Bagging and Random Forest

Taehoon Ko (taehoonko@snu.ac.kr)

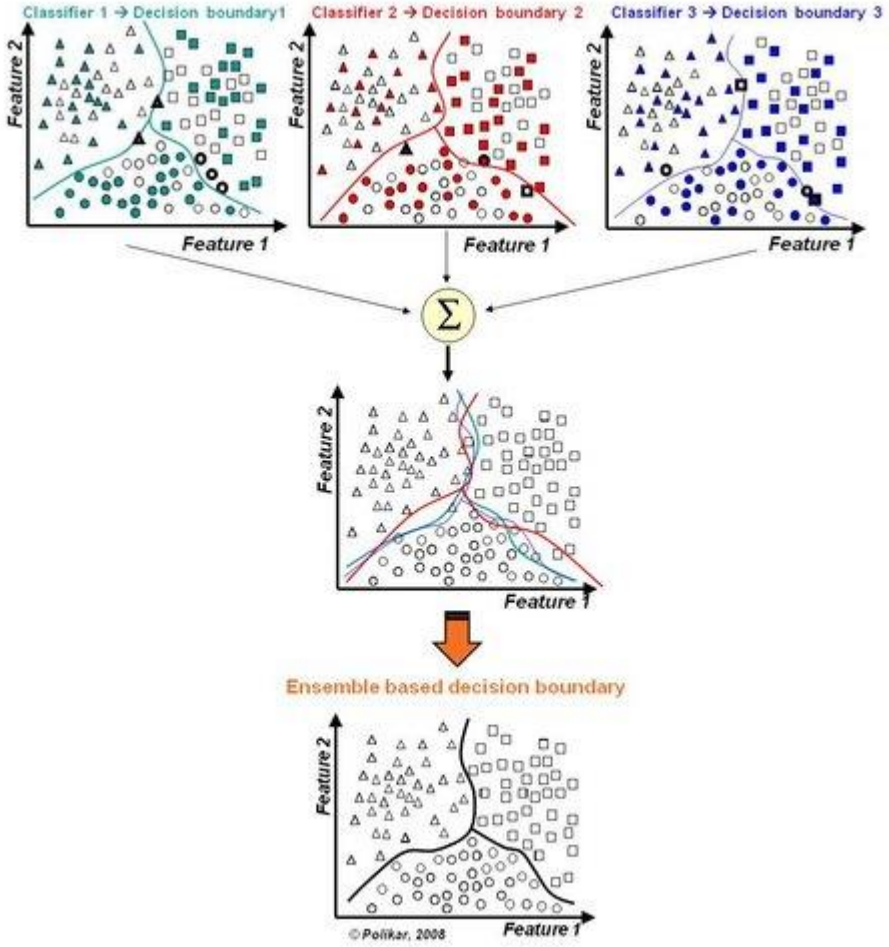
목표

- 앙상블 (Ensemble) 의 개념을 이해한다.
- 앙상블 방법 중 하나인 배깅 (Bagging: Bootstrap aggregating) 을 이해한다.
- 랜덤 포레스트 (Random forest) 는 의사결정나무 모델을 단순 배깅보다 더 고도화된 방법으로 앙상블하는 것이다. 이에 대해 이해한다.

앙상블과 배깅

앙상블 (Ensemble)

머신러닝에서 앙상블이란 단일 모델이 아닌 여러 모델을 혼합하여 의사결정을 내리는 방법



https://music.ubc.ca/student-ensembles/
http://magizbox.com/training/machinelearning/site/ensemble/

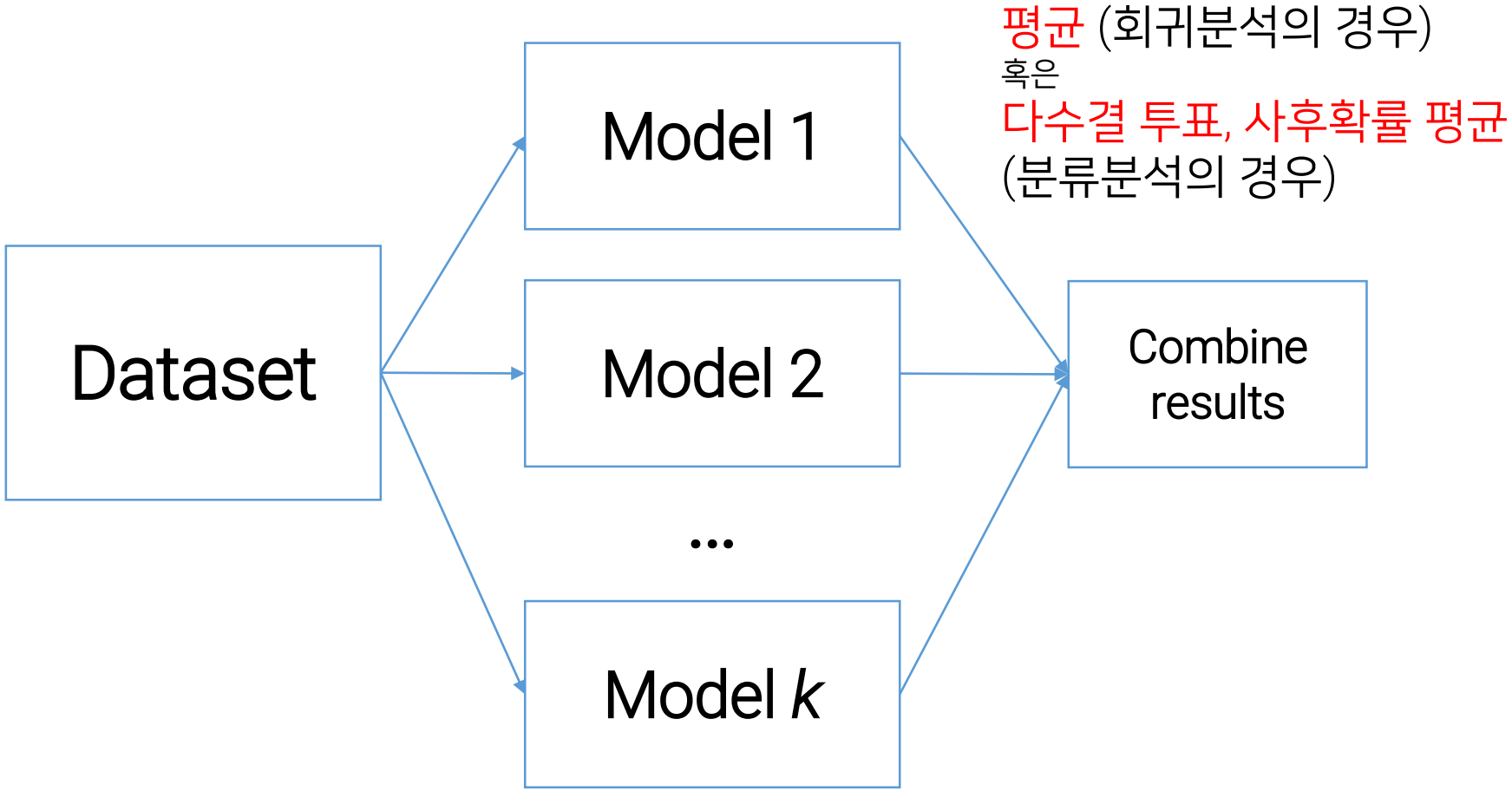
앙상블 (Ensemble)

여러 가지 유형의 앙상블 방법이 존재함

- 단순/가중 평균 (Simple/weighted average)
- 배깅 (Bagging: Bootstrap aggregating)
- 부스팅 (Boosting)
- 스택킹 (Stacking)
- 메타 학습 (Meta-learning)
- ...

단순 / 가중 평균

하나의 데이터셋에 여러 모델을 학습한 후, 각 모델들 결과의 평균 (혹은 다수결 투표) 으로 최종 산출



Averaging can reduce ‘variance’

- **원리: 표본평균의 분산**
  - 모집단에서 N만큼 표본을 추출하고, 그 표본의 평균을 ‘표본평균’이라 하자.
    이 때, 표본평균의 평균은 모평균과 동일하며, 표본평균의 분산은 모분산 / N

$$\mathrm{Var}(\bar{X}) = \frac{\mathrm{Var}(X)}{N}$$

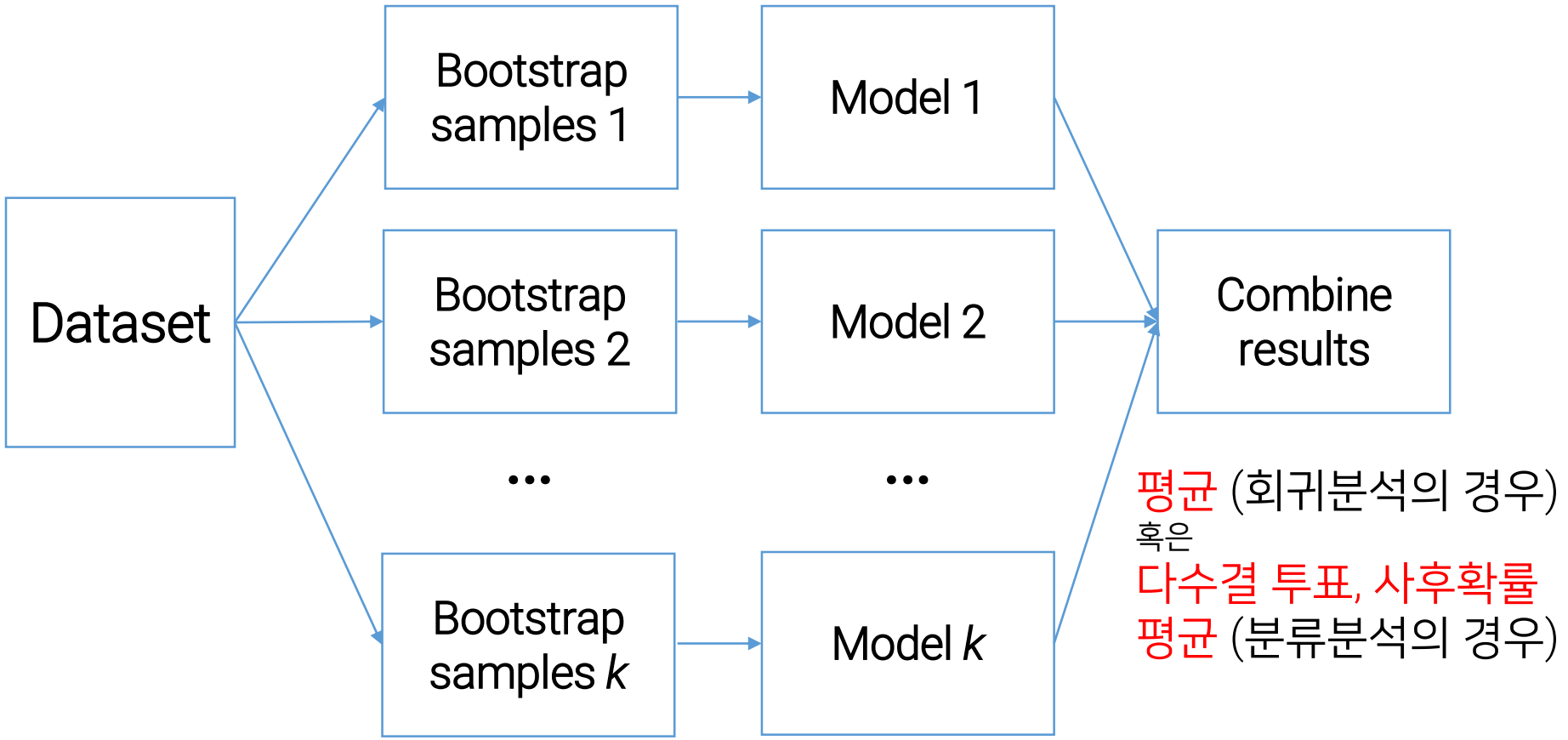
  - X가 각 개별 모델이라고 가정하고, $\bar{X}$를 앙상블된 모델이라고 하자.
    - 앙상블된 모델의 분산이 각 개별 모델의 분산보다 낮을 것이라 기대
  - **문제점**: 각 개별 모델을 만드는 학습용 데이터셋을 동일하게 가져가면,
    개별 모델이 비슷한 결과를 내뱉게 됨으로써 앙상블의 효과가 떨어질 것이다.
    - e.g., 자문위원회 구성 시, 자문위원을 다양한 분야에서 뽑는 것이 좋은가,
      하나의 분야에서 여럿을 뽑는 것이 좋을까?

Bagging (Bootstrap Aggregating)

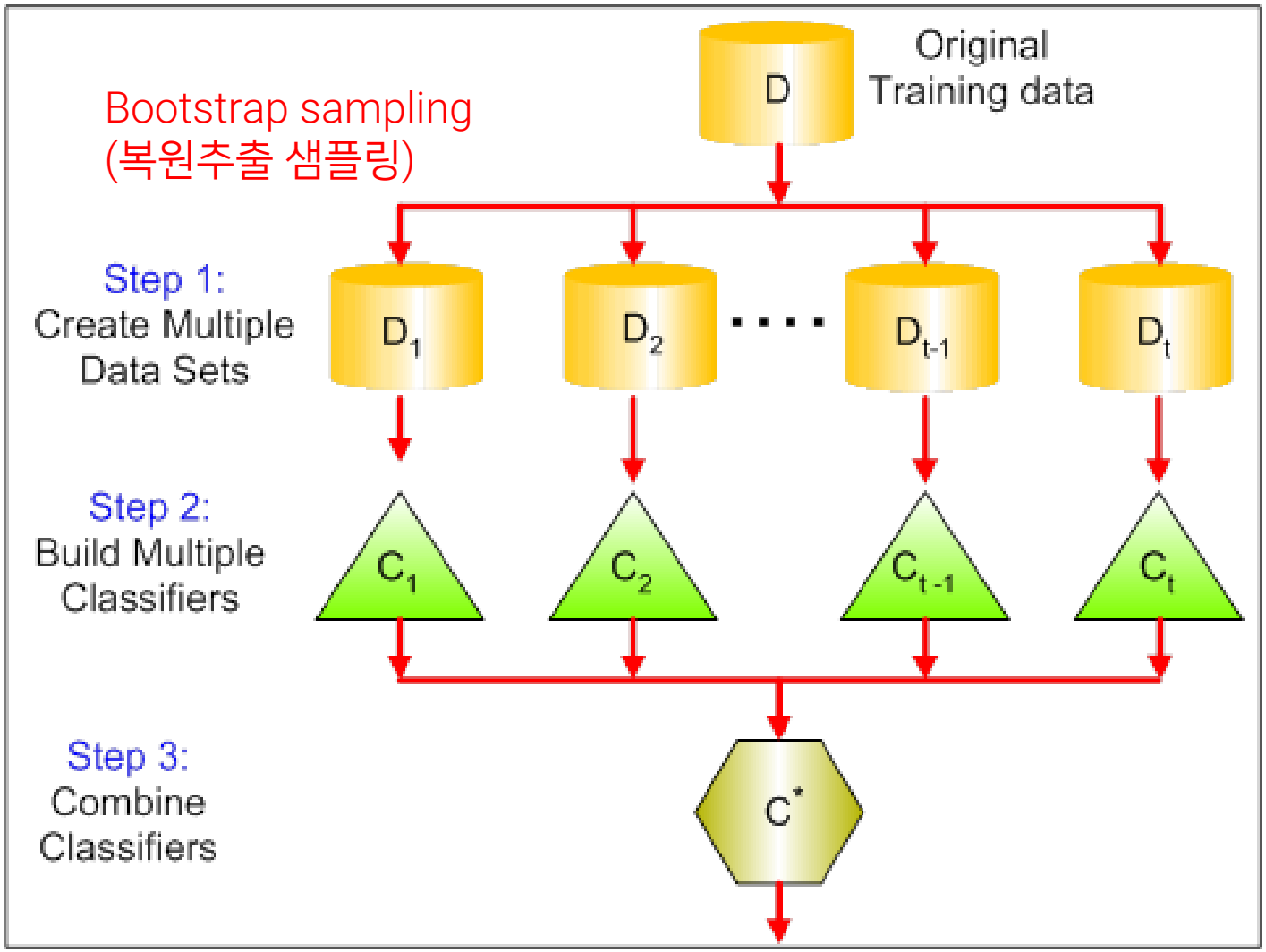
- Solution
  - Take repeated bootstrap samples from training set D.
  - *Bootstrap sampling (복원추출)* : Given set D containing *N* training points, create D' by drawing *n* (*N* > *n*) points at random with replacement from D

- Bagging
  - proposed by Leo Breiman (1994), who proposed CART algorithm
  - Create *k* bootstrap samples $D_1, D_2, \ldots, D_k$.
  - Train distinct classifier on each D_i.
  - Classify new instance by majority vote or average.

Bagging


Dataset
Bootstrap samples 1
Bootstrap samples 2
Bootstrap samples *k*
Model 1
Model 2
Model *k*
Combine results
평균 (회귀분석의 경우)
혹은
다수결 투표, 사후확률 평균 (분류분석의 경우)

Bagging



https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

Bagging

Single model

Bagging

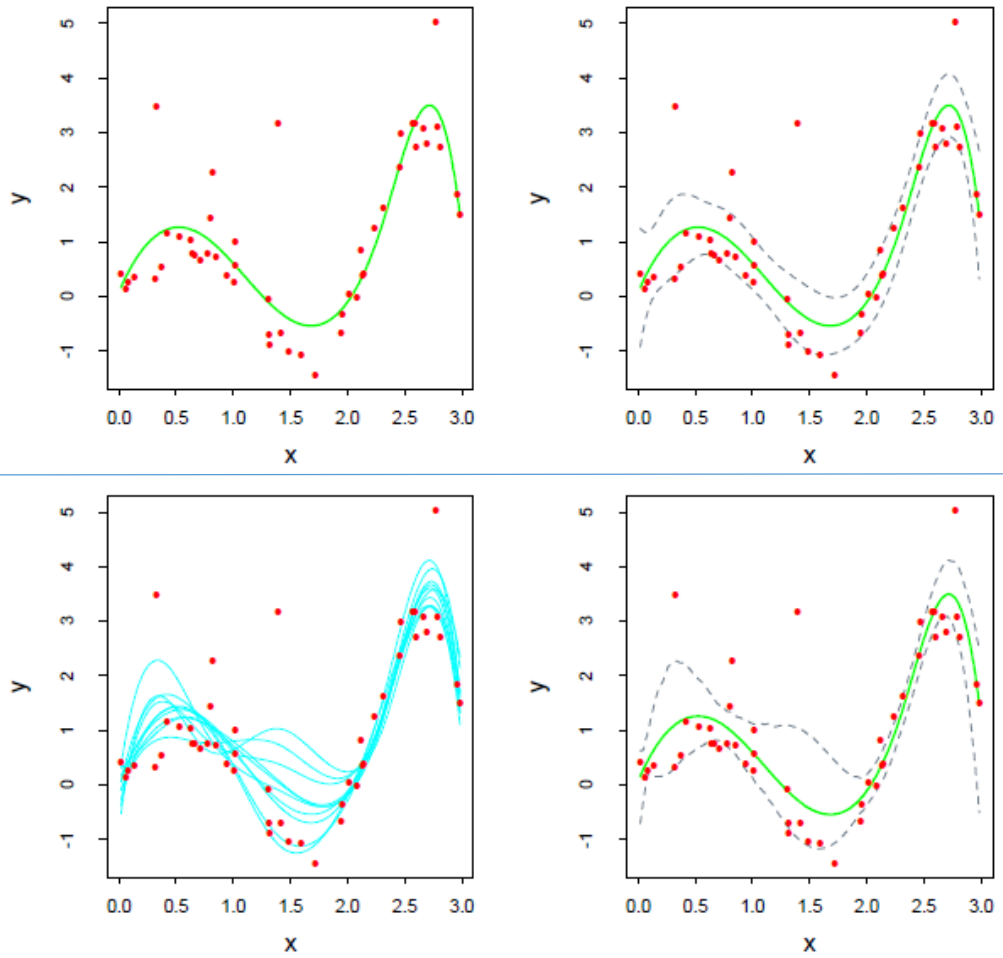


FIGURE 8.2. *(Top left:) B-spline smooth of data. (Top right:) B-spline smooth plus and minus* $1.96\times$ *standard error bands. (Bottom left:) Ten bootstrap replicates of the B-spline smooth. (Bottom right:) B-spline smooth with* 95% *standard error bands computed from the bootstrap distribution.*

Bagging

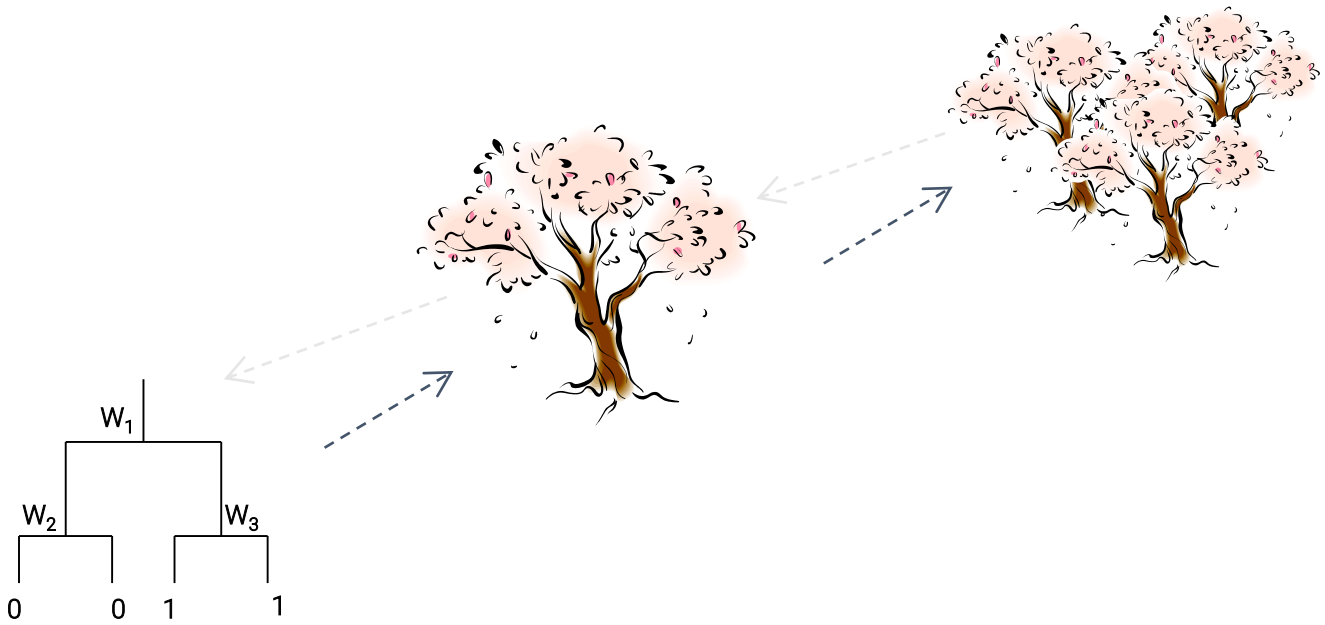
- The bagging method is similar to expert system.
  - 하나의 의안을 해결하기 위해 여러 분야의 전문가에게 도움을 구함

- What algorithms can be used?
  - (이론상으로는) 모든 알고리즘이 사용 가능함
  - decision tree가 자연스럽게 많이 이용되기 시작
    - Full-grown tree는 bias를 매우 낮고, variance가 매우 높기 때문에 여러 개의 full-grown tree를 이용하여 variance를 낮추자.
    - 여러 의사결정나무 모델의 관련성을 더 떨어뜨리는 방법은?

랜덤 포레스트

Random forest

- Developed by Leo Breiman(father of CART and bagging) at University of California, Berkeley (1996, 1999)
- Special case of the “bagging”
- Attempt to reduce bias of single tree



Journal of Machine Learning Research 15 (2014) 3133-3181 Submitted 11/13; Revised 4/14; Published 10/14

Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?

Manuel Fernández-Delgado MANUEL.FERNANDEZ.DELGADO@USC.ES
Eva Cernadas EVA.CERNADAS@USC.ES
Senén Barro SENEN.BARRO@USC.ES
CITIUS: Centro de Investigación en Tecnoloxías da Información da USC
University of Santiago de Compostela
Campus Vida, 15872, Santiago de Compostela, Spain

Dinani Amorim DINANIAMORIM@GMAIL.COM
Departamento de Tecnologia e Ciências Sociais- DTCS
Universidade do Estado da Bahia
Av. Edgard Chastinet S/N - São Geraldo - Juazeiro-BA, CEP: 48.305-680, Brasil

Editor: Russ Greiner

Abstract

We evaluate **179 classifiers** arising from **17 families** (discriminant analysis, Bayesian, neural networks, support vector machines, decision trees, rule-based classifiers, boosting, bagging, stacking, random forests and other ensembles, generalized linear models, nearest-neighbors, partial least squares and principal component regression, logistic and multinomial regression, multiple adaptive regression splines and other methods), implemented in Weka, R (with and without the caret package), C and Matlab, including all the relevant classifiers available today. We use **121 data sets**, which represent **the whole UCI** data base (excluding the large-scale problems) and other own real problems, in order to achieve significant conclusions about the classifier behavior, not dependent on the data set collection. **The classifiers most likely to be the bests are the random forest** (RF) versions, the best of which (implemented in R and accessed via caret) achieves 94.1% of the maximum accuracy overcoming 90% in the 84.3% of the data sets. However, the dif-

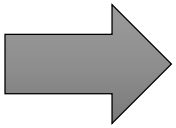
Random forest: Motivation

- Decision tree (especially CART algorithm)
  - Advantages
    - Extracting decision rules (If A, then B)
    - Selecting important predictors automatically
  - Limitation
    - not great for nonlinear decision boundary

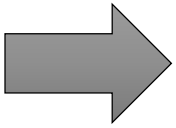
- Bagging trees can be a solution to overcome aforementioned limitation.
  - However, in the bagging structure, same classifiers may be correlated highly so that generalization performance of bagging is degraded.

Random forest: 2 randomizations

- To maintain some advantage(Selecting important predictors) of a single tree model while reducing bias, 2 randomizations are applied to the RF model.
  - 1st randomization: Bagging
  - 2nd randomization: Predictor subsets chosen randomly
    - Sub-classifiers in bagging structure are trained by same algorithm, CART. However, by this randomization, each sub-classifier's training set has different characteristics from others.



Bootstrapping training set + Bootstrapping features



Trees will become more uncorrelated when using only bagging.

Random forest: 2 randomizations

$$X = \begin{pmatrix} 1 & 5 & 0 & -1 & 2 \\ 0 & 3 & 9 & 1 & -3 \\ 2 & 8 & 9 & 0 & 3 \\ 3 & -1 & 0 & -2 & 3 \end{pmatrix} \qquad Y = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

row: {1, 3, 4}
column: {1, 3, 5}

row: {1, 2, 3}
column: {2, 4, 5}

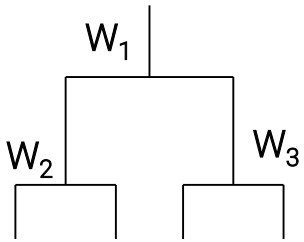
$$X^{(1)} = \begin{pmatrix} 1 & 0 & 2 \\ 2 & 9 & 3 \\ 3 & 0 & 3 \end{pmatrix} \qquad Y^{(1)} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

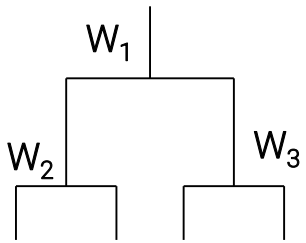
$$X^{(2)} = \begin{pmatrix} 5 & -1 & 2 \\ 3 & 1 & -3 \\ 8 & 0 & 3 \end{pmatrix} \qquad Y^{(2)} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

...

train a single tree

train a single tree





...

Random forest: 2 randomizations

- Randomization through "predictor subsets"
  - If each single tree in forest uses all predictors, it is just a "simple" bagging method
  - Random Forests algorithm chooses predictor subsets randomly, and constructs a single tree by training each predictor subset.
  - The number of predictors of each tree = m_{try}
    - Generally, in a classification problem, $m_{try} = \sqrt{p}$ (*p*: number of all predictors)
    - Generally, in a regression problem, $m_{try} = \mathrm{p}/3$

[Example] Predictor set = {X_1, X_2, X_3, …, X12} & regression problem

➔ predictor subset when training tree 1 = {X1, X4, X5, X9}

➔ predictor subset when training tree 2 = {X1, X2, X10, X11}

➔ predictor subset when training tree 3 = {X7, X8, X10, X12}

…

➔ predictor subset when training tree *n* = {X1, X2, X6, X12}

Random forest: Training

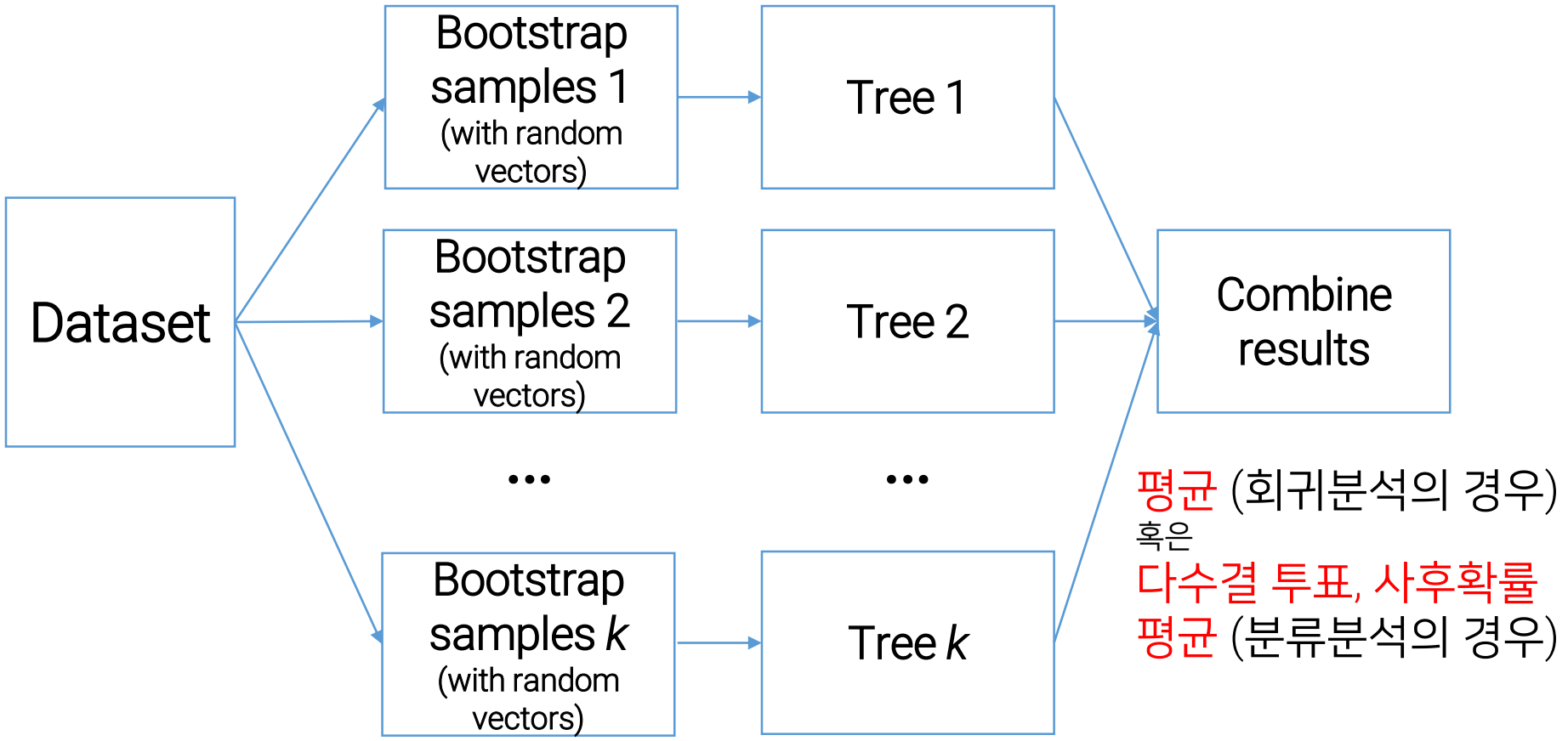
- By 2 randomization, construct a bootstrap sample from training set.
- Train a tree using above bootstrap sample.
  - Each tree is fully-grown until the impurity of each terminal node is nearly zero.
  - As RF combines full-grown tree, we can expect that RF does not overfit. (Breiman, 2001)
  - Key parameters: **number of predictors** in each tree, **number of trees**

Random forest: Inference

Same inference to bagging method

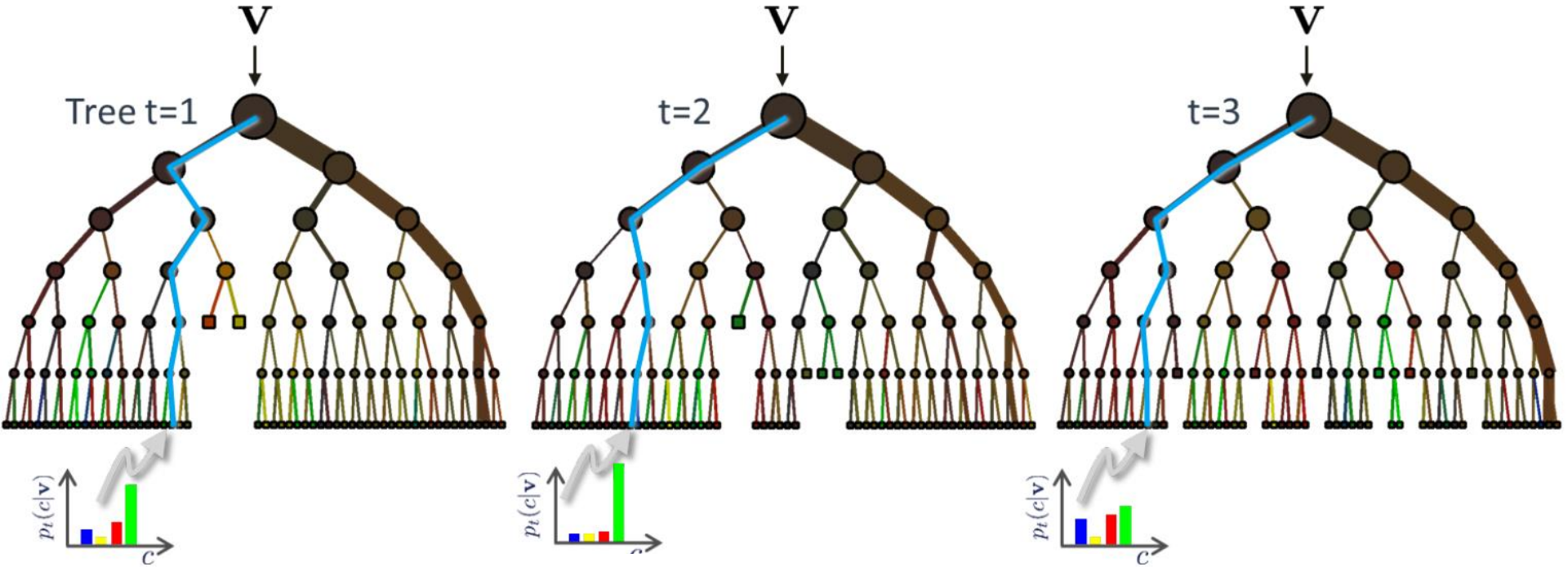
- For classification
  - Majority vote: 각 single tree의 classification 결과를 보고 다수가 예측한 클래스로 최종 예측
  - Probability prediction: 전체 single tree에서 각 클래스로 예측한 tree의 비율을 이용. Cut-off를 이용하여 분류할 때 사용
- For regression
  - Average: 각 single tree가 예측한 값의 평균으로 최종 예측

Random forest



Random forest

- Random forest: Testing
  - A new data point $\mathbf{v}$ is pushed through each component tree.
  - The forest class posterior $p(\mathbf{c}|\mathbf{v})$ is simply the average of all tree



$$p(c|\mathbf{v}) = \frac{1}{T}\sum_{t}^{T} p_t(c|\mathbf{v})$$

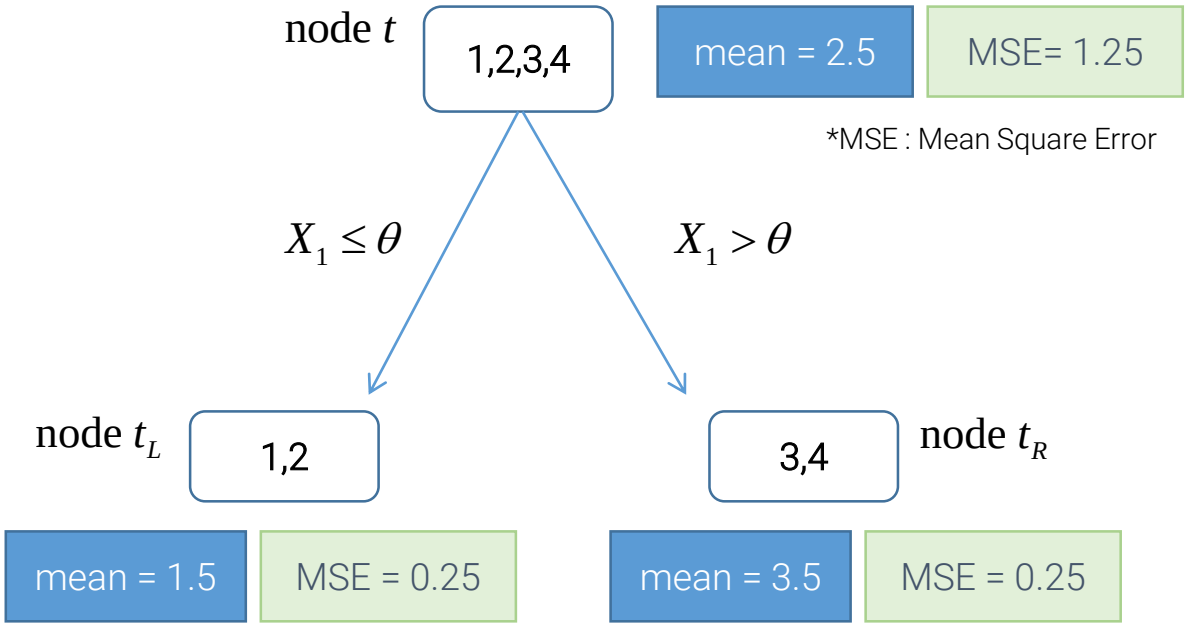
(Criminisi et al., 2011)

Random forest: Feature importance

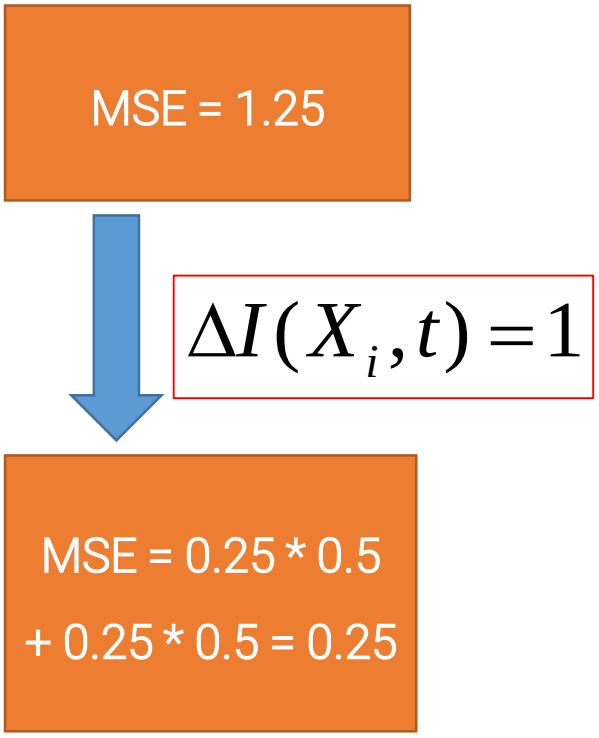
- Feature importance in random forest
  - Each single tree selects important predictors automatically.
  - Random forest can evaluate each predictor by combining all single tree's opinion.
  - 2 ways
    - Mean decrease impurity
    - Mean decrease accuracy

Random forest: Feature importance

- Mean decrease impurity
  - In a single tree,

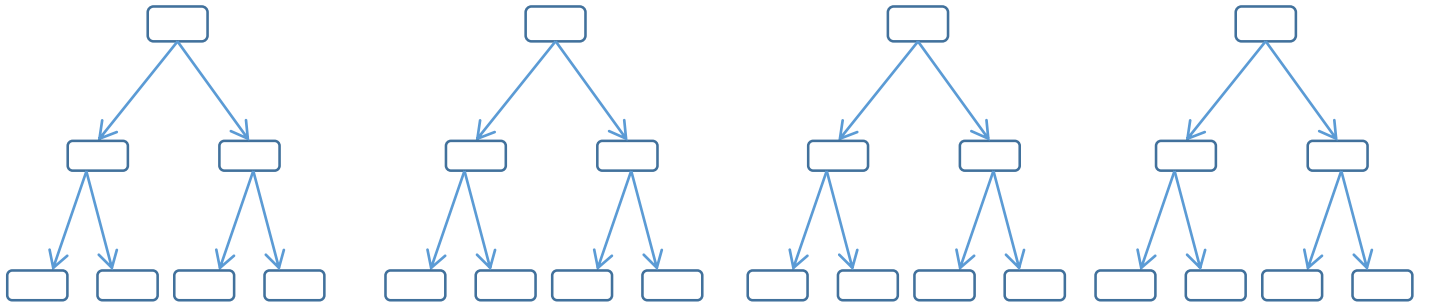


$$\Delta I(X_i, t) = I(t) - p_L I(t_L) - p_R I(t_R)$$



Random forest: Feature importance

- Mean decrease impurity
  - Consider a forest of 4 trees.



$\Delta I(X_1, T_1) = 1$ $\Delta I(X_1, T_2) = 1.2$ $\Delta I(X_1, T_3) = 3.2$ $\Delta I(X_1, T_4) = 0.75$

$$\text{Importance}(X_1) = \frac{1}{M}\sum_{m=1}^{M}\Delta I(X_1, T_m) = \frac{1+1.2+3.2+0.75}{4} = 1.5375$$

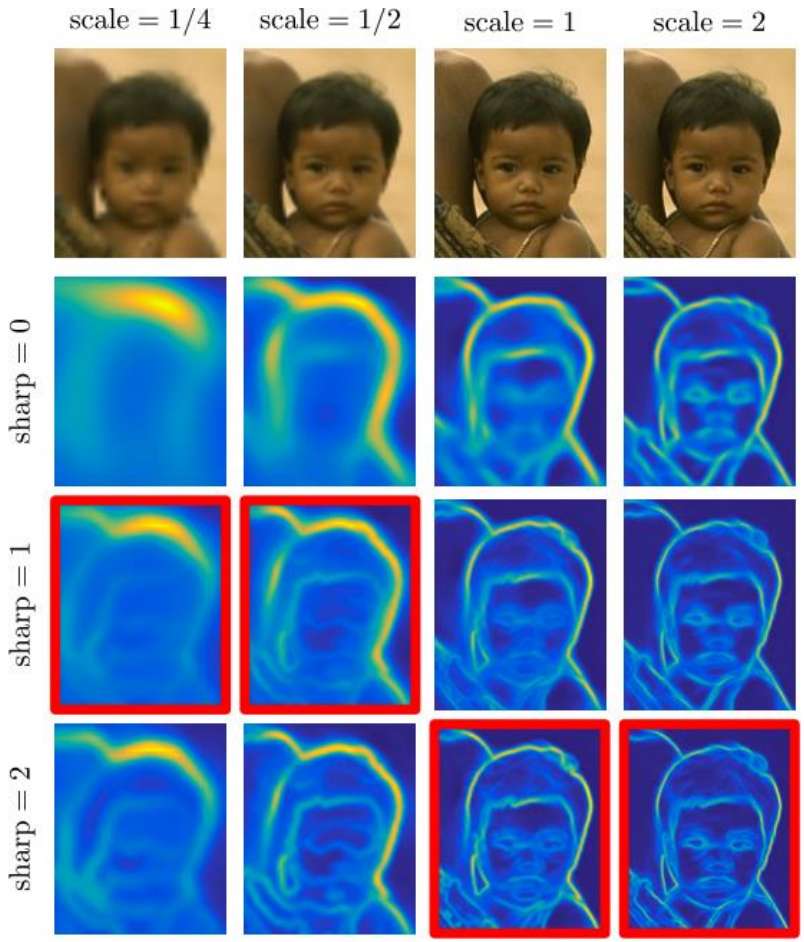
Random forest: Feature importance

- Mean decrease accuracy
  - 1. Train a random forest.
  - 2. Calculate test error ($test\_error\_base$).
  - 3. Permute variable X_i.
    - Rearrange the values of X_i or change them randomly.
  - 4. Re-calculate test error ($test\_error\_changed$).
  - 5. Importance of $X_i = test\_error\_changed - test\_error\_base$

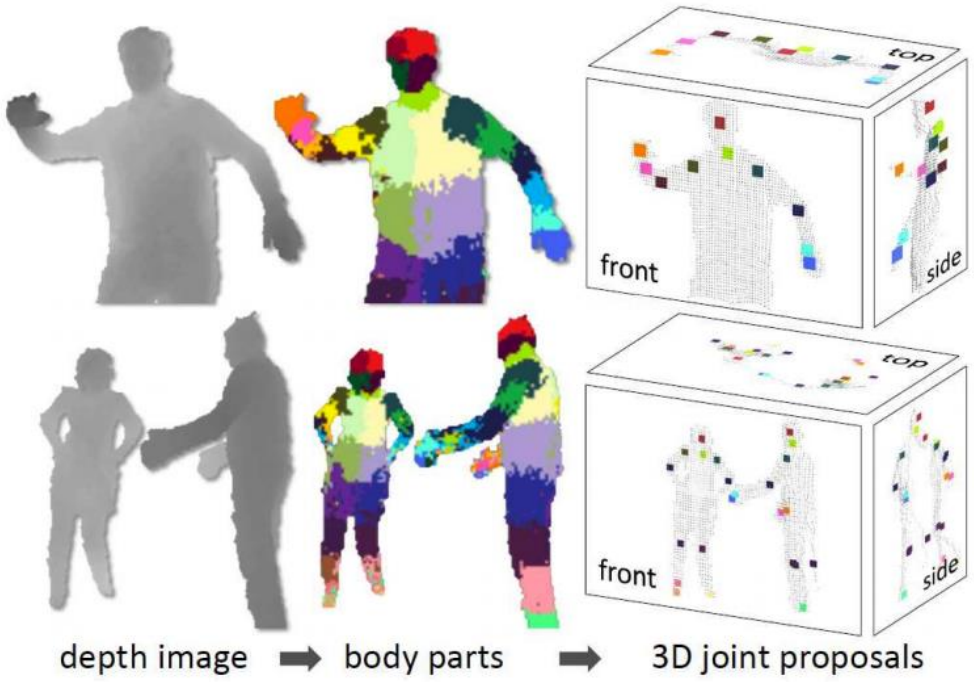
If the variable is not important, then permuting that variable will not degrade prediction accuracy.

Random forest: Applications

- Random forest is widely used for various purpose in the field of computer vision. ➔ Object / edge detection, Object tracking, Image classification, etc.



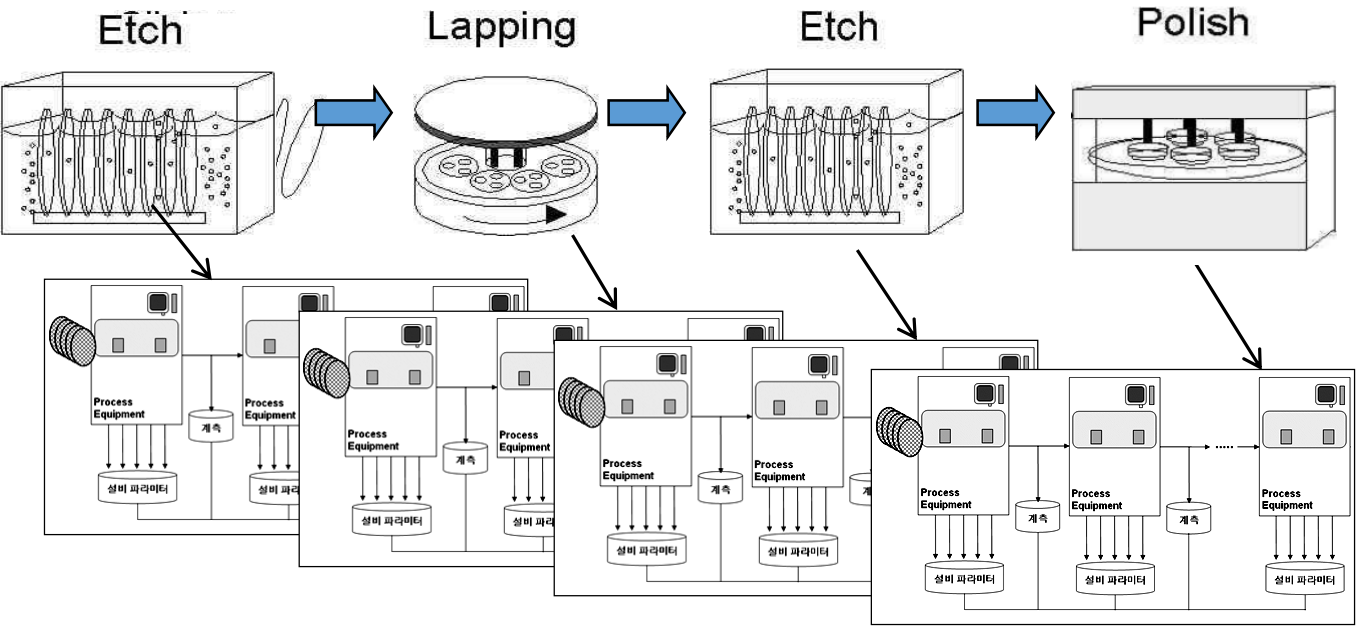
(Hallman & Fowlkes, 2015)

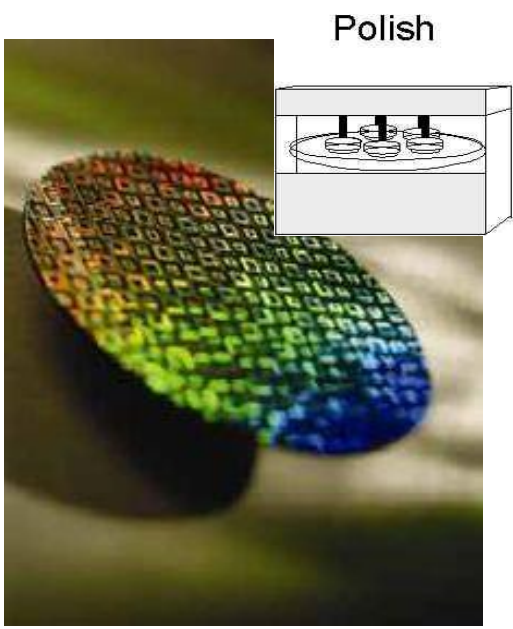


(Newcombe et al., 2011)

Random forest: Applications

- Semiconductor process and wafer yield
  - The semiconductor process consists of hundreds of processes.
  - Yield: Measured by the number of semiconductors operating normally among the semiconductor in a wafer
  - It is very important to predict the yield and to see the influence of the process sensor on it.

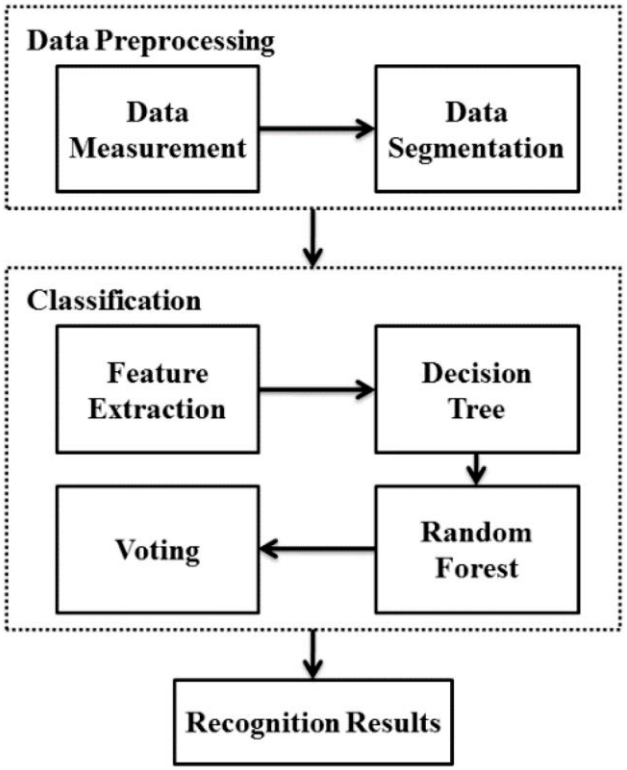




(Ko et al., 2010)

Random forest: Applications

- Random forest shows good prediction performance even for fat data with a large number of predictors.



Random forest can extract importance values of all predictors.

Period	Y	Best Model	RMSE	R^2
2	1	GA-LR	0.1646	0.9820
	2	GA-LR	0.5073	0.9175
	3	GA-LR	0.1616	0.9570
5	1	GA-LR	0.3250	0.6570
	2	GA-LR	0.3323	0.7994
	3	GA-LR	0.3253	0.7780
6	1	GA-LR	0.5406	0.8780
	2	GA-LR	1.0010	0.6288
	3	GA-LR	0.6110	0.7829
8	1	Stepwise-LR	1.6608	0.8849
	2	GA-LR	1.1897	0.9685
	3	GA-LR	1.1967	0.9113
All	1	Random Forests	1.4944	0.7448
	2	Random Forests	2.1527	0.7516
	3	Random Forests	1.4753	0.7540

(Ko et al., 2010)

장점 및 단점

- **장점**
  - 대부분의 데이터에서 모델 성능이 좋음
  - 변수가 굉장히 많은 데이터 학습에 용이
- **단점**
  - Bagging이 bias를 줄이기는 어려움
  - Overfitting 이슈
    - 2 randomizations가 정말 충분히 각 나무를 uncorrelated하게 만들까?