Data Mining

Combining Learners

2003.10.06

최 대 우

한국외국어대학교 정보통계학과

Agenda

- Methods
- Bagging
- Arcing
- Bumping
- etc.

Combining learners to improve prediction

- Stacking (David Wolpert)
- Bagging (Leo Breiman)
- Arcing (Yoav Freund and Robert Shapire)
- Bumping (Robert Tibshirani)

Bagging

- Bagging (Bootstrap AGGregatING) averages a given procedure over many samples, to reduce its variance.
- Suppose $C(\mathbf{x},t)$ is a classifier, producing a K-vector output with 1 one and $K-1$ zeros, at input point t
- To bag C, we draw bootstrap samples

$$\mathbf{x}^{*1},\dots,\mathbf{x}^{*B}$$

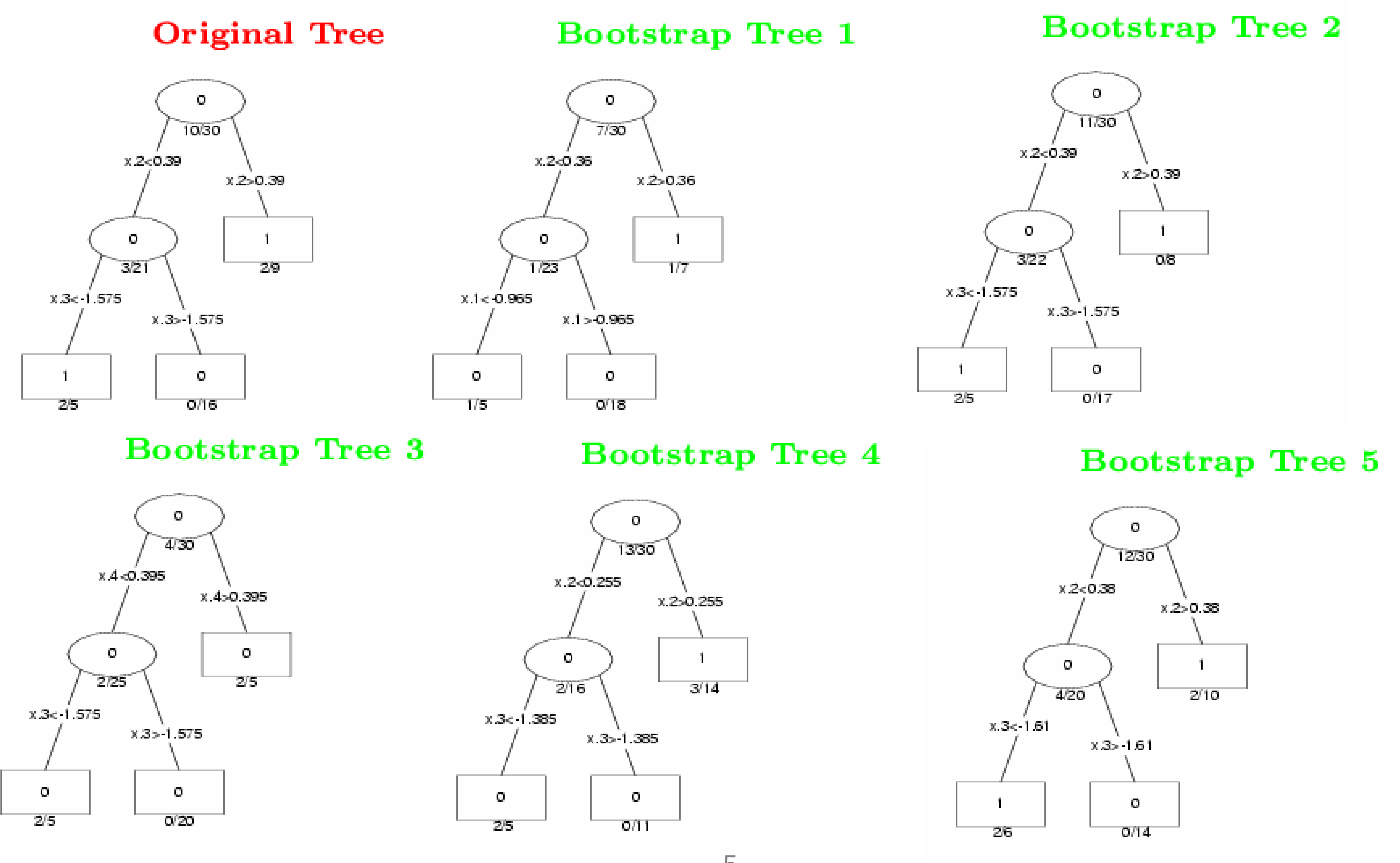
each of size N with replacement from the training data. Then

$$\hat{C}_{bag}(t)=\frac{1}{B}\sum C(\mathbf{x}^{*b},t)$$

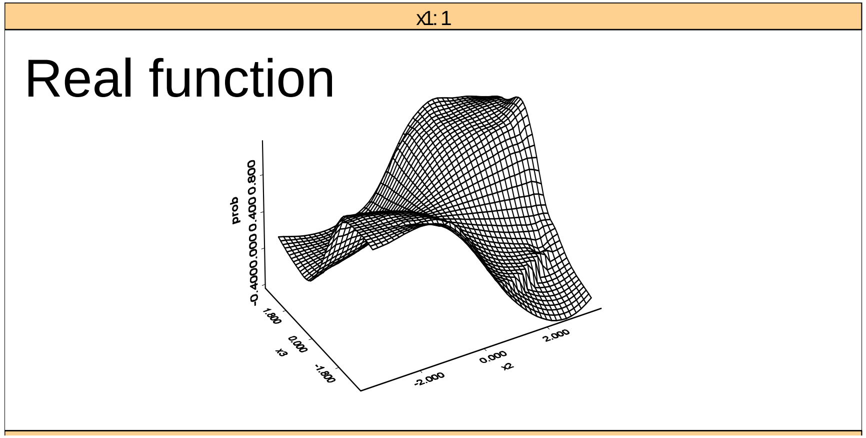
and we classify to the class with largest “vote” in $\hat{C}_{bag}$.

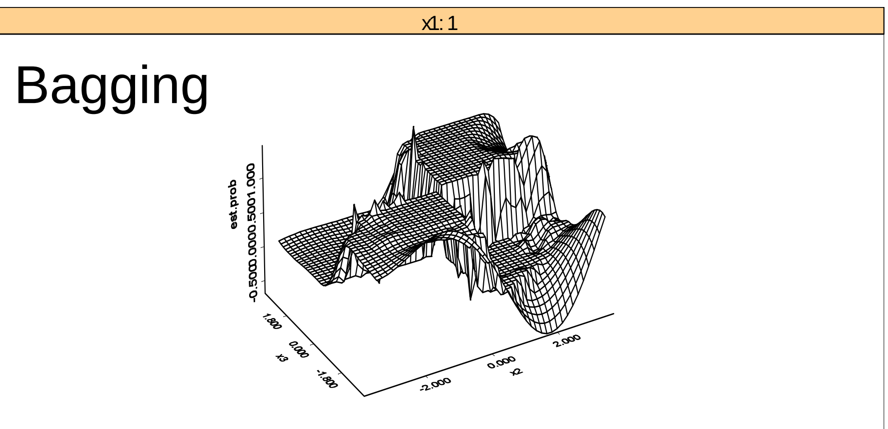
- Bagging can dramatically reduce the variance of unstable procedure (like trees), leading to improved prediction. However any simple structure in C (e.g a tree) is lost.

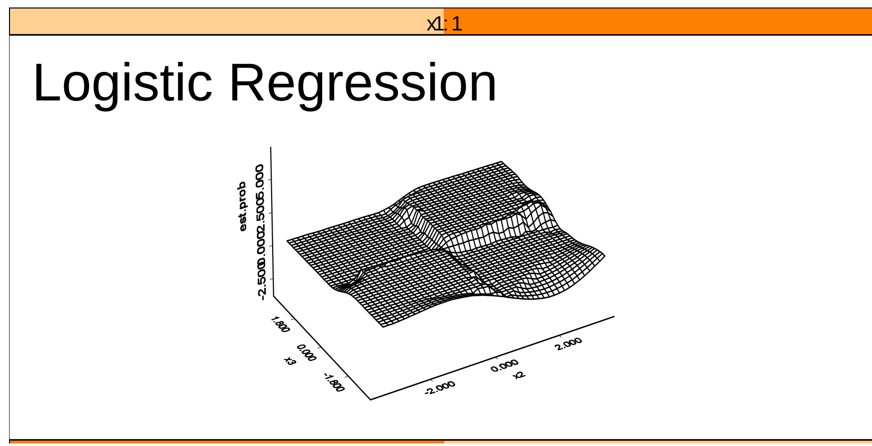
Discussions on Trees



Decision boundary : Bagging

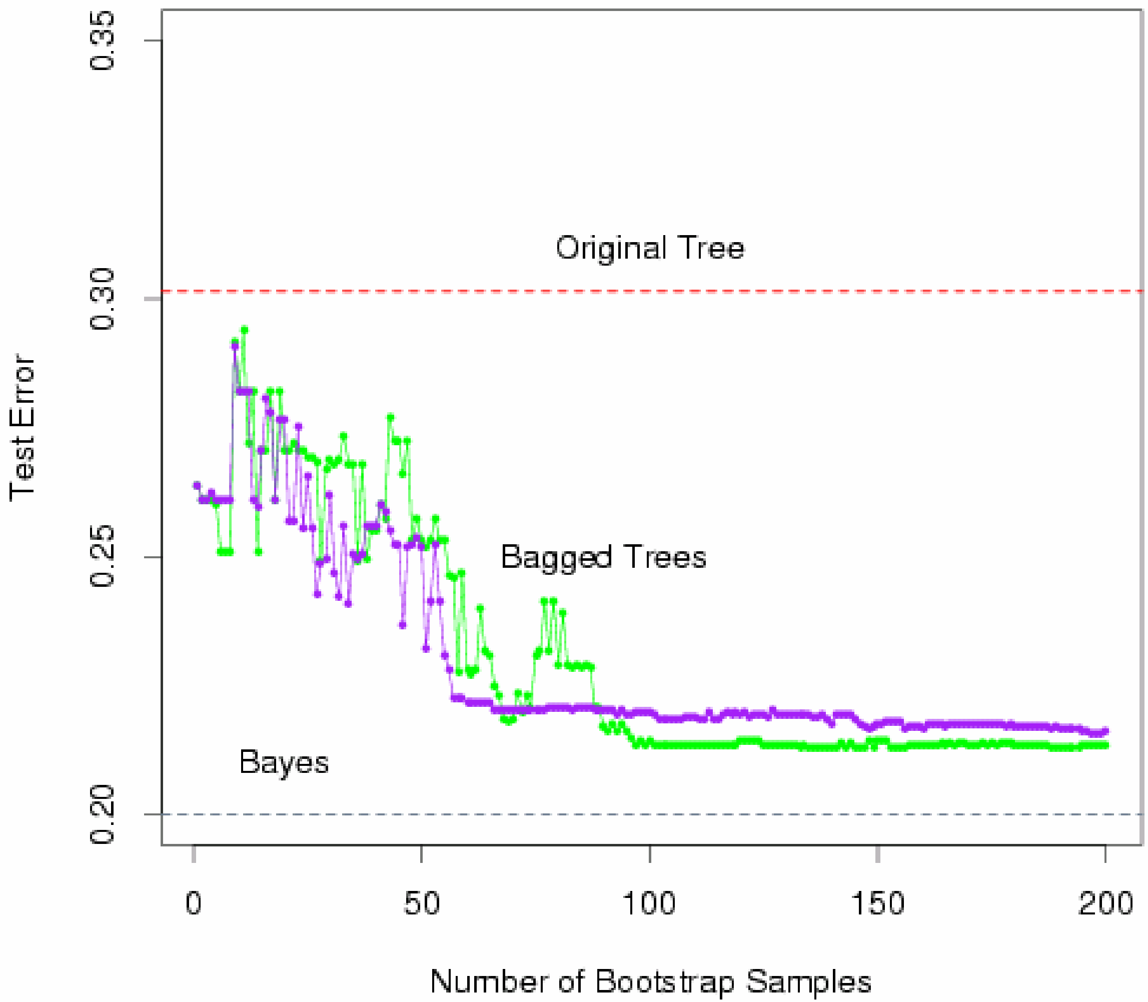






- Bagging averages many trees, and produces *smoother* decision boundary.

Discussions on bagging



Arcing

- Arcing (Adaptive Resampling and Combining) designed exclusively for classification problem.
- Developed by Yoav Freund and Robert Shapire (1995), in the name of “boosting”, but Breiman renamed it as “arcing”.
- Idea: Like bagging, but take unequal probability bootstrap samples. Put more weight on observations that are misclassified, to make the classifier work harder on those points
- Arcing can improve on bagging in many instances.

Arcing-fs algorithm-1

- Step 0. Start with equal observation weights $p(i)=1/N$
- Step 1. At the k-th step, using $\{p(i), i=1,\ldots,N\}$, sample with replacement from the training set T to get the updated training set $T^{(k)}$ and construct classifier C_k using $T^{(k)}$.
- Step 2. Run T down the classifier C_k and let $d(i)=1$ if the i th case is classified incorrectly, otherwise zero.

Arcing-fs algorithm-2

- Step 3. Define

$$\varepsilon_k = \sum_i p(i)d(i), \quad \beta_k = (1-\varepsilon_k)/\varepsilon_k$$

and the updated $(k+1)$st step probabilities by

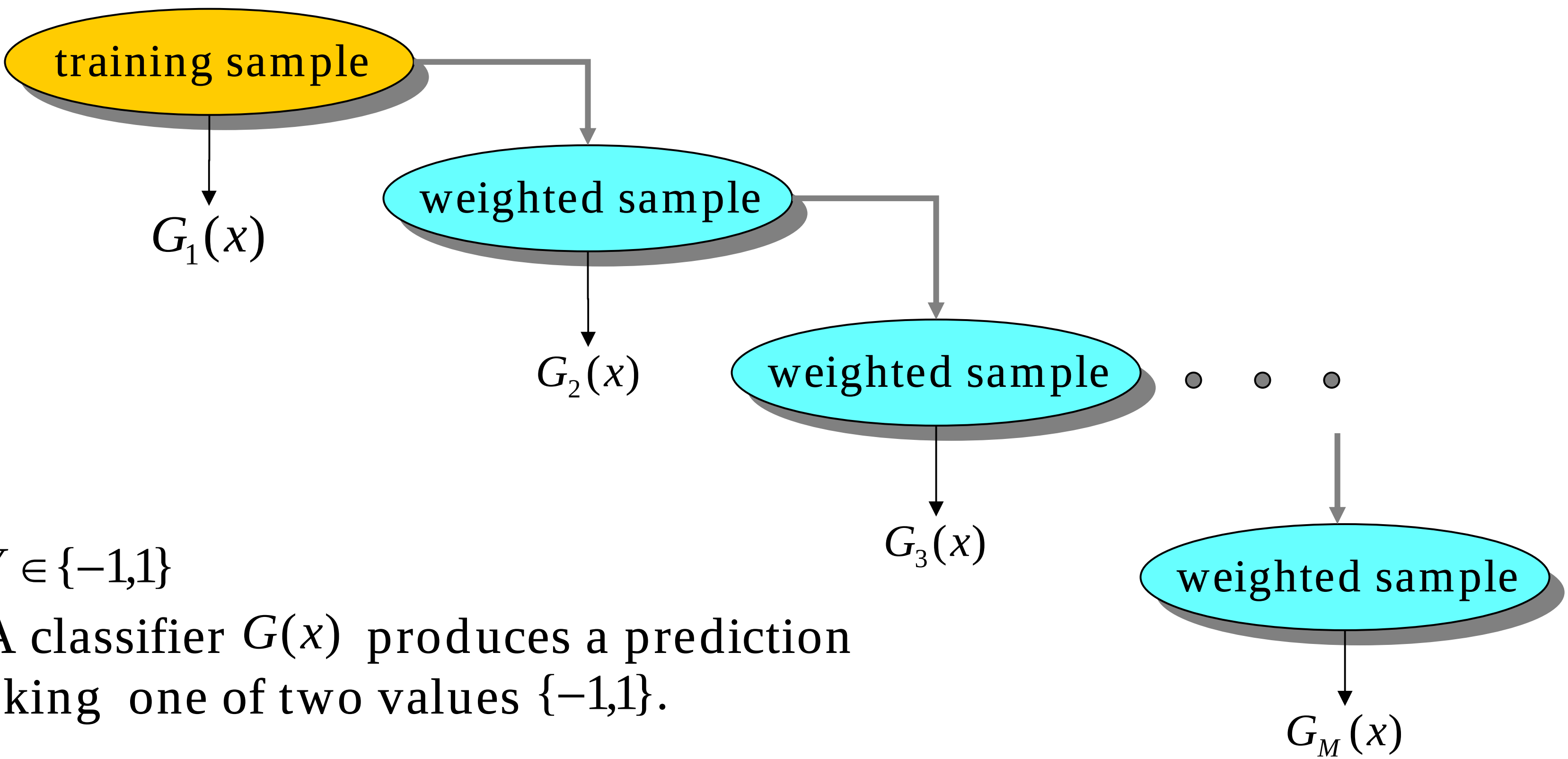
$$p(i) = \frac{p(i)\beta_k^{d(i)}}{\sum p(i)\beta_k^{d(i)}}$$

After K iterations, take a weighted vote of the classifications, with weights $\log(\beta_k)$.

Arcing-fs algorithm-3

- The smaller ε_k, the larger β_k
- $\varepsilon_k > 1/2$ means that the number of misclassification cases is greater than that of good ones.
- If ε_k becomes equal to or greater than 1/2, then the original Freund and Schapire's algorithm exits from the construction loop.
- Breiman (1996) found that better results were gotten by setting all $\{p(i)\}$ equal and restarting.
- If $\varepsilon_k = 0$, again set the probabilities equal to restart.

Boosting



- $Y \in \{-1,1\}$
- A classifier $G(x)$ produces a prediction taking one of two values $\{-1,1\}$.

Final Classifier

$$G(x) = sign[\sum_{m=1}^{M} \alpha_m G_m(x)]$$

Stochastic Model search – Bumping

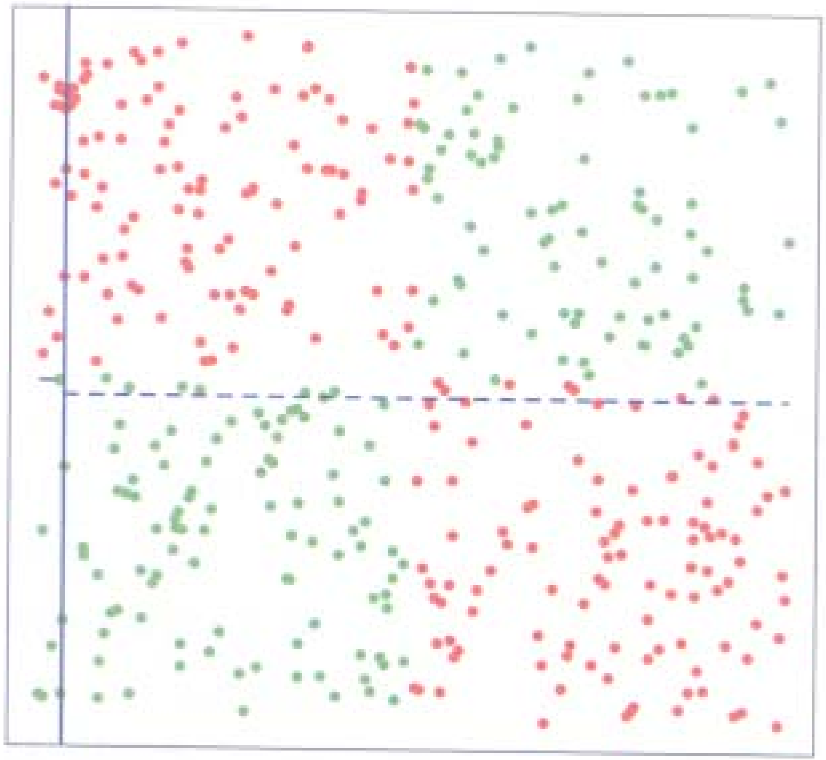
- Bumping stands for *Bootstrap Umbrella of Model Parameters*.
- As in bagging, we draw bootstrap samples and fit a model to each. Rather than average the predictions, we choose the bootstrap model that best fits the training data.
- In detail, we draw bootstrap samples $\mathbf{x}^{*1},\ldots,\mathbf{x}^{*B}$ and compute the classifiers

$$C_1 = C(\mathbf{x}^{*1},t),\ldots,C_B = C(\mathbf{x}^{*B},t)$$

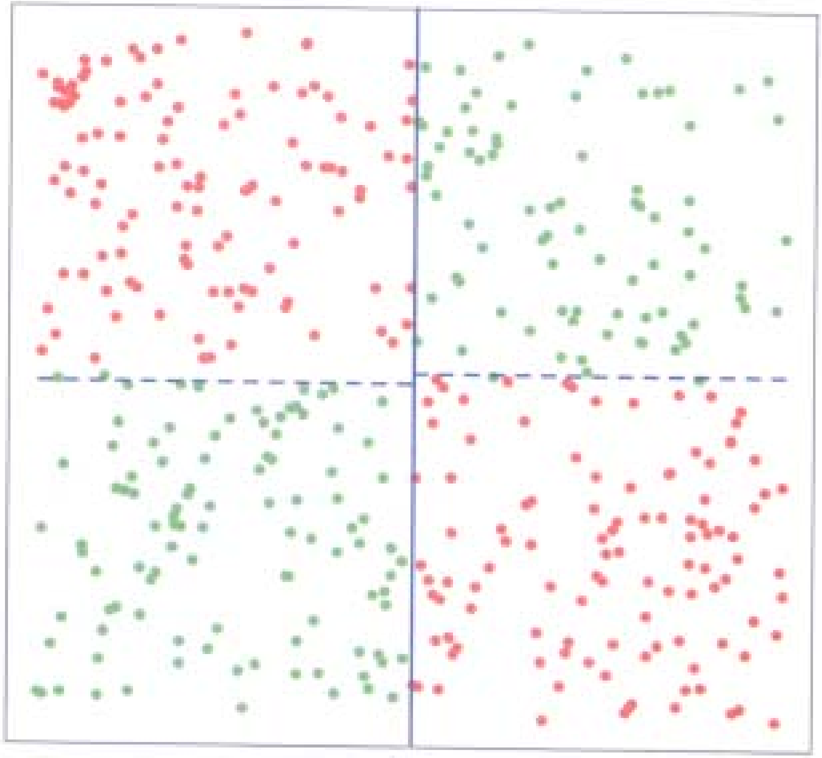
- Then we choose the classifier among $C_1,\ldots,C_B$ that has smallest misclassification error on the original training data.
- Bumping procedure applying to trees and neural nets can find a better local minimum, while still producing a model with the same structure.

Effect of Bumping

Regular 4-Node Tree



Bumped 4-Node Tree



Bumping search in NN

- Algorithm
  - STEP-1. Training data $\mathbf{z}$에 대하여 B개의 bootstrap sample $\mathbf{z}^{*1},\ldots,\mathbf{z}^{*B}$ 을 생성한다.
  - STEP-2. 각 bootstrap sample에 대하여 neural net $C_1,\ldots,C_B$ 을 적합한다. 이 경우 초기 값은 $[-\delta,\delta]$ 에서의 균일분포에서 발생한 난수를 사용한다.
  - STEP-3. Training data $\mathbf{z}$를 분류예측을 위한 신경망 $C_1,\ldots,C_B$ 에 적용하여 오분류 비율이 가장 작은 모형 C_{b_0} 의 최종 수렴한 가중치를 초기치로 설정한다.
- 효과
  - Random search에 비해 1/5 정도의 computing time 하에서도 bumping search는 안정적인 결과를 제공함

Features

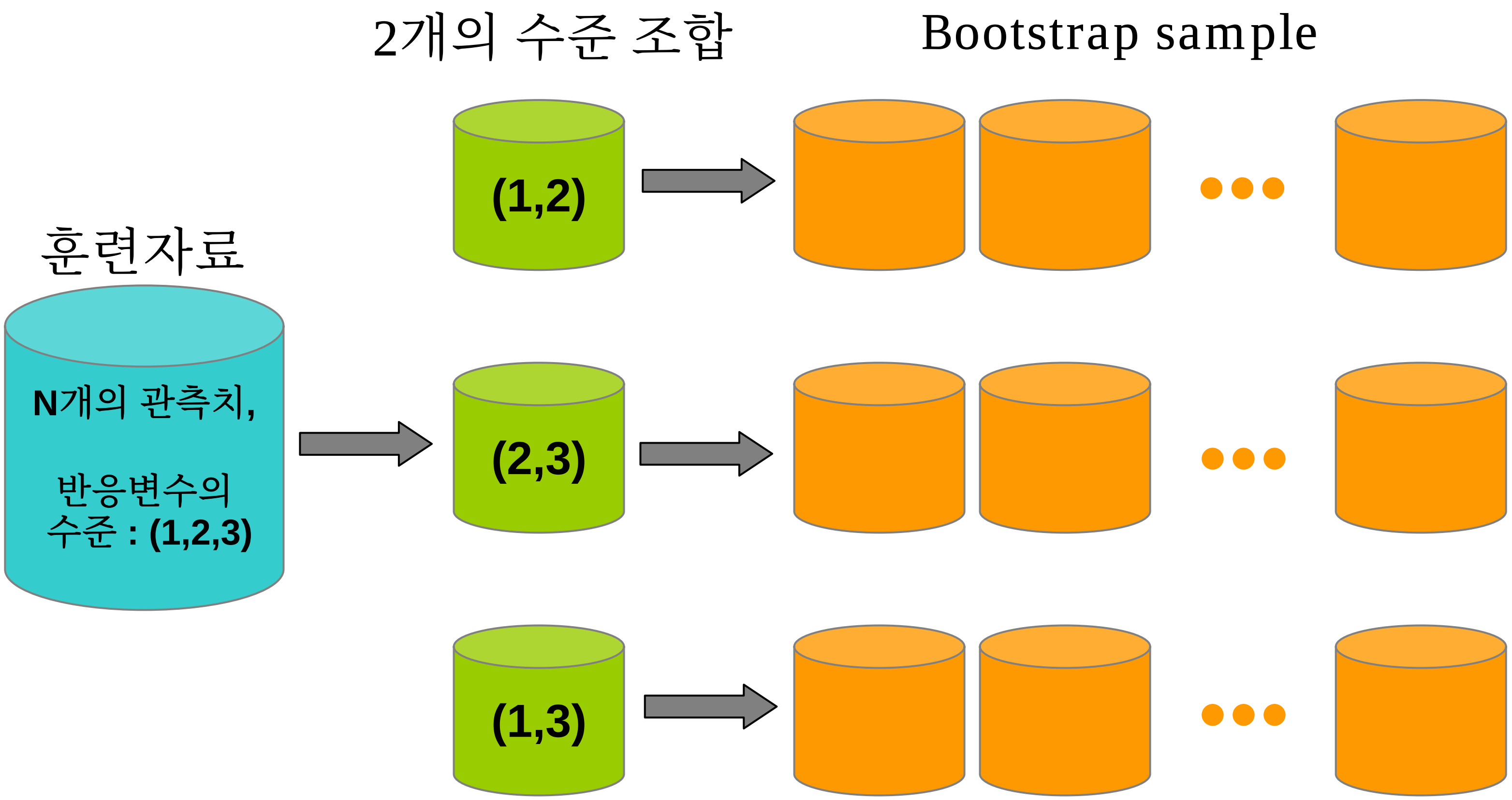
- Bagging improves the performance of classification in tree-based model upto 40% (considering 0-1 loss misclassification error measurement).
- Bagging reduces “variance”, not “bias”
- Bagging destroys tree structure.
- Bagging could not improve the performance of K-nearest neighbor method.
- Arcing is better than Bagging in the sense of misclassification error measurement.

Model Averaging

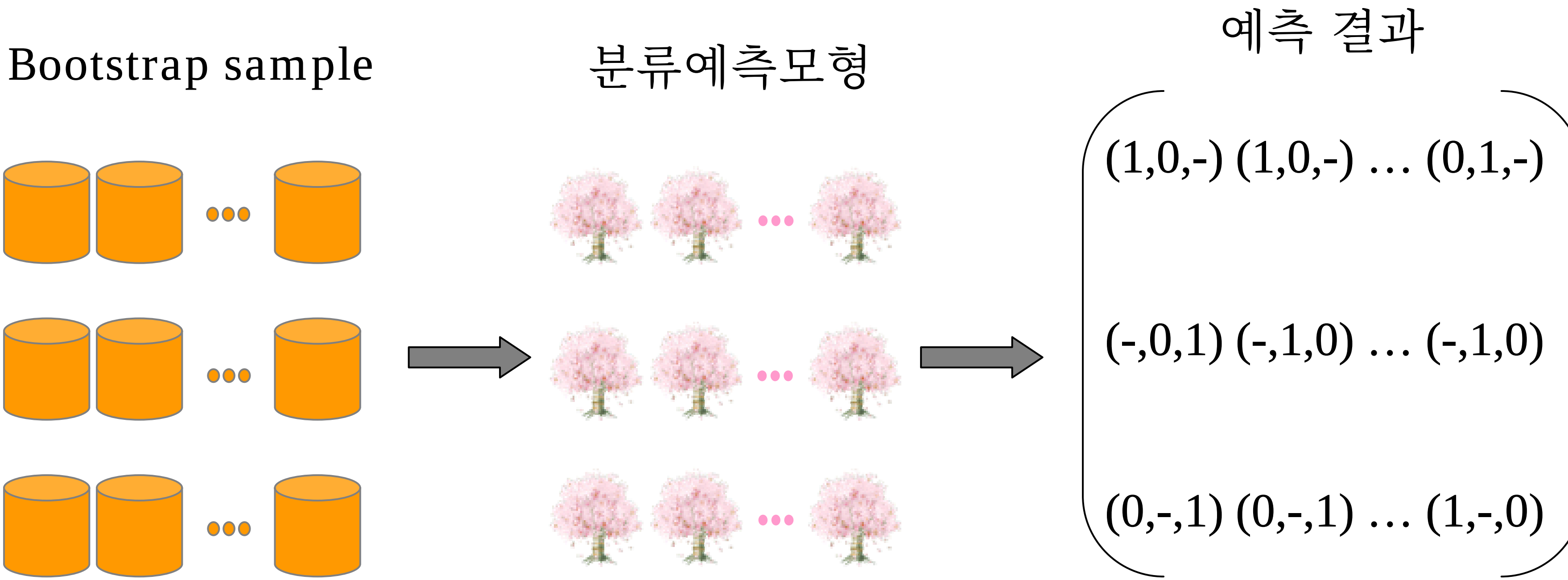
- Classification trees can be simple, but often produce noisy (bushy) or weak (stunted) classifiers.
  - Bagging (Breiman 1996): Fit many large trees to bootstrap-resampled versions of the training data, and classify by majority vote.
  - Boosting (Freund & Schapire 1996): Fit many large or small trees to reweighted versions of the training data. Classify by reweighted majority vote.
- In general Boosting > Bagging > Single Tree in stability and power.
- "AdaBoost … best off-the-shell classifier in the world" (Leo Breiman, NIPS workshop 1996).

Applications – coupled with bagging

- Pairwise comparison

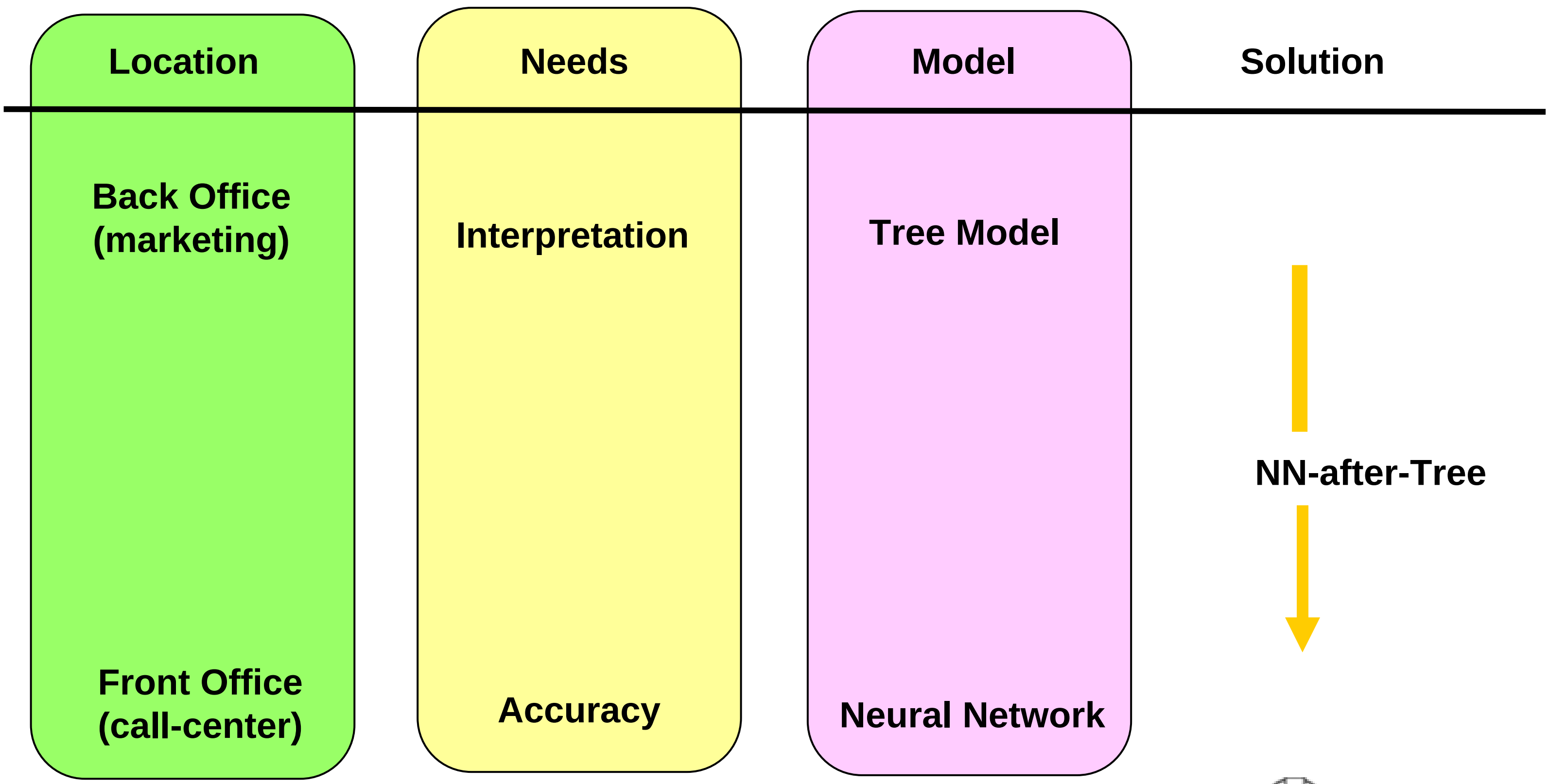


Applications – coupled with bagging *(cont.)*



Applications – hybrid model

- 모 이동통신사에서 고객의 이탈방지 scoring 시, 부서별 needs가 일치하지 않아 scoring 방법선택이 어려웠음.
- 문제해결은 위해 우선 tree model로 segmentation rule을 도출하고, 그 후 segmentation을 dummy 변수로 변환한 뒤, 신경망분석 scoring 결과를 도출함.



Q&A