Bài 12
HỌC CỘNG ĐỒNG (ENSEMBLE LEARNING)

AI Academy Vietnam

Nguyễn Quang Uy

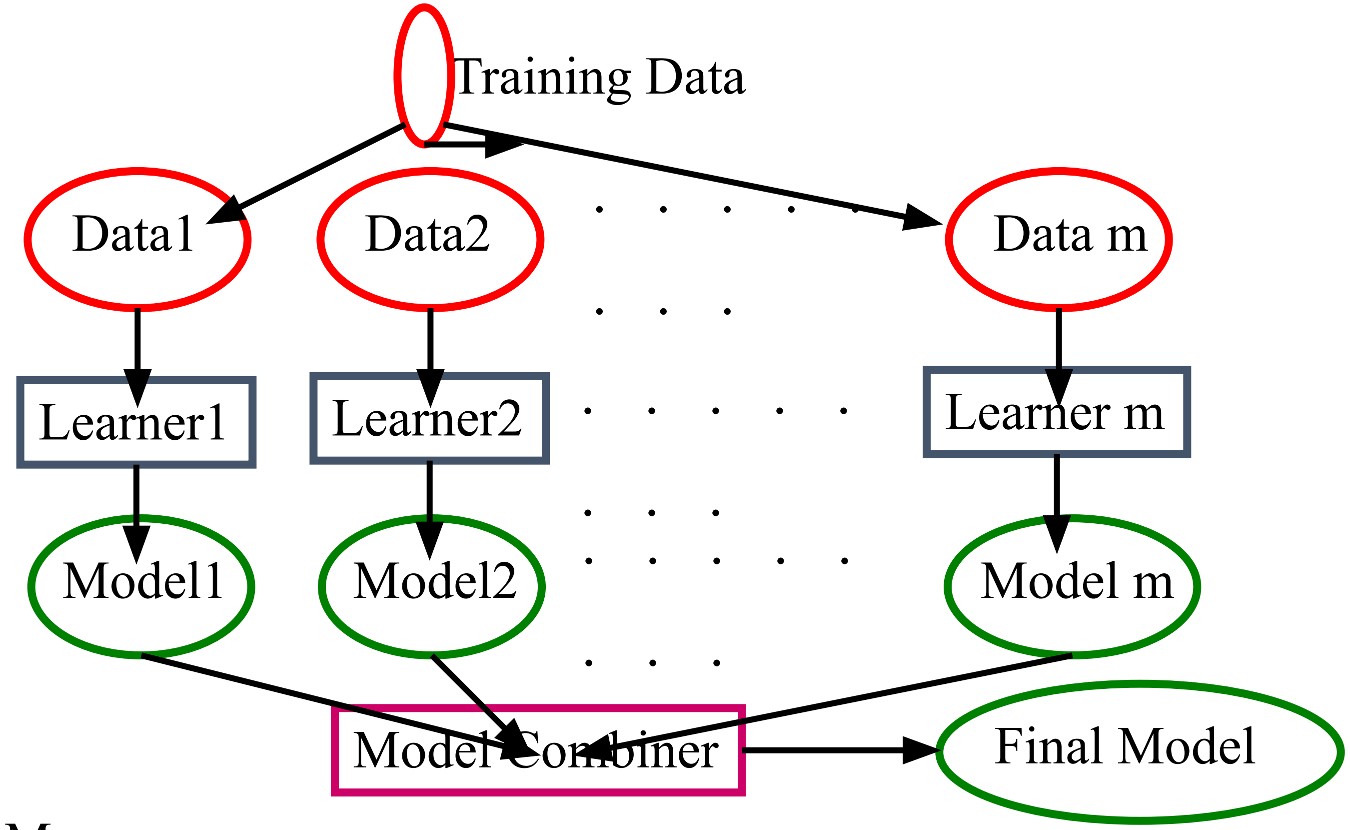
Nội dung buổi học

- ❖ Giới thiệu về ensemble learning
- ❖ Lý do xây dựng ensemble learning
- ❖ Phương pháp xây dựng ensemble learning
  - Bagging
  - Boosting
  - Random subspace ensemble
  - Random Forest
- ❖ Phương pháp kết hợp các bộ phân lớp

Giới thiệu về ensemble learning

Ensemble learning là gì?

- Set of Classifiers
- Decisions combined in ”some” way

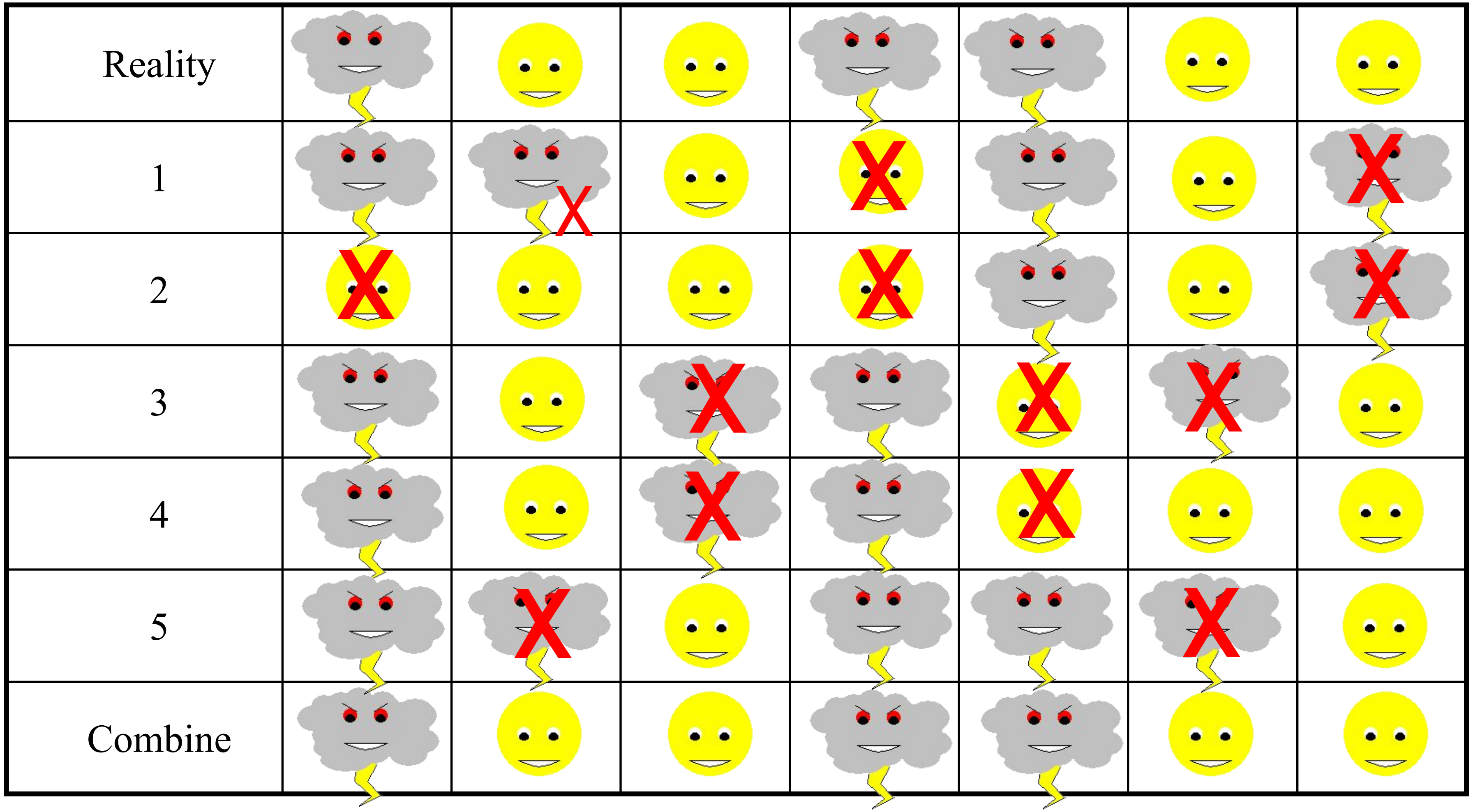


Source: Ray Mooney

Lợi ích của ensemble learning

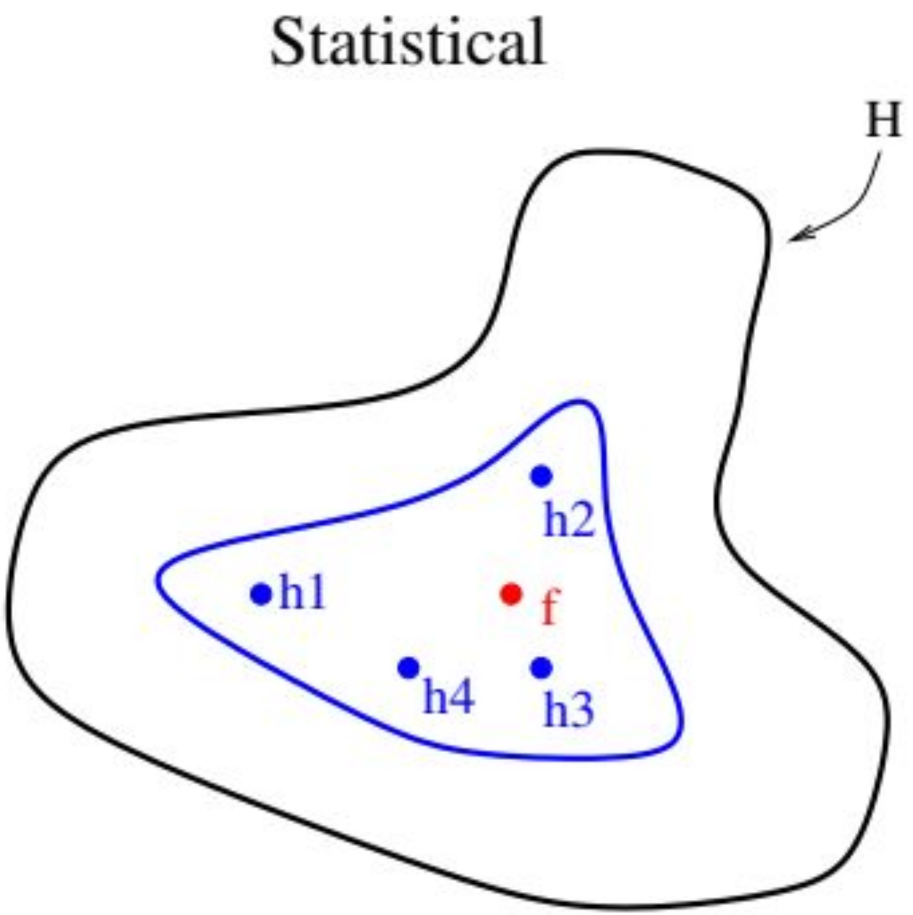
- “No Free Lunch” Theorem
  - No single algorithm wins all the time!
- When combining multiple independent and diverse decisions each of which is at least more accurate than random guessing, random errors cancel each other out, correct decisions are reinforced.
- Examples: Human ensembles are demonstrably better
  - How many jelly beans in the jar?: Individual estimates vs. group average.
  - Who Wants to be a Millionaire: Audience vote.

Lợi ích của ensemble learning

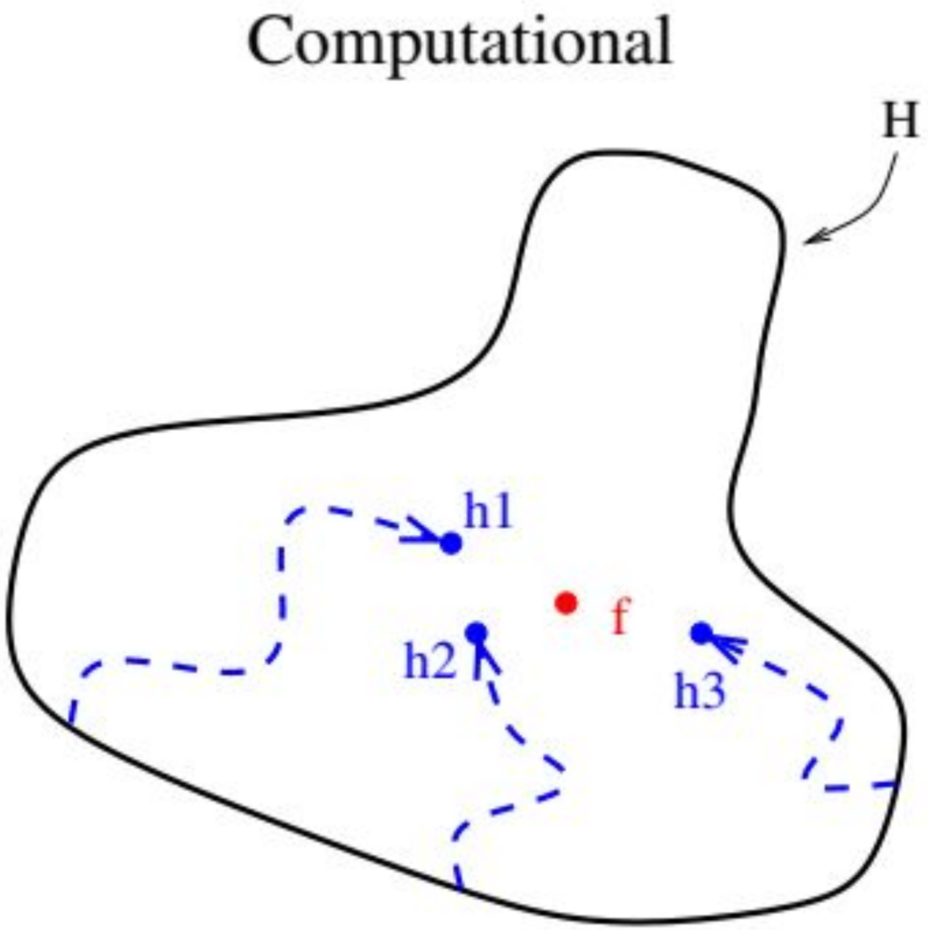


Lý do xây dựng ensemble learning

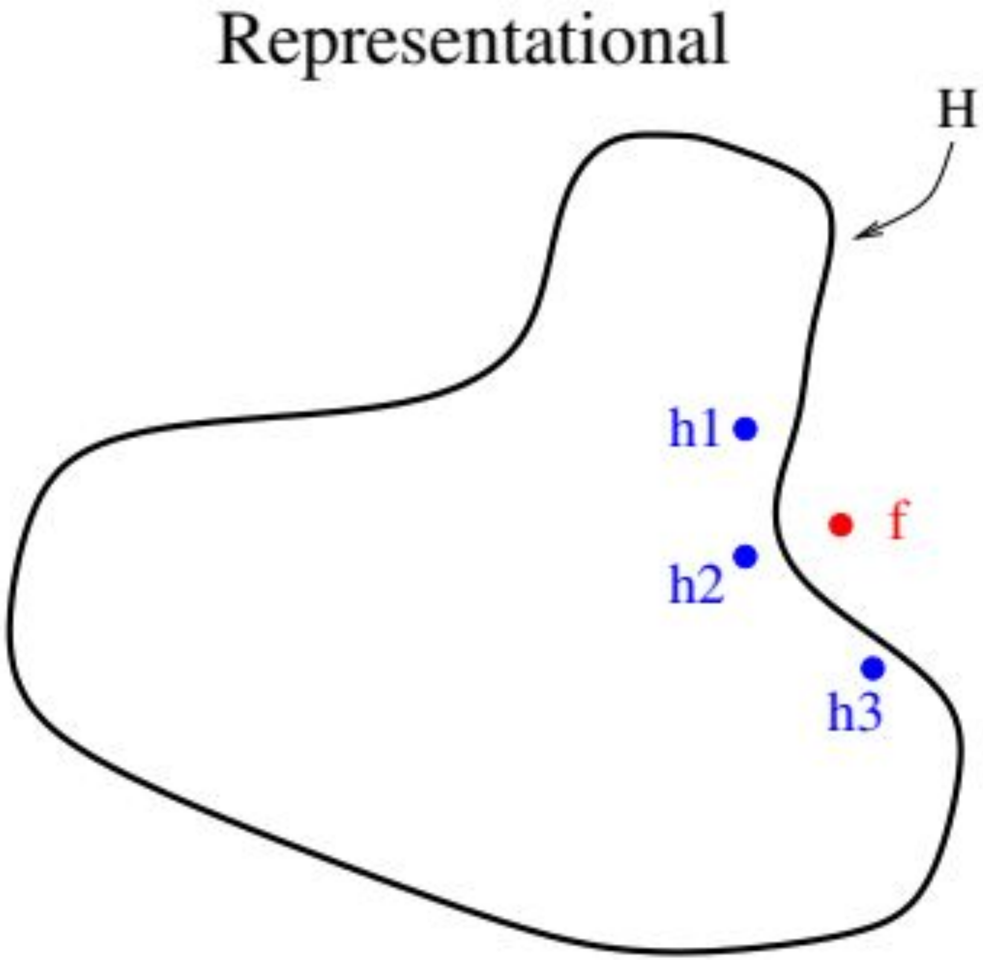
Lý do về xác suất



Lý do về tính toán



Lý do về biểu diễn



Phương pháp xây dựng

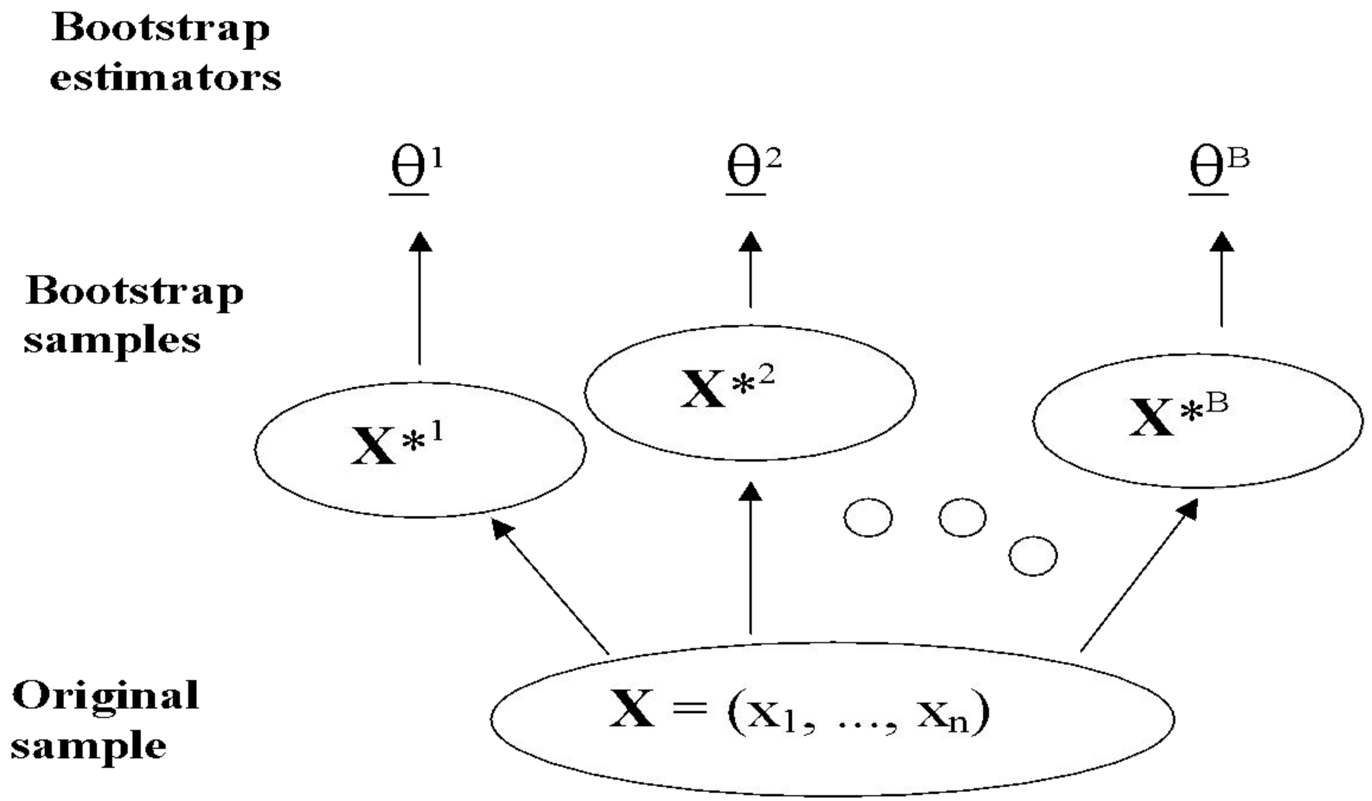
Phương pháp xây dựng ensemble

- Manipulating the Training Examples
  - ✔ Bagging
  - ✔ Boosting
- Manipulating the Input Features
  - ✔ Random subspace ensemble
  - ✔ Random forest
- Stacking

Phương pháp Bagging

- Create bootstrap replicates of training set
- Train a classifier (e.g., a decision tree) for each replicate
- Estimate classifier performance using out-‐of-‐bootstrap data
- Average output of all classifiers

Phương pháp Bagging



The final estimate: $\underline{\theta} = (\underline{\theta}^1 + \underline{\theta}^2 + \ldots + \underline{\theta}^B)/B$

Ưu và nhược điểm của Bagging

- The estimator can be significantly improved if the learning algorithm is unstable

- Degrade the performance of stable procedures

Phương pháp Boosting

- Turns a base learner (i.e., a “weak hypothesis”) into a high performance classifier

- Creates an ensemble of weak hypotheses by repeatedly emphasizing misspredicted instances

- Popular Boosting methods: Adaboost, Gradient boost,...

Phương pháp AdaBoost

- 1. Initially, all observations in the dataset are given equal weights.
- 2. A model is built on a subset of data.
- 3. Using this model, predictions are made on the whole dataset.
- 4. Errors are calculated by comparing the predictions and actual values.

Phương pháp AdaBoost

- 5. In the next model, higher weights are given to the data points which were predicted incorrectly.

- 6. Weights can be determined using the error value.

- 7. This process is repeated until the error function does not change, or the maximum limit of the number of estimators is reached.

Thuật toán AdaBoost

Initialize a vector of n uniform weights $\mathbf{w}_1$

for $t = 1, \ldots, T$

  Train model h_t on X, y with weights $\mathbf{w}_t$

  Compute the weighted training error of h_t

  Choose $\beta_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$

  Update all instance weights:

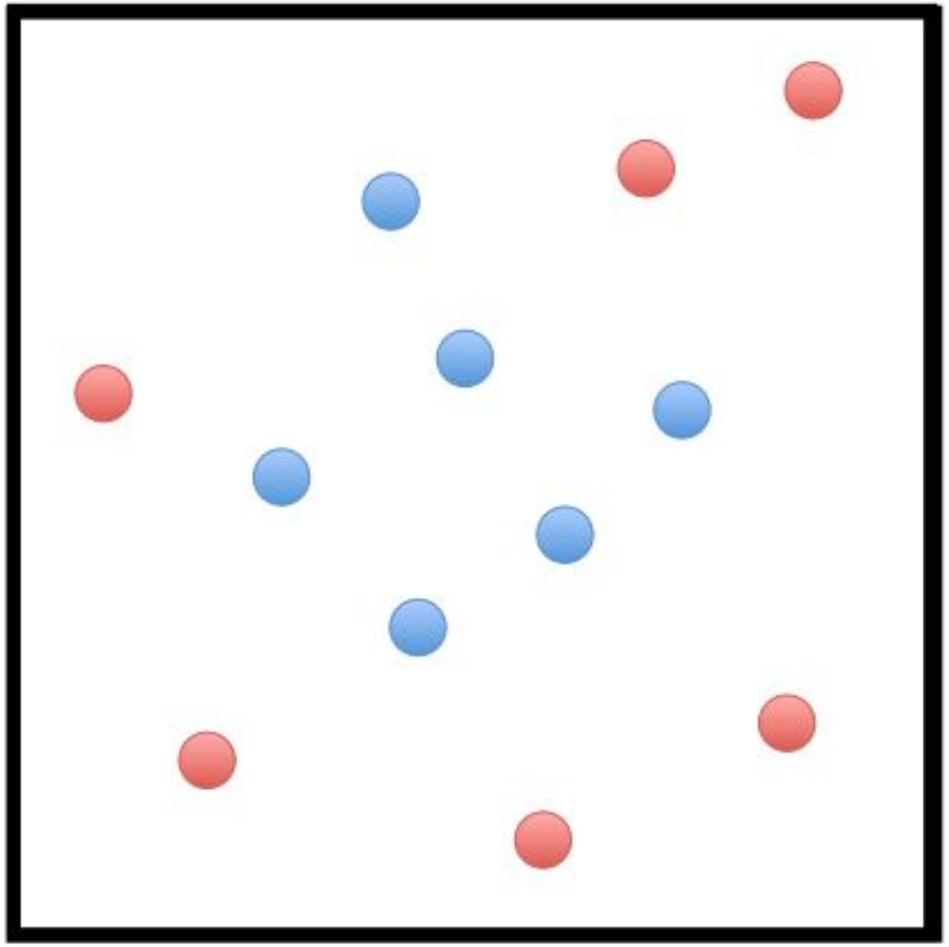
$$w_{t+1,i} = w_{t,i} \exp \left(-\beta_t y_i h_t(\mathbf{x}_i) \right)$$

  Normalize $\mathbf{w}_{t+1}$ to be a distribution

end for

Return the hypothesis

$$H(\mathbf{x}) = \text{sign} \left(\sum_{t=1}^{T} \beta_t h_t(\mathbf{x}) \right)$$



Gradient Boosting (GBM)

- Fit a model to the data, $F_1(x) = y$
- Fit a model to the residuals, $h_1(x) = y - F_1(x)$
- Create a new model, $F_2(x) = F_1(x) + h_1(x)$

Gradient Boosting (GBM)

- We can generalize this idea by inserting more models. Specifically:

- F(x) = F_1(x) -> F_2(x) = F_1(x) + h_1(x) .... -> F_M(x) = F_{M-1}(x) + h_{M-1}(x)

- where F_1(x) is an initial model fit to y

- Since we initialize the procedure by fitting F_1(x), our task at each step is to find h_m(x) = y - F_m(x).

Gradient Boosting (GBM)

ID	Married	Gender	Current City	Monthly Income	Age (target)
1	Y	M	A	51,000	35
2	N	F	B	25,000	24
3	Y	M	A	74,000	38
4	N	F	A	29,000	30
5	N	F	B	37,000	33

Gradient Boosting (GBM)

ID	Married	Gender	Current City	Monthly Income	Age (target)	Mean Age (prediction 1)	Residual 1
1	Y	M	A	51,000	35	32	3
2	N	F	B	25,000	24	32	-8
3	Y	M	A	74,000	38	32	6
4	N	F	A	29,000	30	32	-2
5	N	F	B	37,000	33	32	1

Gradient Boosting (GBM)

ID	Age (target)	Mean Age (prediction 1)	Residual 1 (new target)	Prediction 2	Combine (mean+pred2)
1	35	32	3	3	35
2	24	32	-8	-5	27
3	38	32	6	3	35
4	30	32	-2	-5	27
5	33	32	1	3	35

Gradient Boosting (GBM)

ID	Age (target)	Mean Age (prediction 1)	Residual 1 (new target)	Prediction 2	Combine (mean+pred2)	Residual 2 (latest target)
1	35	32	3	3	35	0
2	24	32	-8	-5	27	-3
3	38	32	6	3	35	-3
4	30	32	-2	-5	27	3
5	33	32	1	3	35	-2

Ưu và nhược của AdaBoost

- Theoretical guarantee (maximizes the likelihood)
- Often better than bagging
- Need large number of base classifier
- Sensitive to noise

Bagging vs Boosting

A sample of a single classifier on an imaginary set of data.	
	(Original) Training Set
Training-set-1:	1, 2, 3, 4, 5, 6, 7, 8

A sample of Bagging on the same data.	
	(Resampled) Training Set
Training-set-1:	2, 7, 8, 3, 7, 6, 3, 1
Training-set-2:	7, 8, 5, 6, 4, 2, 7, 1
Training-set-3:	3, 6, 2, 7, 5, 6, 2, 2
Training-set-4:	4, 5, 1, 4, 6, 4, 3, 8

A sample of Boosting on the same data.	
	(Resampled) Training Set
Training-set-1:	2, 7, 8, 3, 7, 6, 3, 1
Training-set-2:	1, 4, 5, 4, 1, 5, 6, 4
Training-set-3:	7, 1, 5, 8, 1, 8, 1, 4
Training-set-4:	1, 1, 6, 1, 1, 3, 1, 5

Phương pháp Random Subspace

- Randomly selecting subsets of components of the feature vector
- Each classifier constructed in the randomly chosen subspaces
- Often good for datasets having numerous features but less samples

Tin Kam Ho, 1998. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell*, *20*(8), pp.1-22.

Thuật toán Random Forest

- **Random forest** (or **random forests**) is an ensemble classifier that consists of many decision trees and outputs the class that is the ensemble of the class's output by individual trees.

- The term came from **random decision forests** that was first proposed by Tin Kam Ho of Bell Labs in 1995.

- The method combines Breiman's "bagging" idea and the random selection of features.

Thuật toán Random Forest

Each tree is constructed using the following algorithm:

1. Let the number of training cases be *N*, and the number of variables in the classifier be *M*.
2. We are told the number *m* of input variables to be used to determine the decision at a node of the tree; *m* should be much less than *M*.
3. Choose a training set for this tree by choosing *n* times with replacement from all *N* available training cases (i.e. take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.
4. For each node of the tree, randomly choose *m* variables on which to base the decision at that node. Calculate the best split based on these *m* variables in the training set.
5. Each tree is grown as be done in constructing a normal tree classifier.

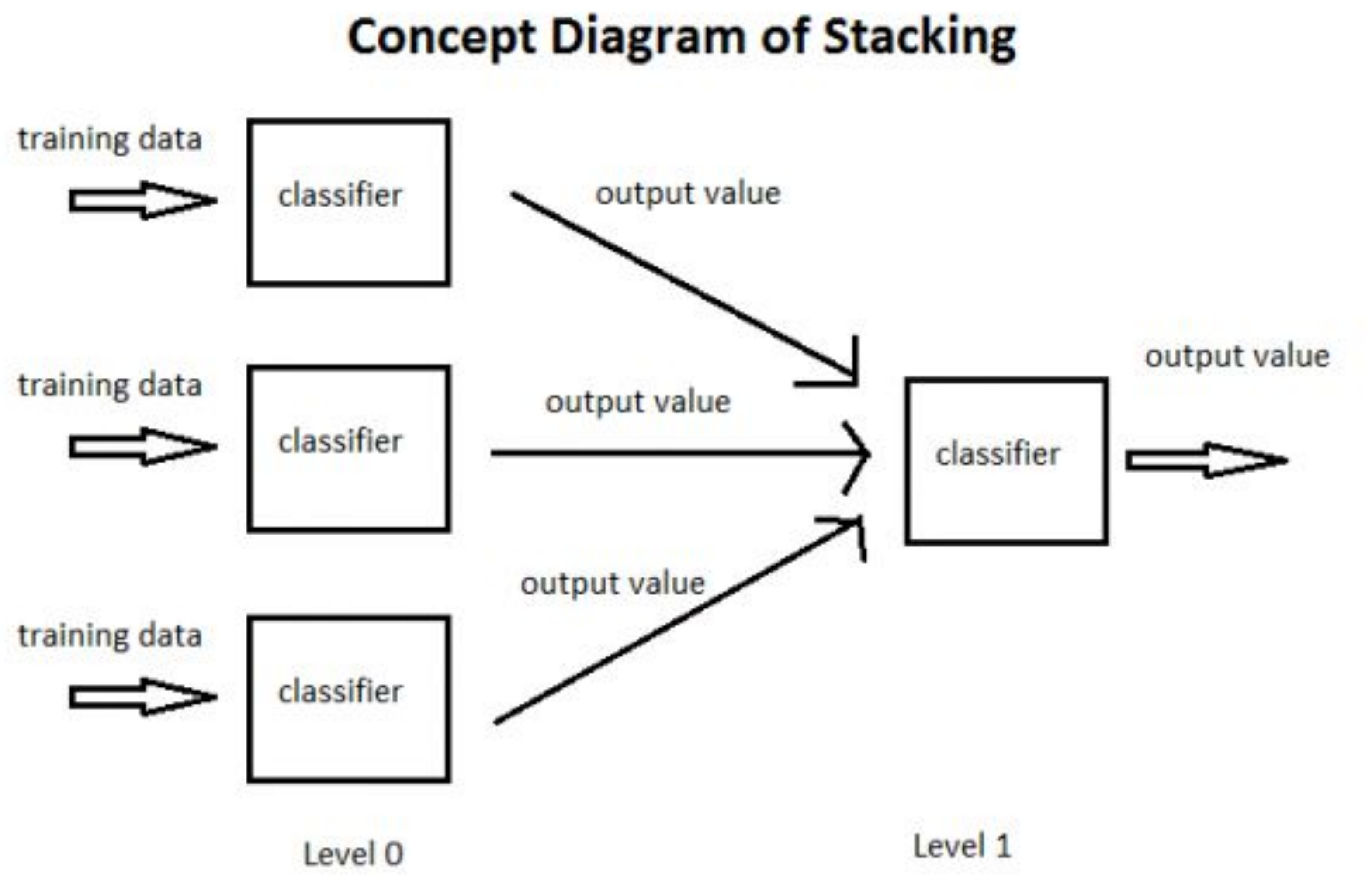
Stacking

- Stacking is an ensemble learning technique that combines multiple models via a meta-model.

- The base level models are trained on a complete training set.

- The meta-model is trained on the outputs of the base level model as features.

Stacking



Concept Diagram of Stacking
training data
classifier
output value
training data
classifier
output value
training data
classifier
output value
classifier
output value
Level 0
Level 1

Stacking

Algorithm Stacking

Input: training data $D = \{x_i, y_i\}_{i=1}^{m}$

Ouput: ensemble classifier H

Step 1: learn base-level classifiers

for $t = 1$ to T **do**

  learn h_t based on D

end for

Step 2: construct new data set of predictions

for $i = 1$ to m **do**

  $D_h = \{x'_i, y_i\}$, where $x'_i = \{h_1(x_i), ..., h_T(x_i)\}$

end for

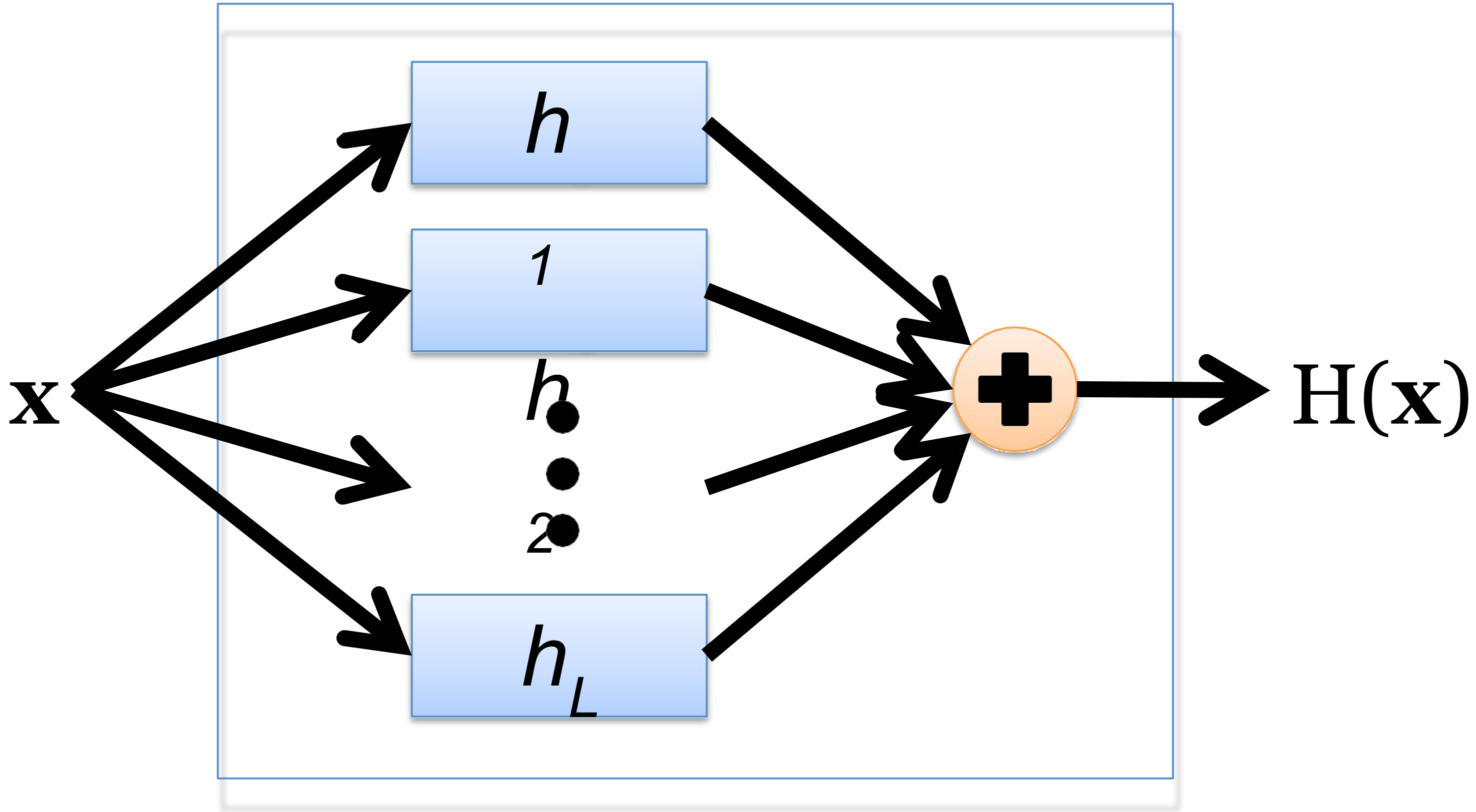
Step 3: learn a meta-classifier

learn H based on D_h

return H

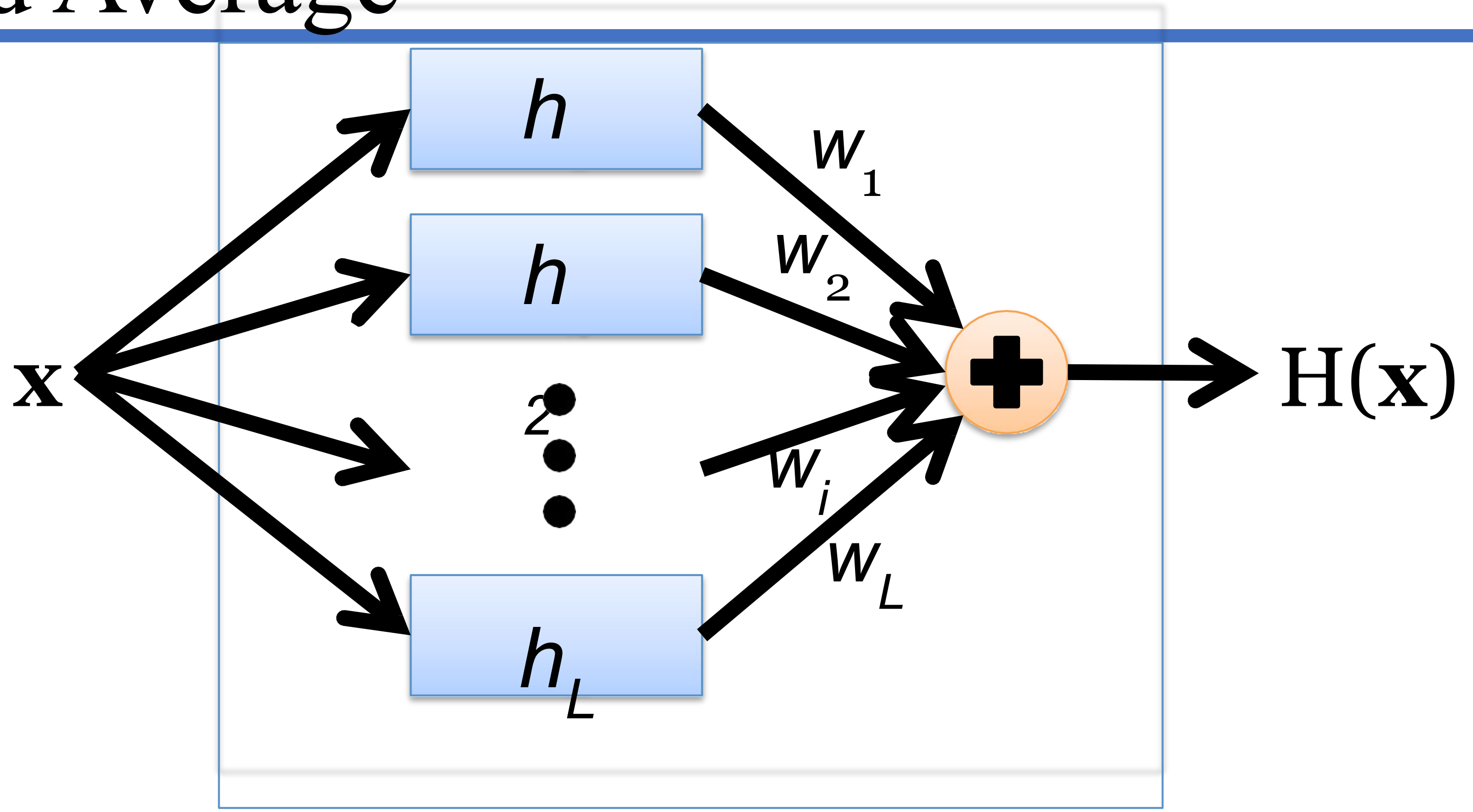
Kết hợp các bộ phân lớp

Phương pháp kết hợp



- Final hypothesis is a simple vote of the members

Combining Classifiers: Weighted Average



- Coefficients of individual members are trained using a validation set

Q&A

Thank you!

Thank you.