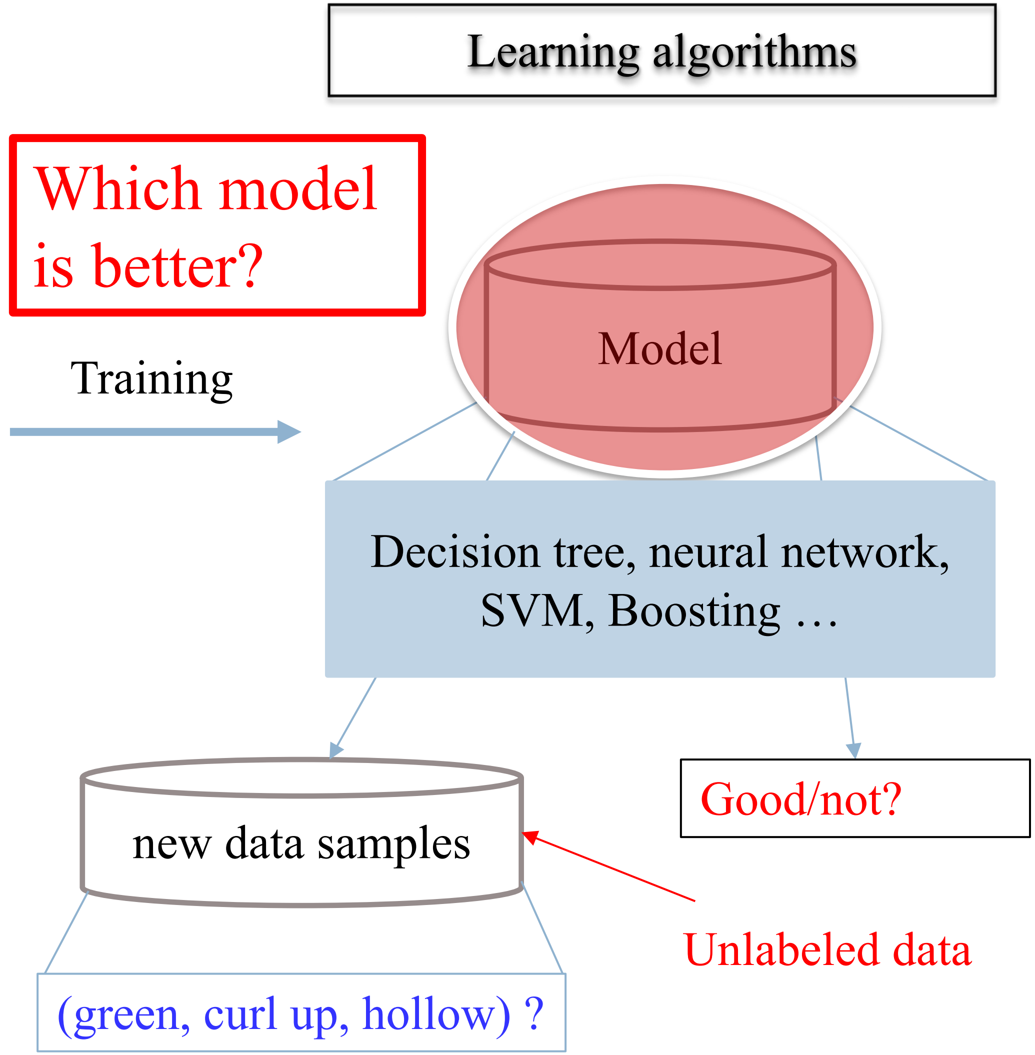
Model Selection and Evaluation

Alymzhan Toleu
alymzhan.toleu@gmail.com

color	Root/vines	sounds	good/not
green	curl up	deep sound	good
Jet-black	curl up	deep sound	good
green	stiff	crisp sound	not
Jet-black	slightly curled up	dull sound	not

labels



Need a model with good generalization!

Generalization error vs. Empirical error

Generalization error (a.k.a out-of-sample error):

- the model's prediction errors on new unseen data;

Empirical error

- the model's prediction errors on training data;

the lower generalization error, the better.

the lower empirical error, the better ?

NO ! It causes overfitting problem.

Overfitting vs. Underfitting

Tranining samples

new samples

Overfitting model predicts it is not a leaf.

(it considers all leaf must have leaf serration.)

underfitting model predicts it is a leaf.

(it considers all leaf must be green.)

Model Selection

Two main questions:

- How to evaluate a model? $\Rightarrow$ Model Evaluation

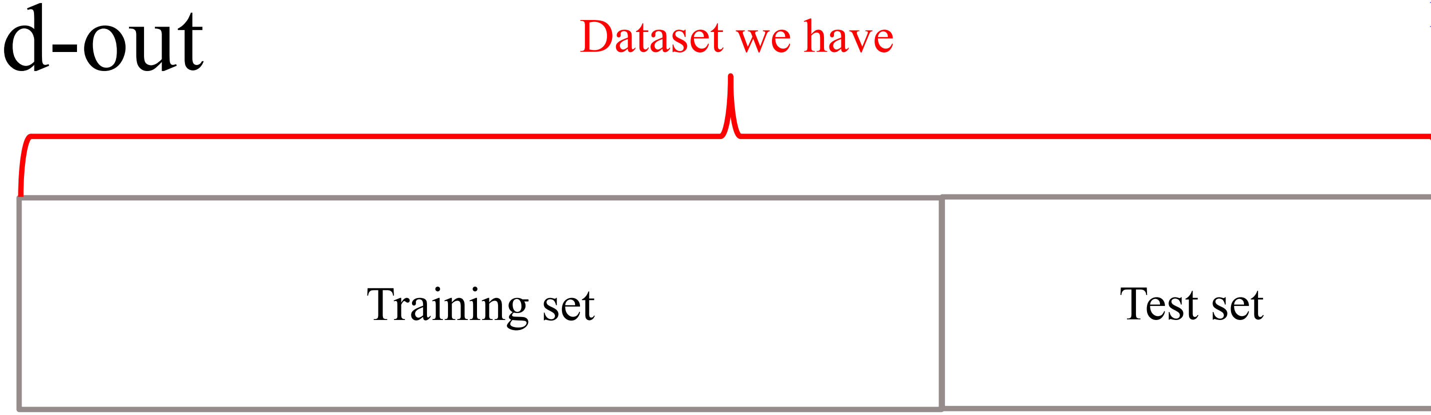
- How to calculate model's performance? $\Rightarrow$ Evaluation Metrics

Model Evaluation

- How to obtain a test set?
  - select a portion of data set that do not overlap with training set.

- Hold-out
- Cross validation
- Bootstrap sampling

Hold-out



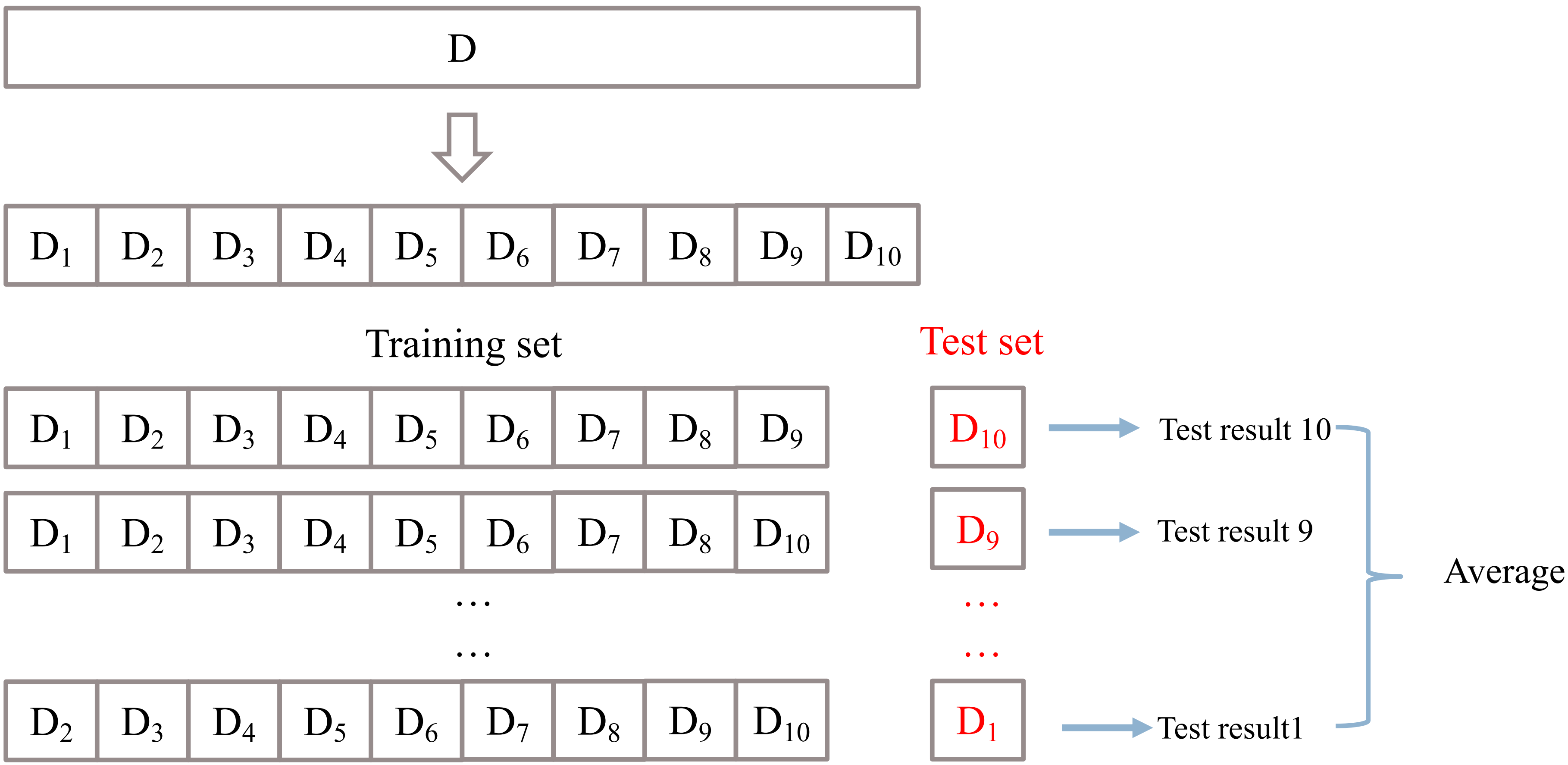
- Test set: the size cannot be large; 20%-30% of the dataset.
- Keep the original distribution of the dataset (take into account the balance of sample's category).
  - use stratified sampling method: e.g. 1000 samples dataset contains 500 negative and 500 positive samples. Then, the test set size should be 300 samples and it must contain equal size of negative and positive samples.
- Repeat multiple times (for example: 100 times random division).
  - report average results of test set.

If training set contains most of the samples, the leared model will close to the original dataset, but the test set becomes small, which does not tell us the real evaluation.

If let training set contains less sample, then the different between the original dataset and training set will increases. It will decrease the fidelity of evaluation.

k - Cross Validation

For given dataset D, it is divided into k-equal subsets. Take k-1 subsets as training set and take one as test set.



10-flod cross validation

k - Cross Validation

- In practice, k can be set to: k=5, k=10 or k=20;
- n times k-fold CV;
  - repeat k-fold CV n times, in order to reduce the variance introduced by sample division.
- Leave-one-out (an extreme case): $k = |D|$
  - training is itself computationally expensive.
  - need to train |D| times.

Pros:

- K >10 helps to minimize the variability in the estimated performance.
- Most of cases, Leave-one-out: the estimated evaluation equals to the expected evaluation.

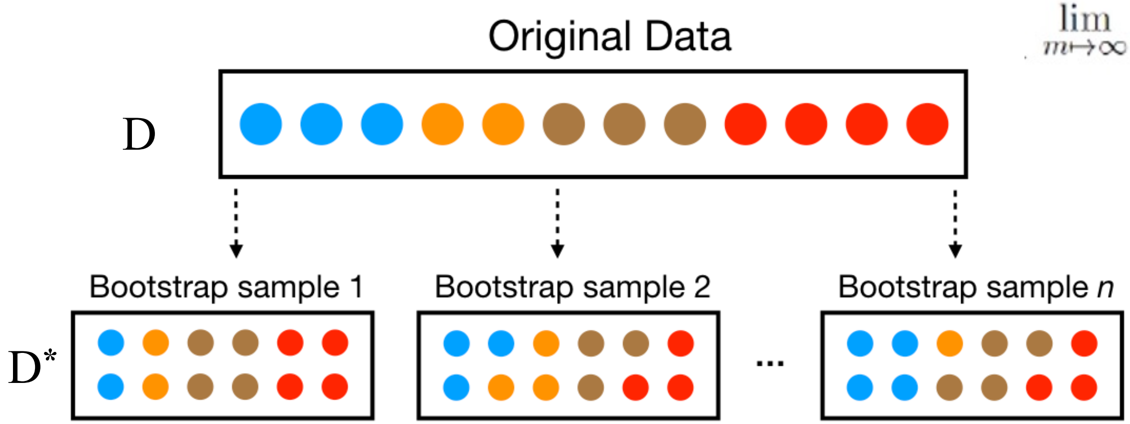
Cons:

- The size of training set and the original data is different, it may introduce variability.
- Leave-one-out requires is computationally expensive.

Consider the parameter tunning, Molinaro et. al. (2005) found that k=10 CV performed similarly to leave-one-out.

Bootstrap sampling [Efron and Tibshirani, 1993]

- Bootstrap sampling is a resampling method that uses random sampling with replacement.
  - after a data point is selected for inclusion in the subset, it's still available for further selection.



$$\lim_{m \mapsto \infty} \left(1 - \frac{1}{m}\right)^m \mapsto \frac{1}{e} \approx 0.368$$

m – the size of D

Pros:

- training set has the same size as the original sample set;
- work well for small sized original data;

Cons:

- Data distribution is changed;

36% of samples of dataset D will not appeared in resampled subsets D*.

Training set: D*

Test set: D \ D*

Out-of-bag estimate

Parameter Tuning and Model Selection

- Algorithm's parameter:
  - a.k.a: hyper-parameters.
  - tuned manually.
- Model's parameter:
  - trained by the learning algorithms.

For example:

- an algorithm has 3 hyper-parameters, each of them can take 5 different values.
- $5^3 = 125$ models should be trained then select the best one.

Parameter tuning process: first generate several models with different setting of hyper-parameters, and then based on some evaluation method to select;

Training set: train our learning algorithms;
Validation set: tune hyperparameters, compare models;
Test set: having chosen a final model, these data are used to estimate the model's performance, which we refer to as the *generalization error*.

After the algorithm parameters are selected, the final model should be retrained with "training set + validation set".

Model Selection

Two main questions:

- How to evaluate a model? ⇒ Model Evaluation
- How to calculate model’s performance? ⇒ Evaluation Metrics

Evaluation Metrics

- Performance measure
  - it is an approach of correctly evaluating model's performance that reflecting the model's generalization.

Using different evaluation metrics could result in different evaluation results. It means that assessing a model's performance depends on the algorithm, data, and task being used.

- Metrics for regression problem is mean squared error (MSE):

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

Error rate and Accuracy

Two typical metrics for classification tasks

Error rate: the proportion of predictions that are incorrect.

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

Accuracy: the proportion of predictions that are correct.

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

m - is the total number of samples

Precision, Recall

For example, we have a collection of documents:

doc1, doc2, doc3, doc4, doc5
doc6, doc7, doc8, doc9, doc10

blue indicates relevant docs.

Search “cat’

Search system

Results:

doc1 -> relevant
doc2 -> nonrelevant
doc3 -> nonrelevant
doc4 -> relevant
doc5 -> relevant
doc6 -> nonrelevant
doc7 -> relevant
doc8 -> relevant
doc9 -> relevant
doc10 -> nonrelrevant

Confusion matrix

actual	Search results	
	relevant	nonrelevant
relevant	5 (True Positive)	2 (False Negative)
nonrelevant	1 (False Positive)	2 (True Negative)

Precision, Recall

- Confusion matrix TP+FN+FP+TN = N

actual	Search results	
	relevant	nonrelevant
relevant	5 (True Positive)	2 (False Negative)
nonrelevant	1 (False Positive)	2 (True Negative)

$$\text{Accuracy} = \frac{TP+TN}{Total} = \frac{5+2}{10} = 0.7$$

$$Precison = \frac{TP}{TP+FP} = \frac{5}{5+1} = 0.83$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{5}{5+2} = 0.714$$

- Precision

How accurately does the system retrieves documents?

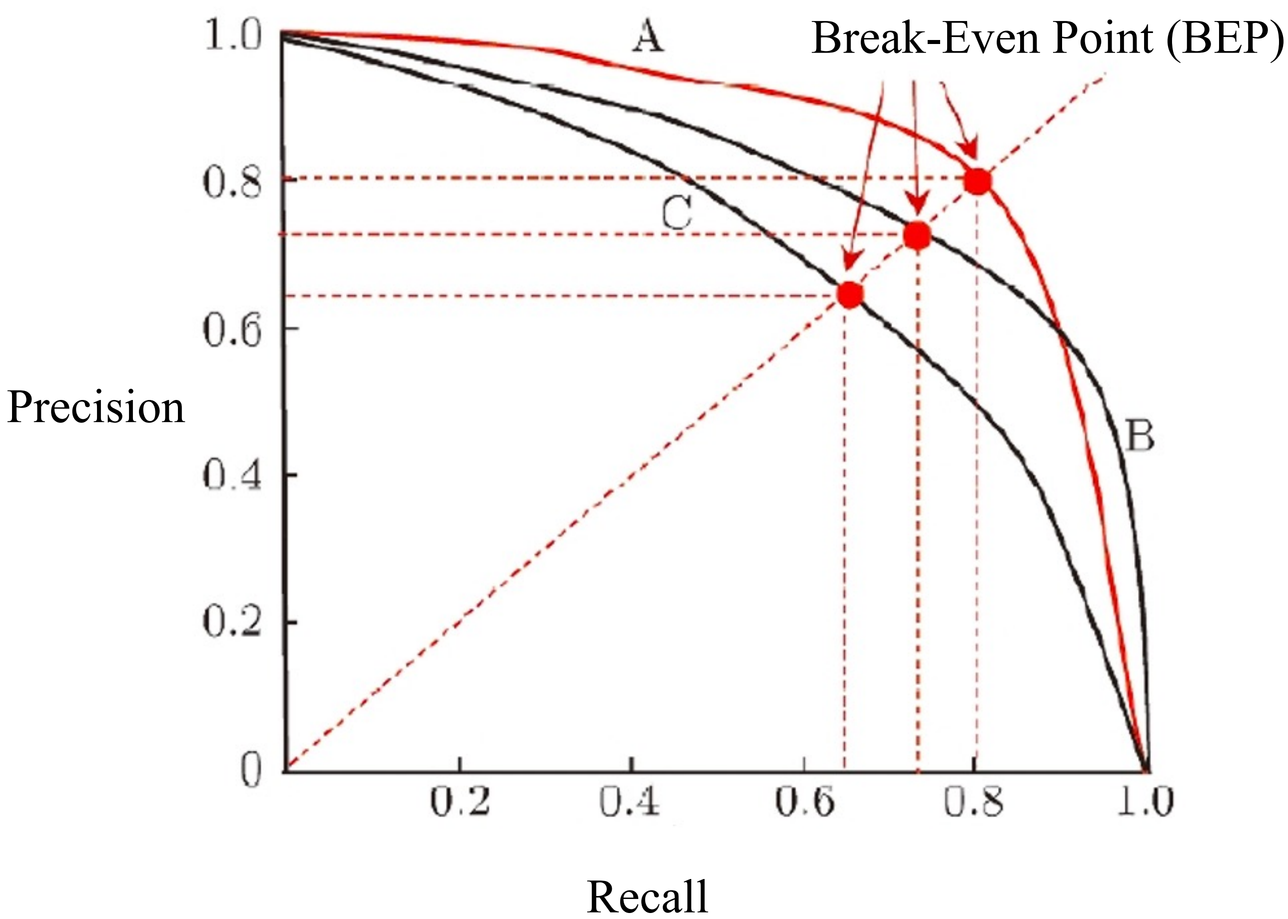
$$P = \frac{TP}{TP+FP}$$

- Recall (sensitivity)

How accurately does the system retrieves relevant documents?

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

PR graph and BEP



PR :

$L_A > L_C$

$L_B > L_C$

$L_A > L_B$?

BEP :

$L_A > L_C$

$L_A > L_B$

$L_B > L_C$

F-score

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

$\beta > 1$: Recall takes dominance;
$\beta < 1$: Precision takes dominance;

When β=1, It becomes standard:

$$\text{F1} = \frac{2 \times P \times R}{P + R}$$

Macros.* and micros.*

If multiple confusion matrices can be obtained:

- such as the results of multiple training/testing;
- multi-category classification - pairwise confusion matrix;

$$\text{macro-}P = \frac{1}{n}\sum_{i=1}^{n} P_i ,$$

$$\text{macro-}R = \frac{1}{n}\sum_{i=1}^{n} R_i ,$$

$$\text{macro-}F1 = \frac{2 \times \text{macro-}P \times \text{macro-}R}{\text{macro-}P + \text{macro-}R} .$$

$$\text{micro-}P = \frac{\overline{TP}}{\overline{TP} + \overline{FP}} ,$$

$$\text{micro-}R = \frac{\overline{TP}}{\overline{TP} + \overline{FN}} ,$$

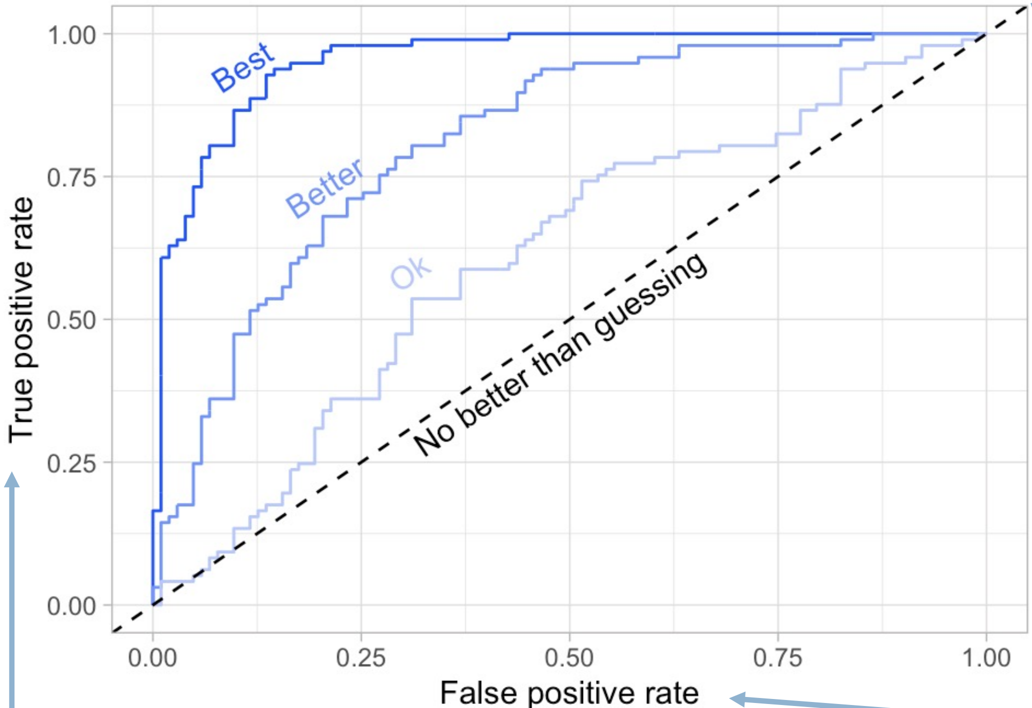
$$\text{micro-}F1 = \frac{2 \times \text{micro-}P \times \text{micro-}R}{\text{micro-}P + \text{micro-}R} .$$

ROC curve

ROC (Receiver Operating Characteristic) Curve

AUC: **A**rea **U**nder the ROC **C**urve

represents a random guess



To compare different models, need to compute the area of AUC.
The lager represents the better.

$$tpr = \frac{TP}{TP+FN} = \frac{TP}{m_+}$$

$$fpr = \frac{FP}{FP+TN} = \frac{FP}{m_-}$$

Unequal cost

- Different algorithms' error tend to cause different losses. e.g.:
  - a system diagnoses a healthy person has a diseases;
  - a door security system allow a stranger enter your house;
- Therefore, need to introduce unequal cost.

Cost matrix $Cost_{ij}$ - the cost of predicting the i-th as j-th.

Actual classes	Predicted classes	
	0-th class	1-th class
0-th class	0	$cost_{01}$
1-th class	$cost_{10}$	0

Cost sensitive (Total cost):

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

Previous metrics assume that they have the equal cost!

Bias-Variance Decomposition

- Prediction errors can be decomposed into two important subcomponents:
  - error due to bias
  - error due to variance
- *Bias* is the difference between the expected (or average) prediction of our model and the correct value which we are trying to predict.
- *Variance* is defined as the variability of a model prediction for a given data point.

Bias-Variance Decomposition

- For regression tasks, the generalization error can be decomposed into:

$$E(f;D) = \underbrace{bias^2(\boldsymbol{x})} + \underbrace{var(\boldsymbol{x})} + \underbrace{\varepsilon^2}$$

difference between the expected output and the actual output.

$$bias^2(\boldsymbol{x}) = \left(\bar{f}(\boldsymbol{x}) - y\right)^2$$

Same sized different traning sets caused this variance.
It express the error caused by data perturbation.

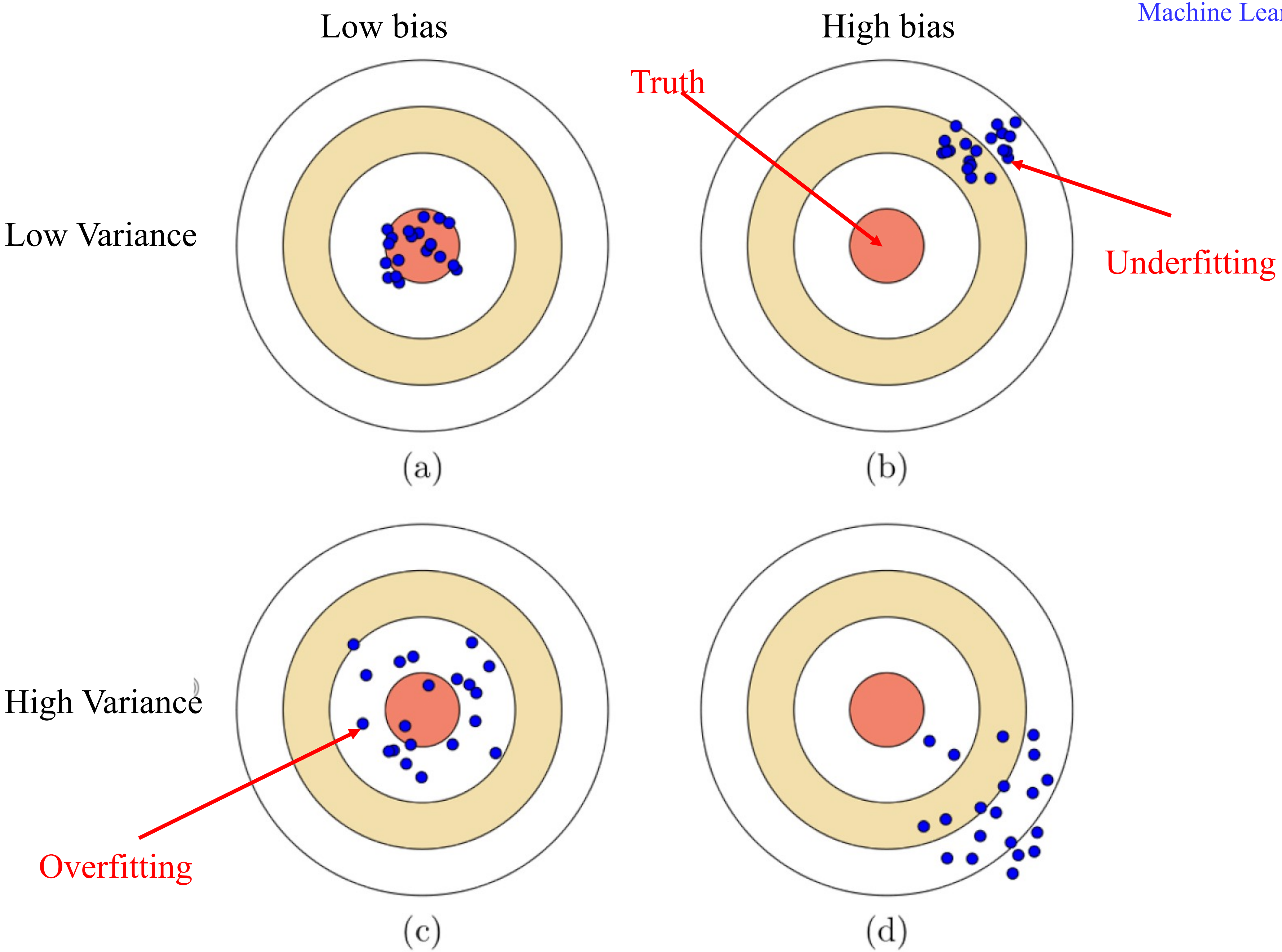
$$var(\boldsymbol{x}) = \mathbb{E}_D\left[\left(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x})\right)^2\right]$$

The labels of the training set are different from the real labels.

Lower bound of expected generalization error.
It express the difficulty level of the task

$$\varepsilon^2 = \mathbb{E}_D\left[(y_D - y)^2\right]$$

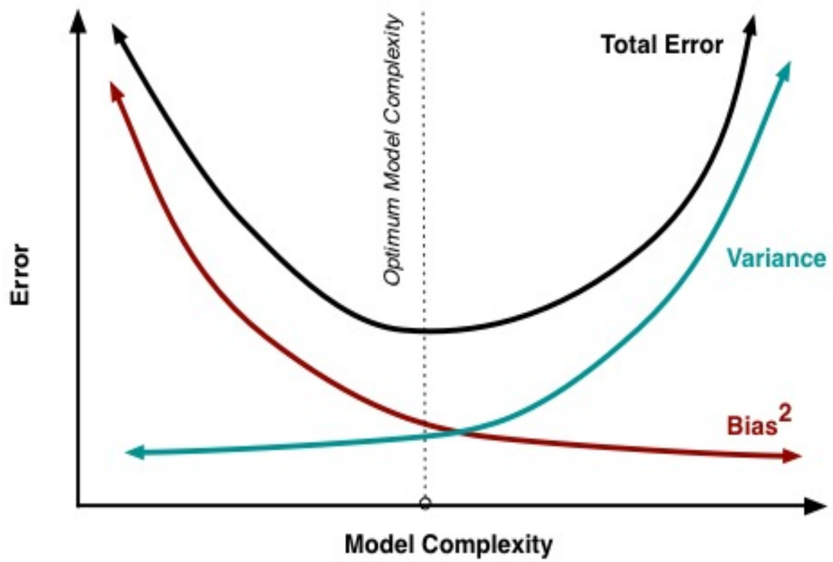
The generalization performance is determined by the ability of the learning algorithm, the sufficiency of the data, and the difficulty of the learning task itself.


Low bias
High bias
Truth
Low Variance
Underfitting
(a)
(b)
High Variance
Overfitting
(c)
(d)

Bias-Variance dilemma

In general, bias conflicts with variance:

- training is insufficient, the learning algorithm's fitting ability is not strong, and the bias dominates the error;
- If training process reached a certain degree, the algorithm's fitting ability becomes good, and then the variance dominates the error;
  - the learner starts to learn the pattern of training data perturbation.



- After get sufficient training, the algorithm's fitting ability becomes very strong, and the variance dominates the error.
  - very small data perturbation can influence model performance.

- Thank you!