

Các phương pháp học máy kết hợp

Bagging và Random Forests, Boosting, Stacking

Nguyễn Thanh Tùng

Khoa Công nghệ thông tin – Đại học Thủ Dầu Một

tungnt@tlu.edu.vn

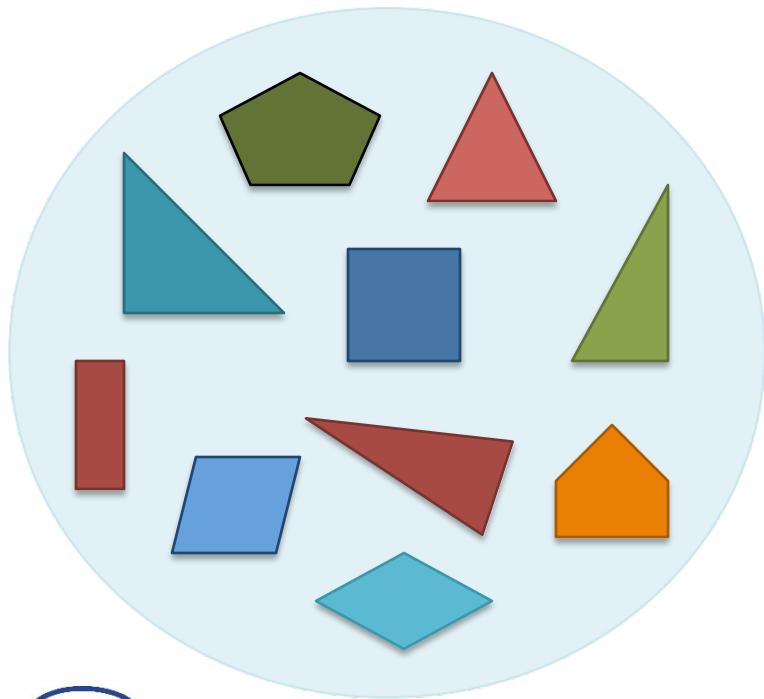
Đánh giá hiệu quả mô hình phân lớp

Phân lớp

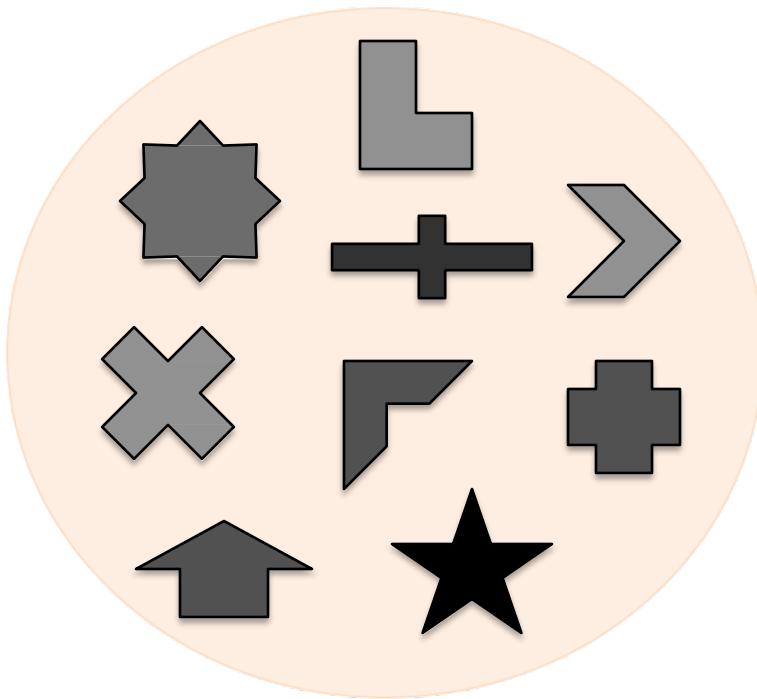
- Học có giám sát: Học từ các mẫu đã gán nhãn
- Biến đích có dạng rời rạc / hạng mục
- Mục tiêu: dự đoán biến đích có kiểu rời rạc
 - Gán mỗi mẫu cho 1 lớp
 - Các bài trước: K-NN, CART
 - Hôm nay: Bagging, Random Forests

Học từ mẫu đã gán nhãn

Lớp “+”

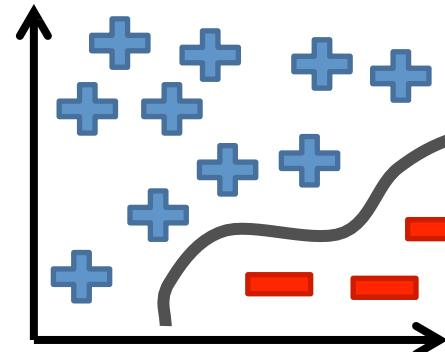
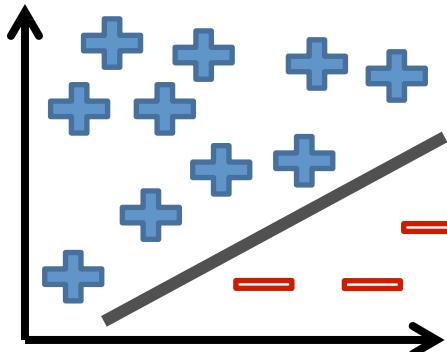


Lớp “-”



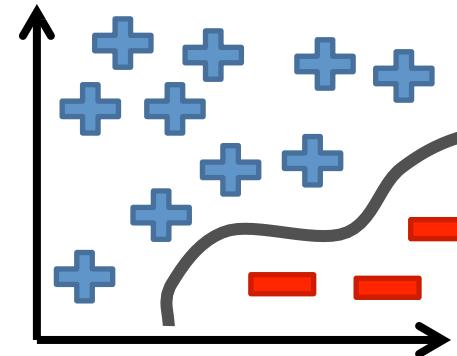
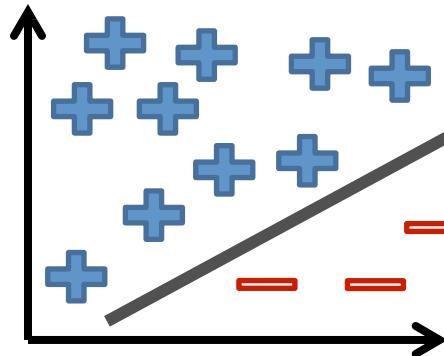
Nhãn mất cân bằng

- Nhãn mất cân bằng (Imbalanced classes): lớp dương (+) xuất hiện với tần suất nhiều hơn lớp âm (-) trong tập dữ liệu huấn luyện
 - vd: phát hiện gian lận, dữ liệu y học



Nhận mất cân bằng

- Tại sao đây là vấn đề?
 - Các thuật toán thực hiện tốt khi huấn luyện trên các mẫu trong mỗi lớp
 - Hiệu quả thấp trên các lớp có ít đại diện



Đánh giá hiệu quả bộ phân lớp

- Hiệu năng của một mô hình thường được đánh giá dựa trên tập dữ liệu kiểm thử (test data).
- Giả sử y_{pred} là vector dự đoán đầu ra với mỗi phần tử là class được dự đoán của một điểm dữ liệu trong tập kiểm thử.
- Ta cần so sánh giữa vector dự đoán y_{pred} này với vector class thật của dữ liệu, được mô tả bởi vector y_{true} .

Đánh giá hiệu quả bộ phân lớp

- Ví dụ với bài toán có 3 lớp dữ liệu được gán nhãn là **0, 1, 2**.
- Giả sử các class được đánh số từ **0** đến **C-1** trong trường hợp có **C** lớp dữ liệu. Có 10 điểm dữ liệu trong tập kiểm thử với các nhãn thực sự được mô tả bởi
 $y_{\text{true}} = [0, 0, 0, 0, 1, 1, 1, 2, 2, 2]$.
- Giả sử bộ phân lớp chúng ta đang cần đánh giá dự đoán nhãn cho các điểm này là
 $y_{\text{pred}} = [0, 1, 0, 2, 1, 1, 0, 2, 1, 2]$.

Đánh giá hiệu quả bộ phân lớp

- Trong bài toán hồi quy, chúng ta đã dùng MSE, RMSE, MAE, R² để đánh giá hiệu quả thuật toán.
- Với bài toán phân lớp, chúng ta cần độ đo để đánh giá hiệu quả của mô hình phân lớp
 - Có nhiều độ đo: Độ chính xác (Accuracy), Ma trận nhầm lẫn (Confusion matrix), Độ chính xác/Hồi tưởng (Precision/Recall), Độ nhạy/Độ đặc hiệu (Sensitivity/ Specificity), Đường cong ROC (ROC curve)

Đánh giá hiệu quả bộ phân lớp

- Accuracy: Cách đánh giá này đơn giản tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử.
- Trong ví dụ trước, ta có thể đếm được có 6 điểm dữ liệu được dự đoán đúng trên tổng số 10 điểm.
- Vậy ta kết luận độ chính xác của mô hình là 0.6 (hay 60%).

Đánh giá hiệu quả bộ phân lớp

- Accuracy chỉ cho ta biết được bao nhiêu phần trăm lượng dữ liệu được phân loại đúng.
- Confusion matrix: Chỉ ra được cụ thể mỗi loại được phân loại như thế nào, lớp nào được phân loại đúng nhiều nhất, và dữ liệu thuộc lớp nào thường bị phân loại nhầm vào lớp khác.

Total: 10	Predicted as: 0	Predicted as: 1	Predicted as: 2	
True: 0	2	1	1	4
True: 1	1	2	0	3
True: 2	0	1	2	3

Là một ma trận vuông với số chiều bằng số lượng lớp dữ liệu. Giá trị tại hàng thứ i , cột thứ j là số lượng điểm i là **đúng** và j là **nhầm**. Như vậy, nhìn vào hàng thứ nhất (**0**), ta có thể thấy được rằng trong số bốn điểm thực sự thuộc lớp **0**, chỉ có hai điểm được phân loại đúng, hai điểm còn lại bị phân loại nhầm vào lớp **1** và lớp **2**.

Đánh giá hiệu quả bộ phân lớp

- True/False Positive/Negative
- Xét bài toán phân lớp nhị phân: Có 2 lớp (+) và (-)
- Trong hai lớp dữ liệu này có một lớp nghiêm trọng hơn lớp kia và cần được dự đoán chính xác.
- Ví dụ:
 - Trong bài toán xác định có bệnh ung thư hay không thì việc không bị sót (miss) quan trọng hơn là việc chẩn đoán nhầm âm tính thành dương tính.
 - Trong bài toán xác định có mìn dưới lòng đất hay không thì việc bỏ sót nghiêm trọng hơn việc báo động nhầm rất nhiều.

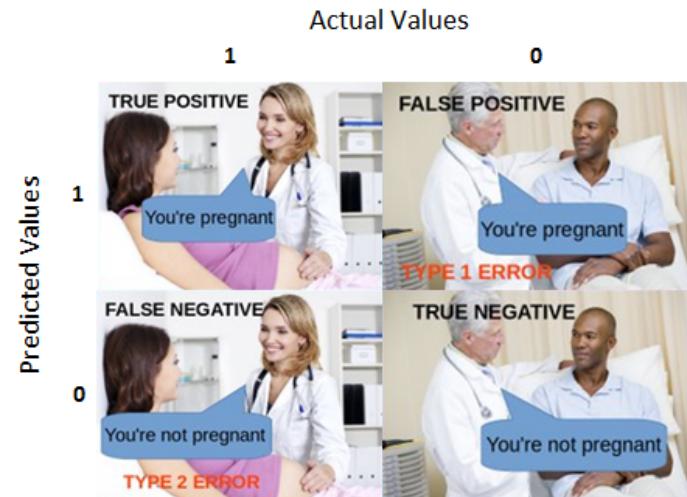
Đánh giá hiệu quả bộ phân lớp

- Định nghĩa lớp dữ liệu quan trọng hơn cần được xác định đúng là lớp Positive (P-dương tính), lớp còn lại được gọi là Negative (N-âm tính).
- Định nghĩa True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN) dựa trên confusion matrix như sau:

		Lớp dự đoán (Predicted class)	
		+	-
Lớp thực (True class)	+	True Positive-TP	False Negative-FN Type II error
	-	(False Positive-FP) Type I error	True Negative-TN

Đánh giá hiệu quả bộ phân lớp

- Bệnh nhân được chẩn đoán là có thai và thực tế đúng là như vậy: True Positive (TP), **dương tính thật**.
- Chẩn đoán là có thai nhưng thực tế không phải vậy: False Positive (FP), **dương tính giả**.
- Bệnh nhân được chẩn đoán là không có thai nhưng thực tế người này đang mang bầu. Tình huống này được gọi là **âm tính giả**, kí hiệu là FN.
- Bệnh nhân được chẩn đoán là không có thai và thực tế đúng như vậy. Tình huống này được gọi là **âm tính thật**, kí hiệu là TN.



<https://rpubs.com/chidungkt/447989>

Đánh giá hiệu quả bộ phân lớp

True positive rate (TPR)

(recall, sensitivity)

Predicted class

		+	-
True class	+	TP	FN
	-	FP	TN

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

False negative rate (FNR)

Predicted class

		+	-
True class	+	TP	FN
	-	FP	TN

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

False positive rate (FPR)

Predicted class

		+	-
True class	+	TP	FN
	-	FP	TN

$$FPR = \frac{FP}{FP + TN}$$

True negative rate (SPC)

(specificity)

Predicted class

		+	-
True class	+	TP	FN
	-	FP	TN

$$SPC = \frac{TN}{FP + TN}$$

Đánh giá hiệu quả bộ phân lớp

- False Positive Rate còn được gọi là False Alarm Rate (tỉ lệ báo động nhầm)
- False Negative Rate còn được gọi là Miss Detection Rate (tỉ lệ bỏ sót).
- Trong bài toán dò mìn, thà báo nhầm còn hơn bỏ sót, tức là ta có thể chấp nhận False Alarm Rate cao để đạt được Miss Detection Rate thấp.

Chú ý: Việc biết một cột của confusion matrix này sẽ suy ra được cột còn lại vì tổng các hàng luôn bằng 1 và chỉ có hai lớp dữ liệu. **Với các bài toán có nhiều lớp dữ liệu**, ta có thể xây dựng bảng True/False Positive/Negative cho **mỗi lớp** nếu coi lớp đó là lớp *Positive*, các lớp còn lại gộp chung thành lớp *Negative*.



Đánh giá hiệu quả bộ phân lớp

Precision/recall

True positive rate (TPR)

(recall, sensitivity)

Predicted class

		+	-
True class	+	TP	FN
	FP	TN	

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

Positive predictive value (PPV) (precision)

Predicted class

		+	-
True class	+	TP	FN
	FP	TN	

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

False positive rate (FPR)

Predicted class

		+	-
True class	+	TP	FN
	FP	TN	

$$FPR = \frac{FP}{FP + TN}$$

True negative rate (SPC)

(specificity)

Predicted class

		+	-
True class	+	TP	FN
	FP	TN	

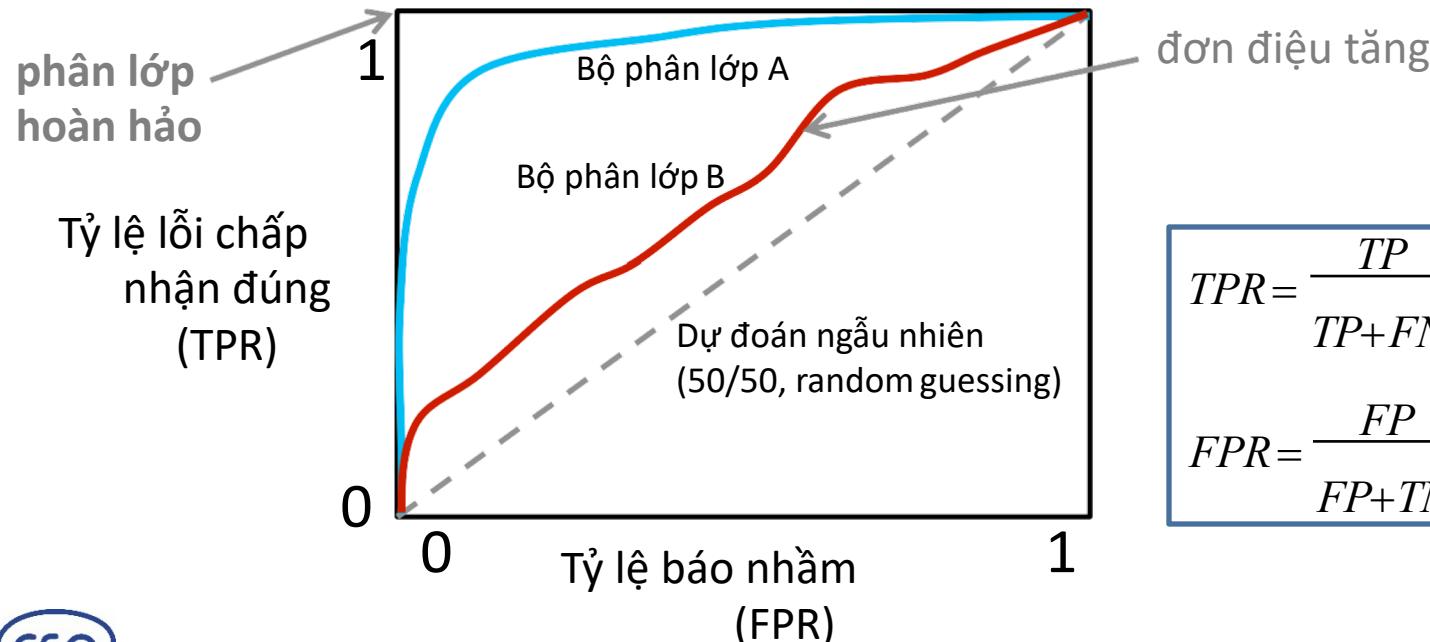
$$SPC = \frac{TN}{FP + TN}$$

ROC curve



Đánh giá hiệu quả bộ phân lớp

- Đường cong ROC (receiver operating characteristic)



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

$$FPR = \frac{FP}{FP+TN}$$

Đánh giá hiệu quả bộ phân lớp

- Nhược điểm của đường cong ROC
 - ROC không biểu thị đúng độ mất cân bằng các mẫu trong lớp thực
 - vd: Xét bộ dữ liệu có 1% mẫu thuộc lớp "+" và 99% mẫu thuộc lớp "-"
 - Giả sử ta nhận được kết quả phân lớp như sau:
 $TPR = 0.9$ và $FPR = 0.12$
 - TPR và FPR không hiểu thi được theo tính chất của đường cong ROC

		Predicted class	
		+	-
True class	+	90	10
	-	1188	8712

Đánh giá hiệu quả bộ phân lớp

- Độ chính xác/Triệu hồi (Precision/recall)
 - Độ chính xác (Positive predictive value): $PPV = \frac{TP}{TP+FP}$
 - Tỷ lệ phần trăm của số mẫu thuộc lớp (+) được dự đoán đúng trên số mẫu thực là (+)
 - Recall (True positive rate): $TPR = \frac{TP}{TP+FN} = \frac{TP}{P}$
 - Tỷ lệ các mẫu (+) phân lớp chính xác lớp (+)
 - Recall và precision tỷ lệ nghịch với nhau
 - Với bộ phân lớp hoàn hảo, Recall = 1, Precision = 1
 - VD phân lớp mất cân bằng: Recall = 0.9, Precision = 0.07

		Predicted class	
		+	-
True class	+	90	10
	-	1188	8712

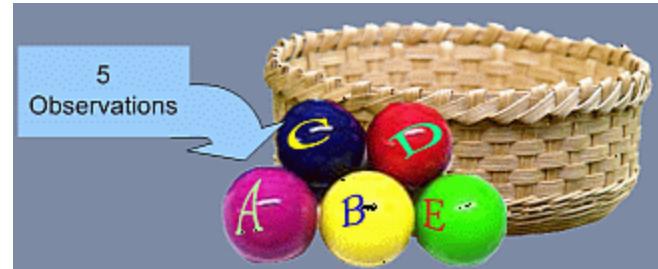


Phương pháp học máy kết hợp

Bootstrap là gì?

- Giả sử ta có 5 quả bóng gắn nhãn A,B,C,D, E và bỏ tất cả chúng vào trong 1 cái giỏ.
- Lấy ra ngẫu nhiên 1 quả từ giỏ và ghi lại nhãn, sau đó bỏ lại quả bóng vừa bốc được vào giỏ.
- Tiếp tục lấy ra ngẫu nhiên một quả bóng và lặp lại quá trình trên cho đến khi việc lấy mẫu kết thúc. Việc lấy mẫu này gọi là lấy mẫu có hoàn lại.
- Kết quả của việc lấy mẫu như trên có thể như sau (giả sử kích thước mẫu là 10):

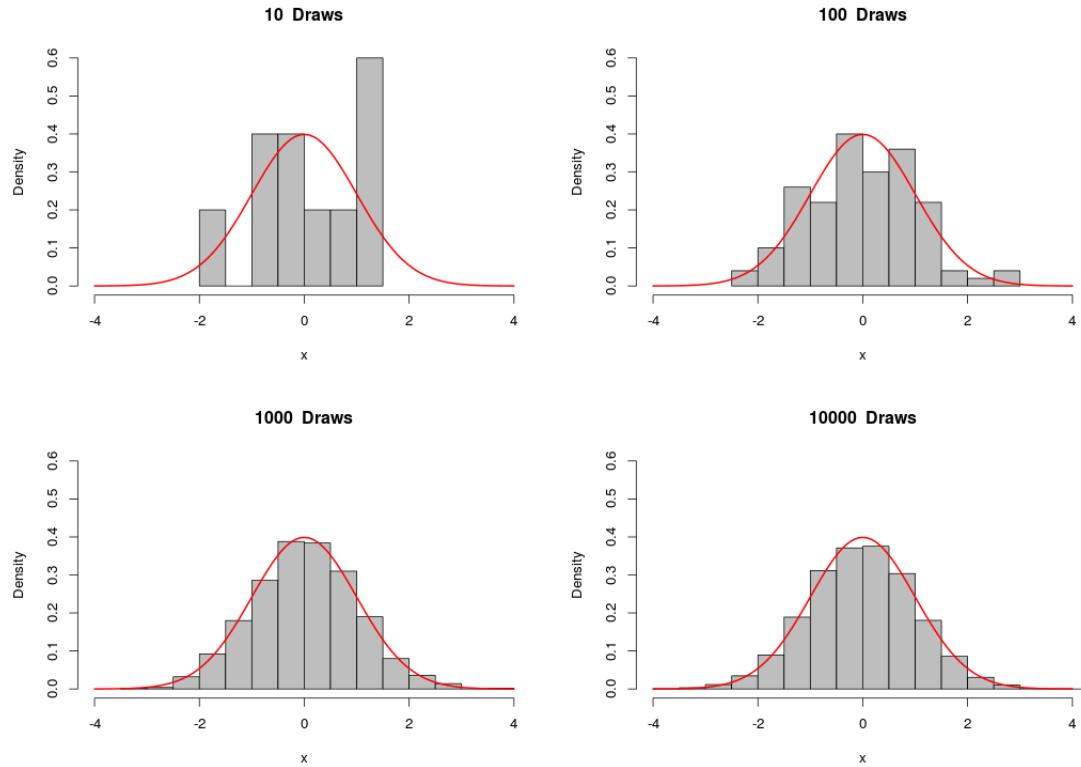
C, D, E, E, A, B, C, B, A, E



Nguồn: bis.net.vn/forums

Bootstrap là gì?

- Bootstrap là phương pháp lấy mẫu có hoàn lại (sampling with replacement) -> một mẫu có thể xuất hiện nhiều lần trong một lần lấy mẫu



Bootstrap là gì?

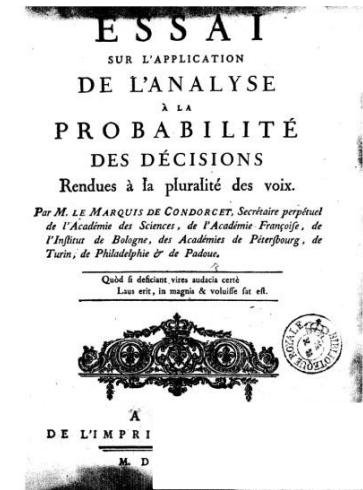
- Là kỹ thuật rất quan trọng trong thống kê
- Lấy mẫu có hoàn lại từ tập dữ liệu ban đầu để tạo ra các tập dữ liệu mới

Các phương pháp kết hợp Ensemble Methods

Sức mạnh của các bộ phận lớp yếu

Condorcet's Jury Theorem – Nếu p lớn

hơn 1/2 (mỗi cử tri bỏ phiếu đúng mong muốn của họ), càng thêm nhiều cử tri sẽ tăng xác suất theo quyết định số đông sẽ chính xác. Trong giới hạn, xác suất bầu chọn theo số đông tiến đến 1 khi số cử tri tăng lên.



Source gallica.bnf.fr / Bibliothèque nationale de France

Sức mạnh của các bộ phân lớp yếu

Condorcet's Jury Theorem – Nếu p lớn

hơn 1/2 (mỗi cử tri bỏ phiếu đúng mong muốn của họ), càng thêm nhiều cử tri sẽ tăng xác suất theo quyết định số đông sẽ chính xác. Trong giới hạn, xác suất bầu chọn theo số đông tiến đến 1 khi số cử tri tăng lên.



Sức mạnh của các bộ phân lớp yếu

- Việc lấy trung bình làm giảm phương sai và không làm tăng bias (bias vẫn được giữ nguyên) $\text{Var}[\bar{Y}] = \sigma^2/n$

Sức mạnh của các bộ phân lớp yếu

- Việc lấy trung bình làm giảm phương sai và không làm tăng bias (bias vẫn được giữ nguyên) $\text{Var}[\bar{Y}] = \sigma^2/n$
- Các phiếu bầu của các bộ phân lớp tương quan không trợ giúp được nhiều

THE CHOICE OF A CANDIDATE

THE NEW YORK TIMES supported Franklin D. Roosevelt for the Presidency in 1932 and again in 1936. In 1940 it will support Wendell Willkie.

Sức mạnh của các bộ phân lớp yếu

- Việc lấy trung bình làm giảm phương sai và không làm tăng bias (bias vẫn được giữ nguyên) $\text{Var}[\bar{Y}] = \sigma^2/n$
- Các phiếu bầu của các bộ phân lớp tương quan không trợ giúp được nhiều $\text{Var}[\bar{Y}] = \sigma^2/n + (\rho\sigma^2)(n-1)/n$

THE CHOICE OF A CANDIDATE

THE NEW YORK TIMES supported Franklin D. Roosevelt for the Presidency in 1932 and again in 1936. In 1940 it will support Wendell Willkie.

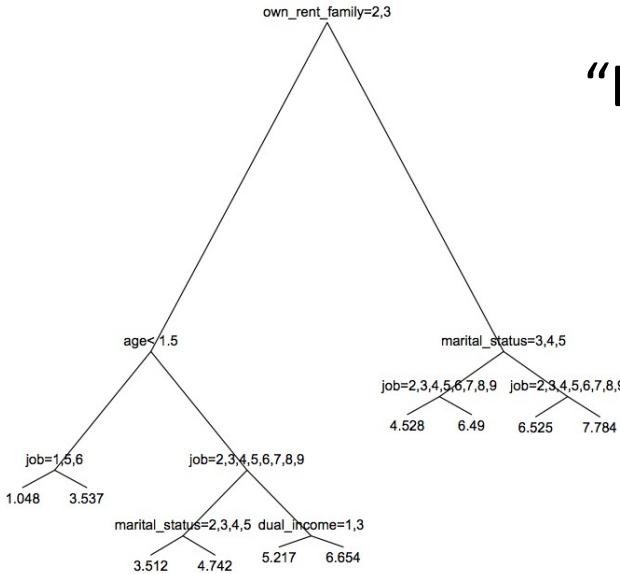
Kết hợp các bộ phân lớp

$$\alpha \times \{CART\} + (1-\alpha) \times \{LinearModel\}$$

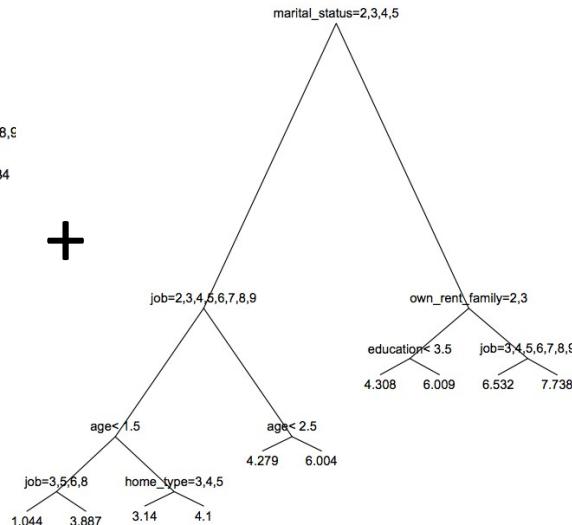
Các phương pháp kết hợp: Bagging

Bagging là gì?

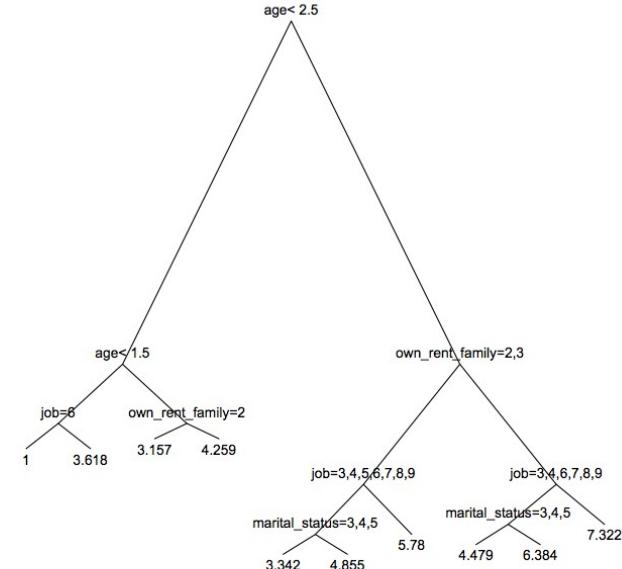
“Bootstrap Aggregation”



+

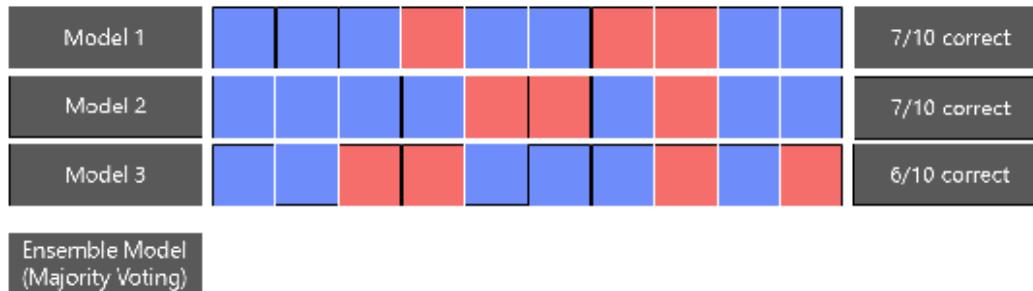


+



Bagging là gì?

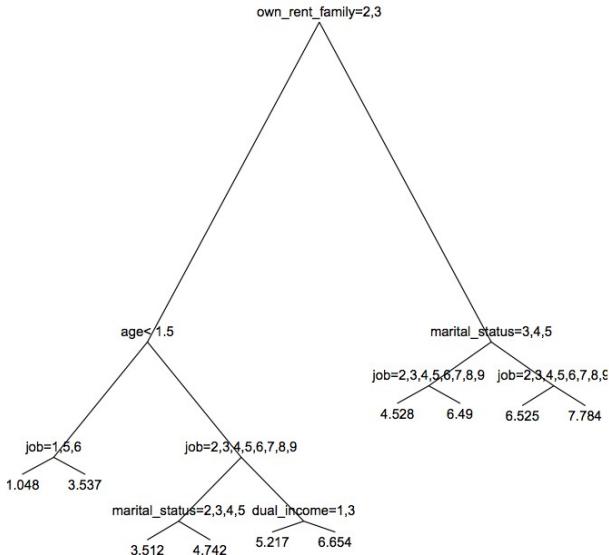
“Bootstrap Aggregation” $\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^{*b}(x)$



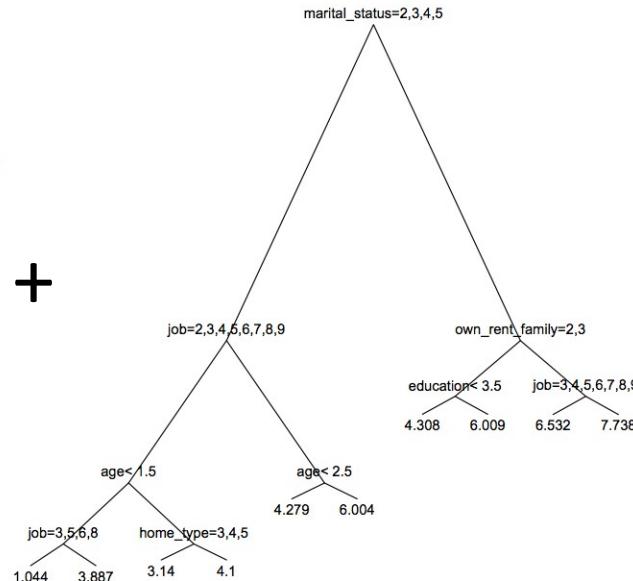
For more tutorials: algobeans.com

Bagging

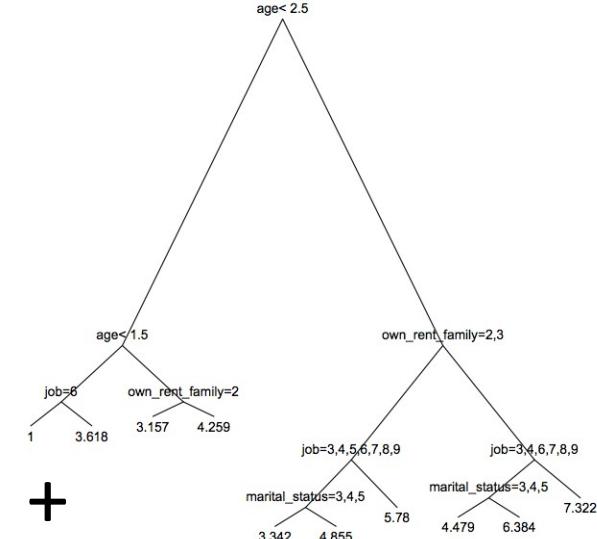
Giải quyết được tính thiếu ổn định của CART



+

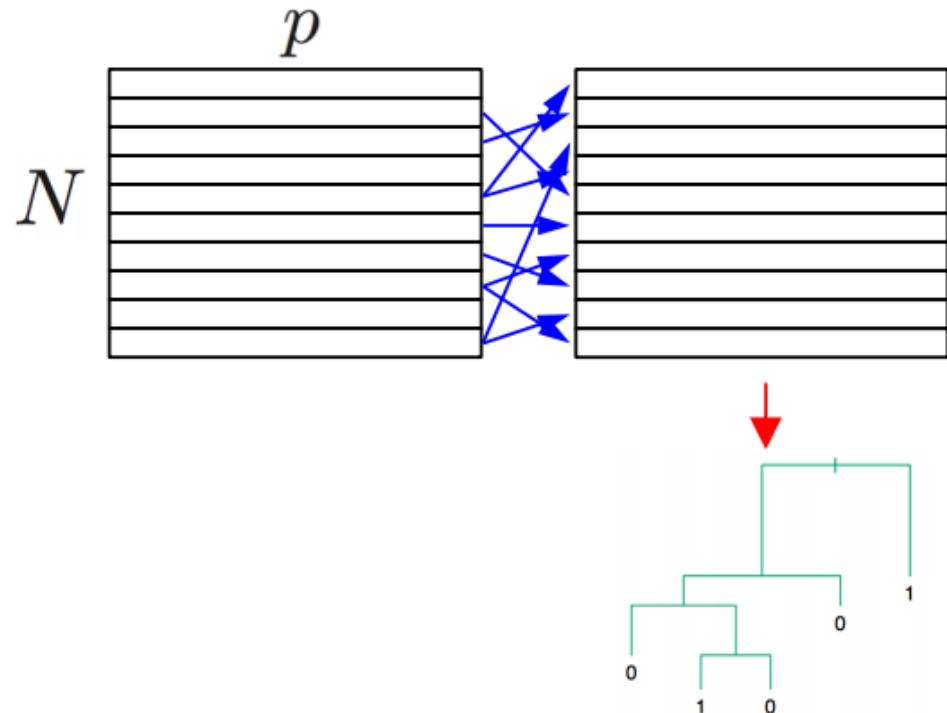


+



Bagging

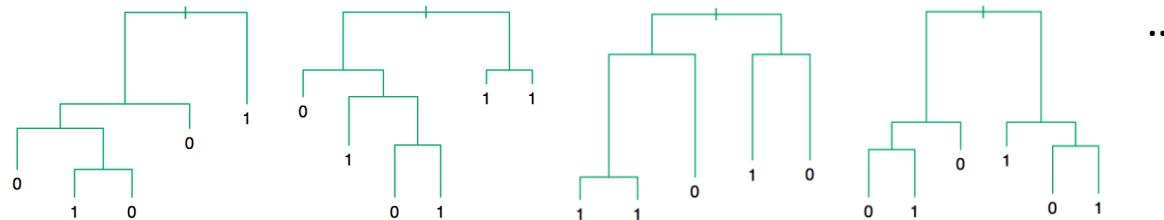
- Lấy mẫu tập dữ liệu huấn luyện theo Bootstrap để tạo ra tập hợp các dự đoán.



Bagging

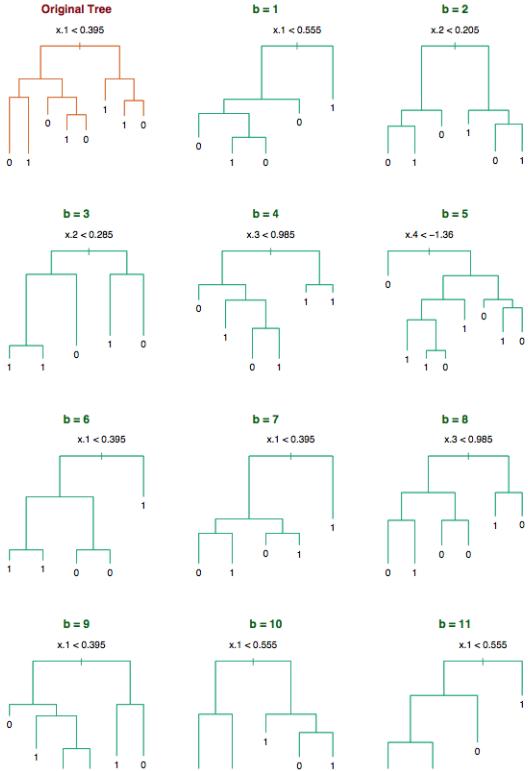
- Lấy mẫu tập dữ liệu huấn luyện theo Bootstrap để tạo ra tập hợp các dự đoán.

Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.



- Lấy trung bình (hoặc bình chọn theo số đông- majority vote) các bộ dự đoán độc lập.
- Bagging giảm phương sai (variance) và giữ bias.

Bagging



Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

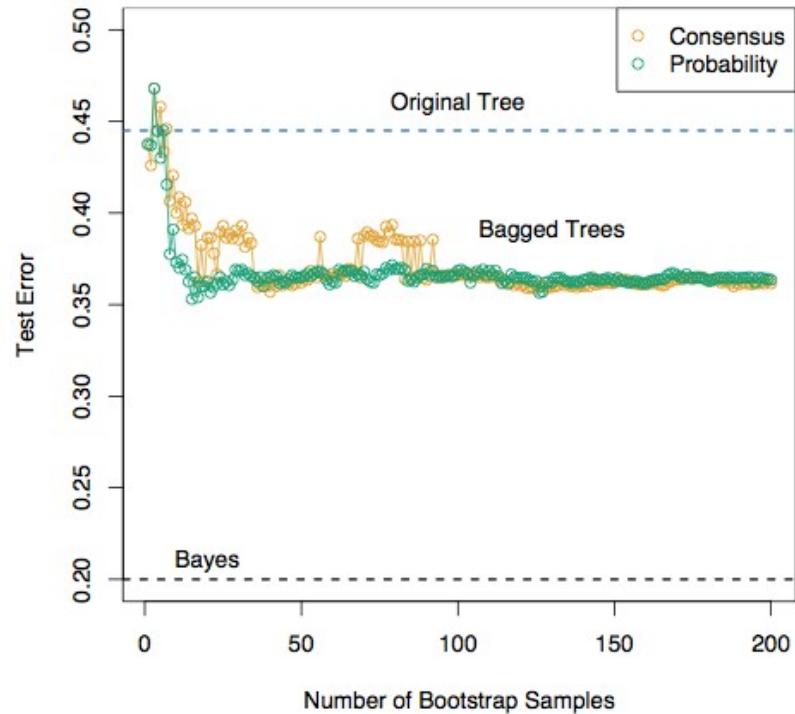


FIGURE 8.9. Bagging trees on simulated dataset. The top left panel shows the original tree. Eleven trees grown on bootstrap samples are shown. For each tree, the top split is annotated.

Bagging

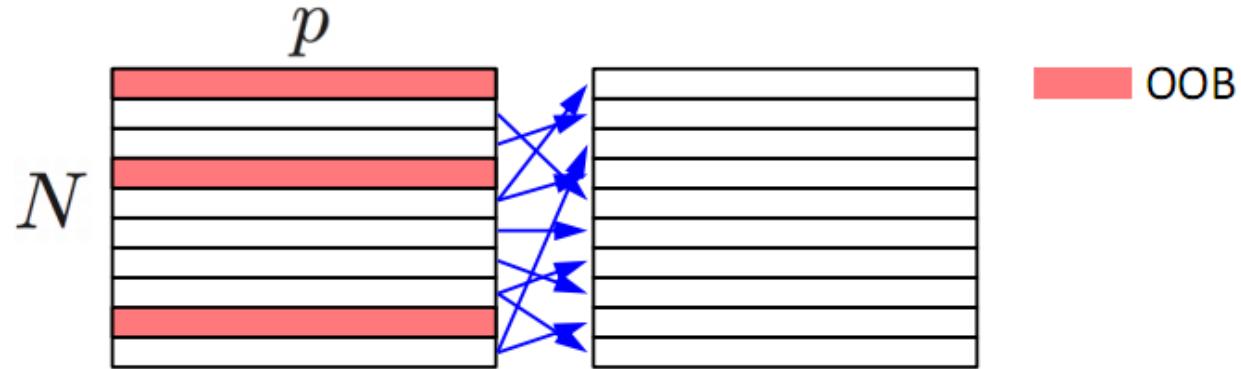
Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- Lấy mẫu có hoàn lại
- Xây dựng bộ phân lớp trên mỗi mẫu bootstrap
- Mỗi mẫu bootstrap chứa xấp xỉ 63.2% số lượng mẫu trong tập dữ liệu ban đầu
- Số lượng mẫu còn lại (36.8%) được dùng để kiểm thử

Bonus! Out-of-bag cross-validation

Các mẫu Out-of-bag (OOB)

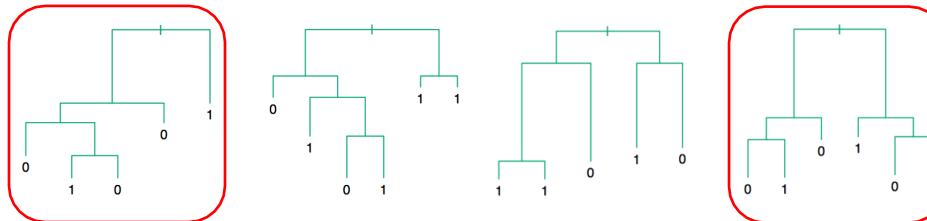
- Quá trình Bootstrapping:



- Mỗi cây chỉ sử dụng một tập con các mẫu huấn luyện (trung bình số mẫu $\sim 2/3$).
- Số mẫu cho OOB khoảng $\sim 1/3$ của cây quyết định.

Dự đoán mẫu OOB

- Với mỗi mẫu, tìm các cây mà nó là OOB.



- Dự đoán giá trị của chúng từ các cây này.
- Ước lượng lỗi dự đoán của cây (bagged trees) dùng tất cả các dự đoán OOB.
- Tương tự như kỹ thuật kiểm tra chéo (cross-validation).

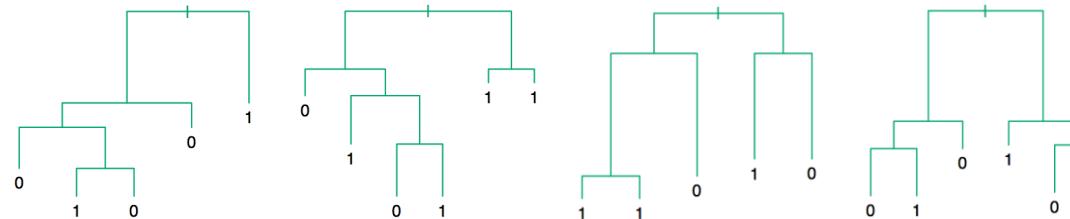
Phương pháp Rừng ngẫu nhiên Random Forests (RF)

Động lực để có Random forest

- Mô hình dựa trên cây phân loại và hồi quy (CART).
- Các mô hình cây có lỗi bias thấp, tuy nhiên phương sai lại cao (high variance).
- Phương pháp Bagging dùng để giảm phương sai.

Nhắc lại: Bagging

- Lấy mẫu tập dữ liệu huấn luyện theo Bootstrap để tạo ra tập hợp các dự đoán.



Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

- Lấy trung bình (hoặc bình chọn theo số đông-majority vote) các bộ dự đoán độc lập.
- Bagging giảm phương sai (variance) và giữ bias.

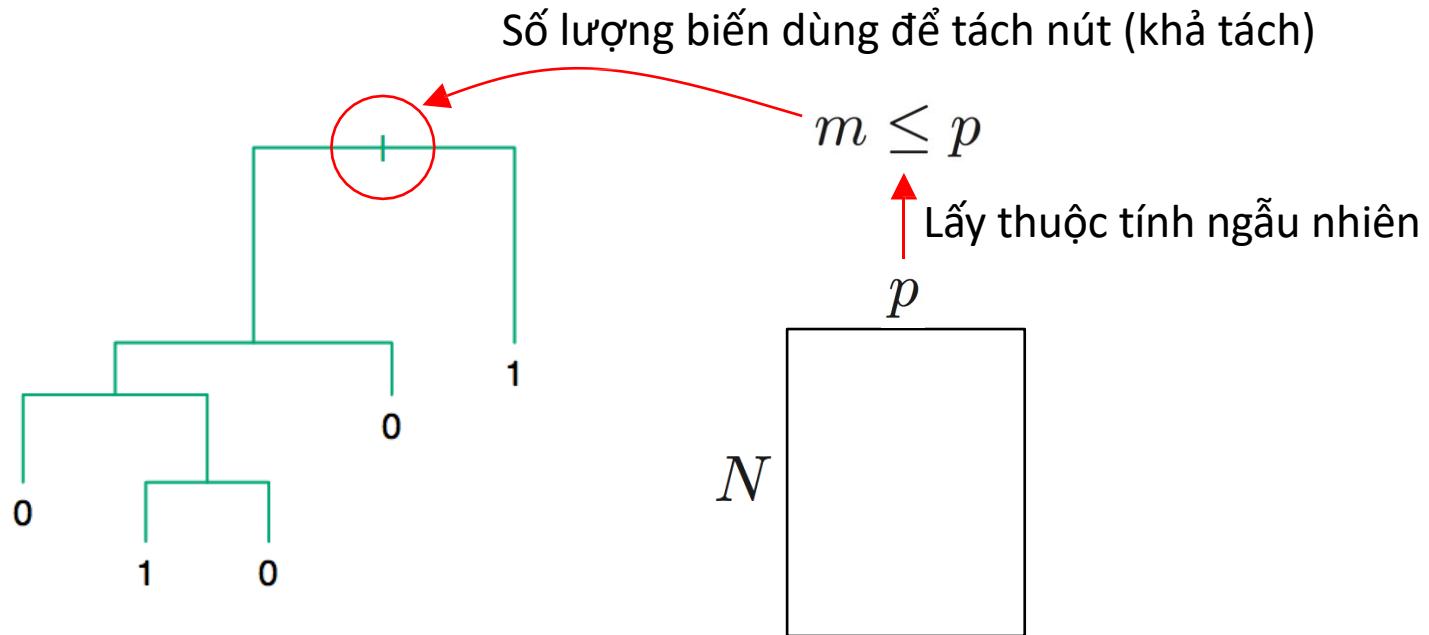
Bagged trees vs. random forests

- Phương pháp Bagging biểu thị sự biến thiên (variability) giữa các cây bởi việc chọn mẫu ngẫu nhiên từ dữ liệu huấn luyện.
- Cây được sinh ra từ phương pháp Bagging vẫn có tương quan lẫn nhau, do đó hạn chế trong việc giảm phương sai.

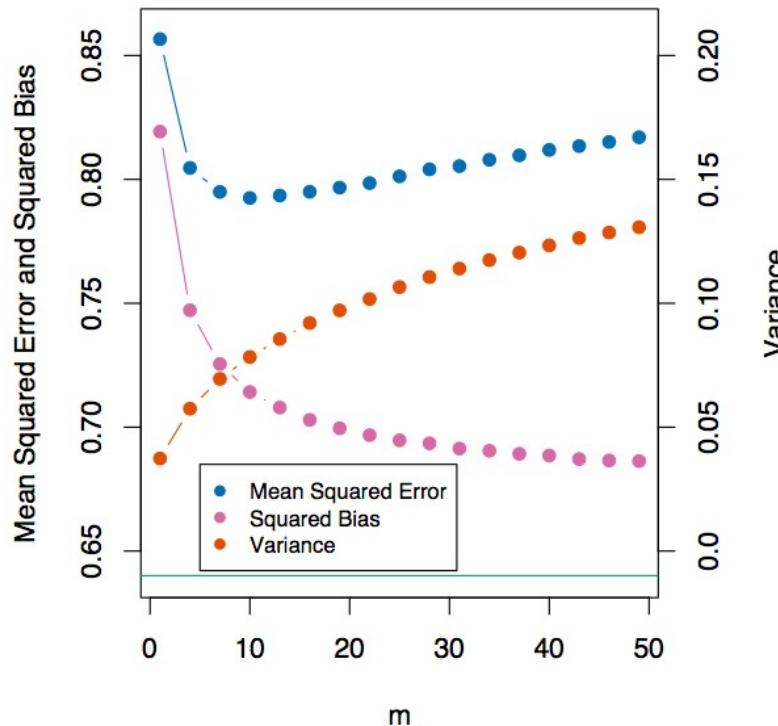
Random forests đưa ra thêm tính ngẫu nhiên (randomness):

- Làm giảm mối tương quan giữa các cây bằng cách lấy ngẫu nhiên các biến khi tách nút của cây.

Các biến dùng cho tách nút

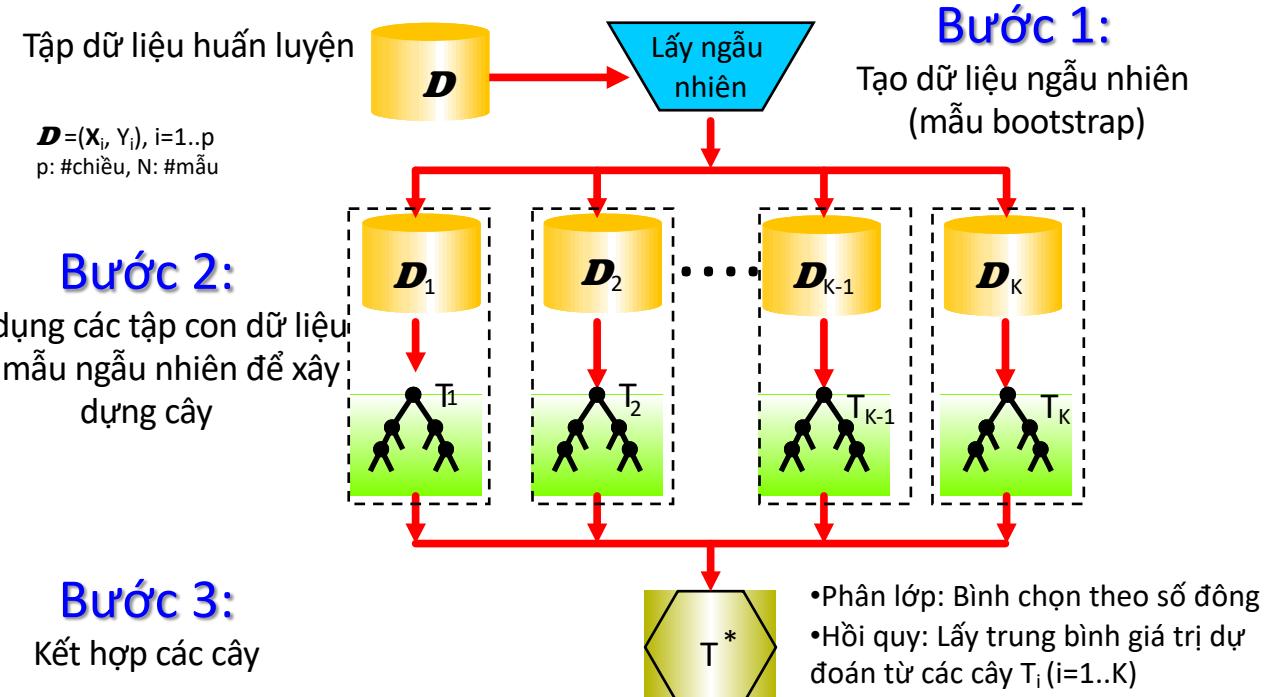


Các biến dùng cho tách nút



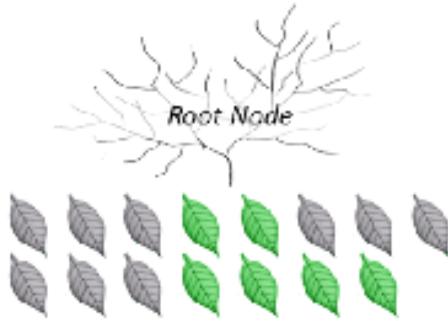
Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

Rừng ngẫu nhiên



Introduction to Data Mining – Tan, Steinbach, Kumar

Rừng ngẫu nhiên



All variables (Total: 9)

For more tutorials: algobeans.com



Các tham số chính

Các tham số quan trọng của Rừng ngẫu nhiên:

- Số lượng biến khả tách tại mỗi nút (m)
- Độ sâu của từng cây trong rừng (số lượng mẫu tối thiểu tại mỗi nút của cây-minimum node size)
- Số lượng cây trong rừng

Số lượng biến khả tách

Giá trị mặc định

Bài toán phân lớp

$$m = \lfloor \sqrt{p} \rfloor$$

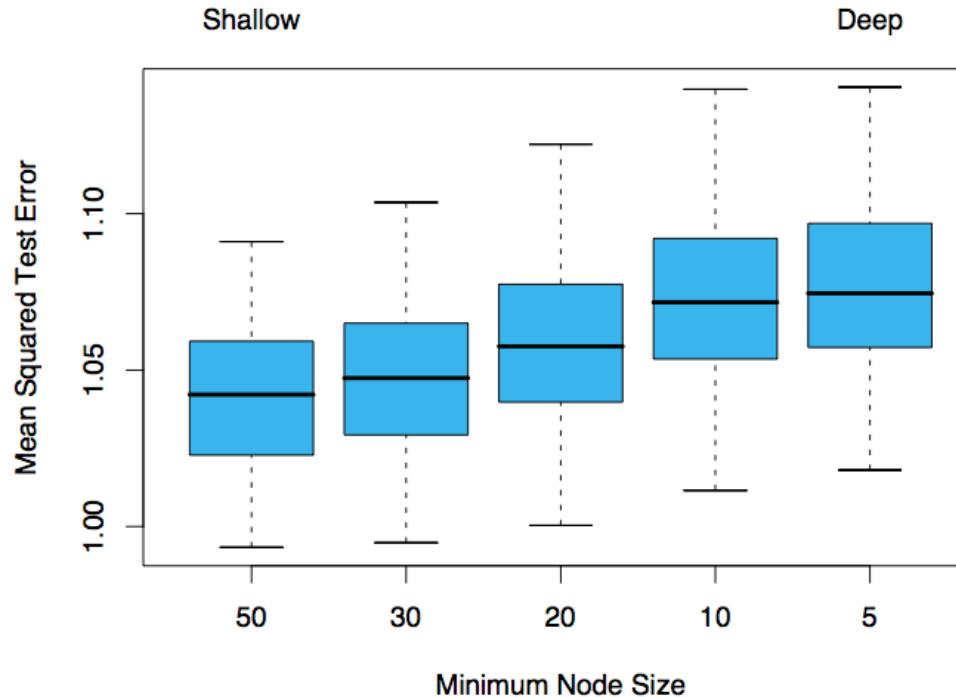
Bài toán hồi quy

$$m = \lfloor p/3 \rfloor$$

gói randomForest trong R dùng *mtry*

Độ sâu của từng cây

(số lượng mẫu tối thiểu tại mỗi nút của cây)



Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

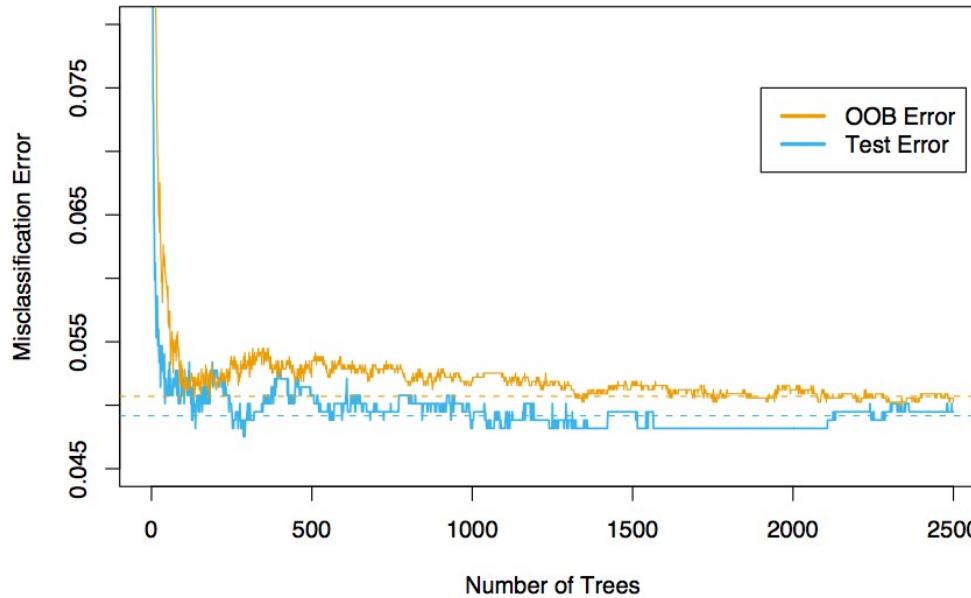
Độ sâu của cây

Giá trị mặc định

Bài toán phân lớp 1

Bài toán hồi quy 5

Số lượng cây trong rừng



Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

- Thêm nhiều cây không gây ra overfitting.

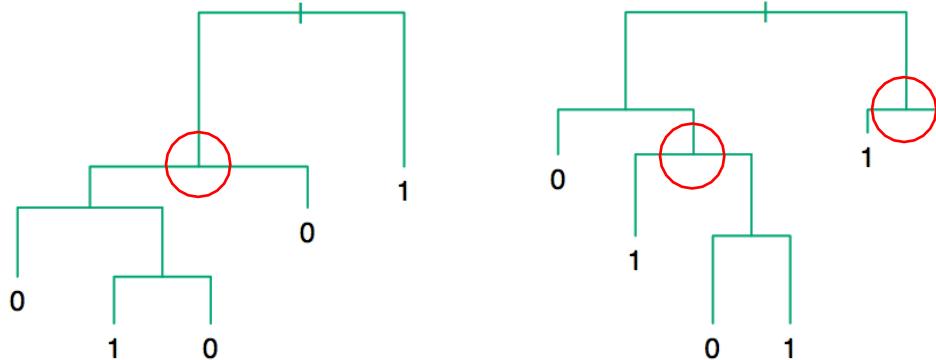
Các tính năng khác của RF

- Các mẫu Out-of-bag (OOB)
- Độ quan trọng của biến (Variable importance measurements)

Độ quan trọng của biến

Dạng 1:

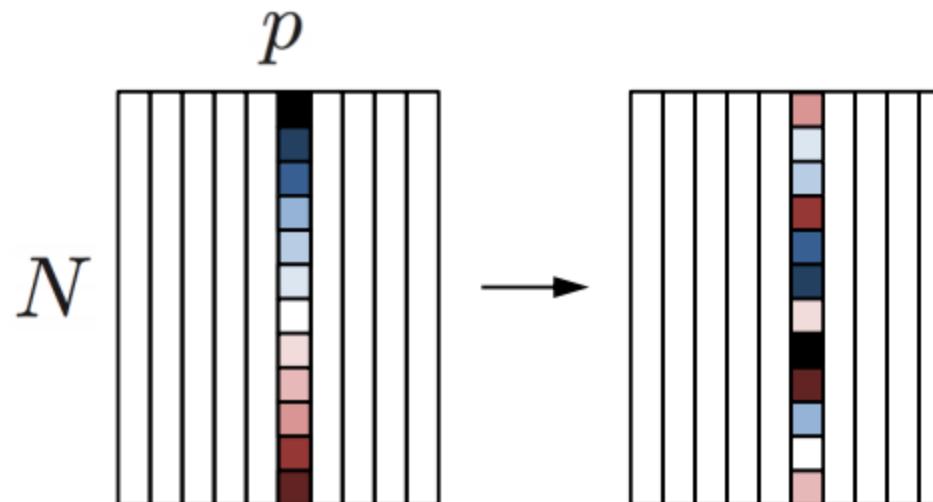
Độ giảm của lỗi dự đoán hoặc impurity từ các điểm tách nút liên quan đến các biến đó, cuối cùng lấy trung bình trên các cây trong rừng.



Độ quan trọng của biến

Dạng 2:

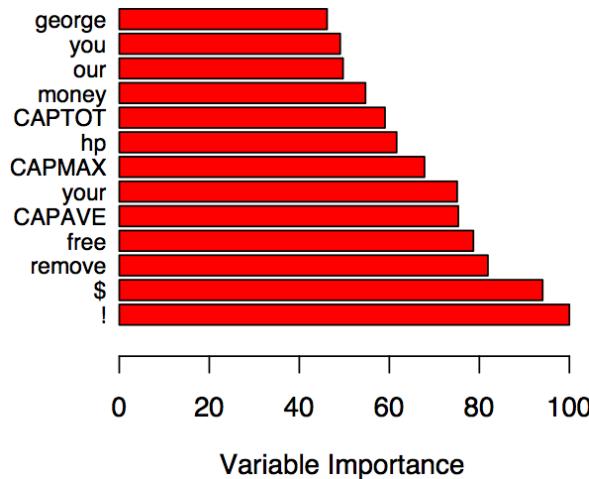
Độ tăng lõi dự đoán tổng thể khi các giá trị của biến được hoán vị ngẫu nhiên giữa các mẫu.



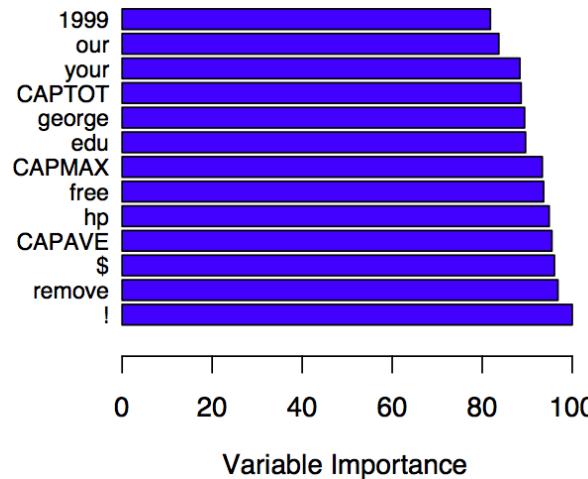
Ví dụ về độ quan trọng của biến

- Cả 2 dạng biểu thị gần giống nhau, tuy nhiên có sự khác biệt về xếp hạng các biến:

Dạng 1



Dạng 2



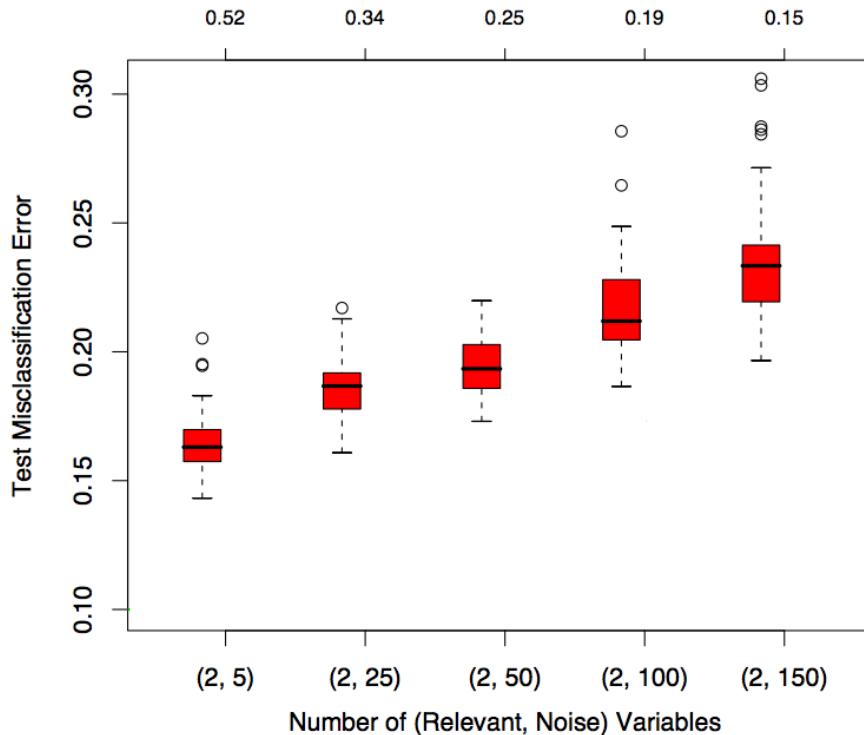
Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

Ưu điểm của RF

Tương tự như CART:

- Tương đối mạnh trong việc xử lý biến rác (non-informative variable)
(Việc lựa chọn biến tích hợp sẵn khi xây dựng mô hình, built-in variable selection)

Ảnh hưởng của biến rác



Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

Ưu điểm của RF

Tương tự như CART:

- Tương đối mạnh trong việc xử lý biến rác (non-informative variable)
- Xử lý (nắm bắt) được độ tương tác bậc cao giữa các biến (Capture high-order interactions between variables)
- Có lỗi bias thấp
- Dễ xử lý các biến hỗn hợp (biến rời rạc, phân loại)

Ưu điểm của RF

Ưu điểm vượt trội CART:

- Lỗi phương sai thấp hơn (mạnh hơn vì sử dụng phương pháp bootstrapping lấy mẫu từ tập huấn luyện)
- Ít bị overfitting hơn
- Không cần tỉa cây (No need for pruning)
- Kiểm tra chéo được tích hợp sẵn trong mô hình (dùng các mẫu OOB)



Nhược điểm của RF

Tương tự như CART:

- Khó nắm bắt độ cộng tính

Nhược điểm so với CART:

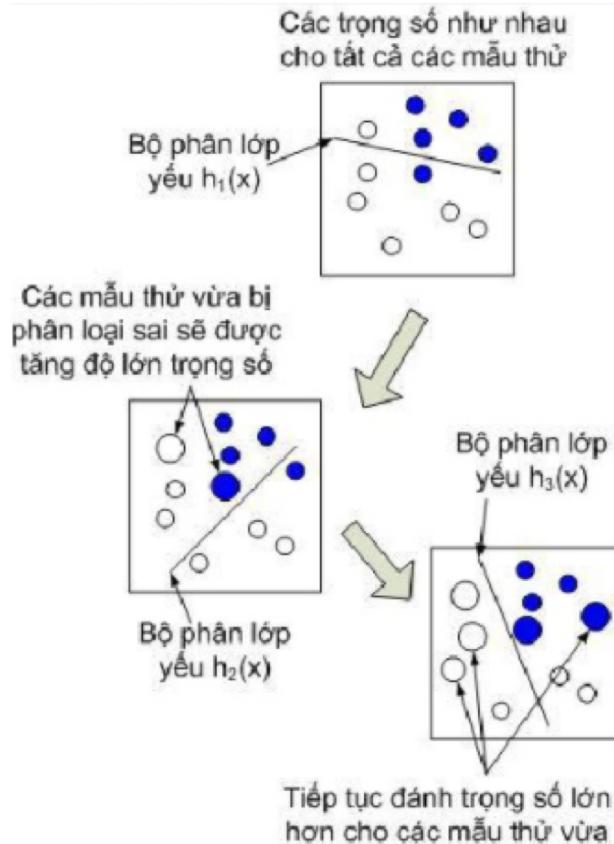
- Khó diễn giải/giải thích mô hình dự đoán

Boosting

- Xây dựng một lượng lớn các mô hình (thường là cùng loại).
- Mỗi mô hình sau sẽ học cách sửa những lỗi của mô hình trước (dữ liệu mà mô hình trước dự đoán sai)
- Tạo thành một chuỗi các mô hình mà mô hình sau sẽ tốt hơn mô hình trước bởi trọng số được update qua mỗi mô hình:
 - Trọng số của những dữ liệu dự đoán đúng sẽ không đổi,
 - Trọng số của những dữ liệu dự đoán sai sẽ được tăng thêm
- Chúng ta sẽ lấy kết quả của mô hình cuối cùng trong chuỗi mô hình này làm kết quả trả về (vì mô hình sau sẽ tốt hơn mô hình trước nên tương tự kết quả sau cũng sẽ tốt hơn kết quả trước)



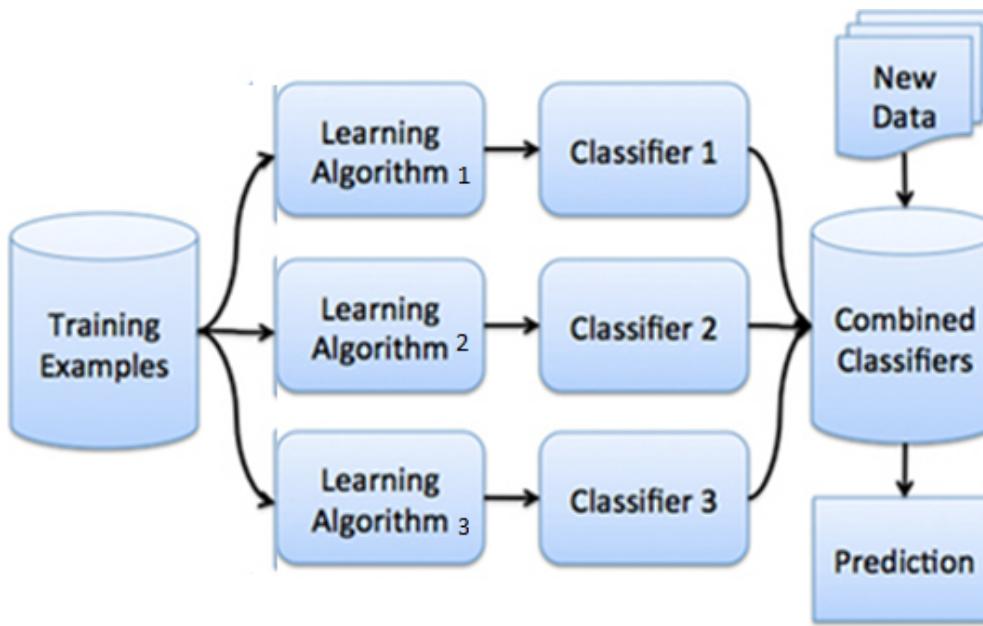
Boosting



Stacking

- Xây dựng một số mô hình (thường là khác loại) và một meta model (supervisor model),
- Train những mô hình này độc lập,
- Sau đó meta model sẽ học cách kết hợp kết quả dự báo của một số mô hình một cách tốt nhất.

Mô hình học kết hợp



Câu hỏi?



Trees, RF: iris data

Dữ liệu về hoa iris cung cấp đo lường liên quan đến chiều dài (sepal length, petal length), bề rộng (width)

- của 50 loại hoa
- từ 3 giống (setosa, versicolor, virginica)
 - Mục tiêu: dùng đo lường để phân biệt các loài hoa

