SVM and SVM Ensembles in Breast Cancer Prediction

Yewen Huang

Anjun Niu

An Chen

Breast Cancer Prevention Model

- Studied Techniques: logistic regression, linear discriminate analysis, artificial neural network and so on.

- Objective: compare performance of SVM and SVM ensembles over small and large scale breast cancer datasets.

Dataset

- Small scale:

  699 data samples: 458 benign (65.5%) and 241 (34.5%) malignant;

  11 different features: 1. sample code number;
  2-10: 9 attributes range from 1 to 10;

  i.e. clump Thickness, Uniformity of Cell, Marginal Adhesion;
  11. class: (benign, malignant)

- Large scale:

  102294 data samples;

  117 different features: detection of breast cancer from X-ray images of the breast

- feature selection(GA): to filter out unrepresentative features;

  Small scale: 10 feature;

  Large scale: 36 features

Experimental Procedure

dataset

- 90% training
- 10% testing

constructing SVM classifier ensembles

- 3 single SVM
- 6 SVM ensembles

Feeding into testing set

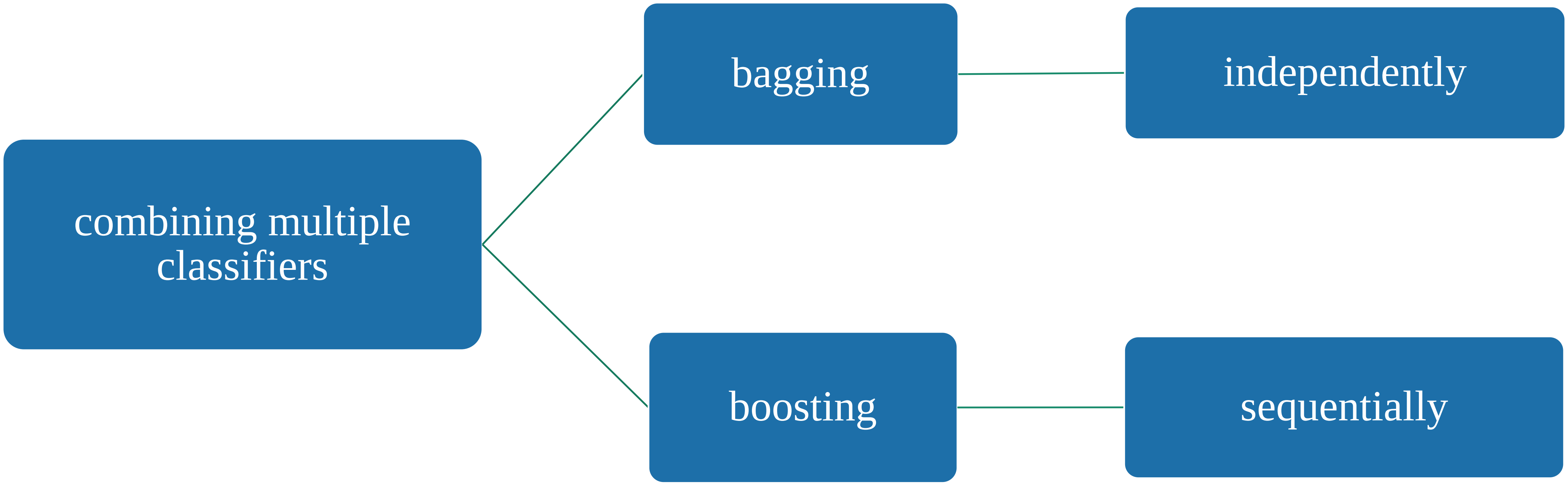
Kernel Function

- Lineal kernel function: K(xi,xj) = Φ(xi) *Φ (xj), Φ : $R^d \rightarrow H^f$, d < f .

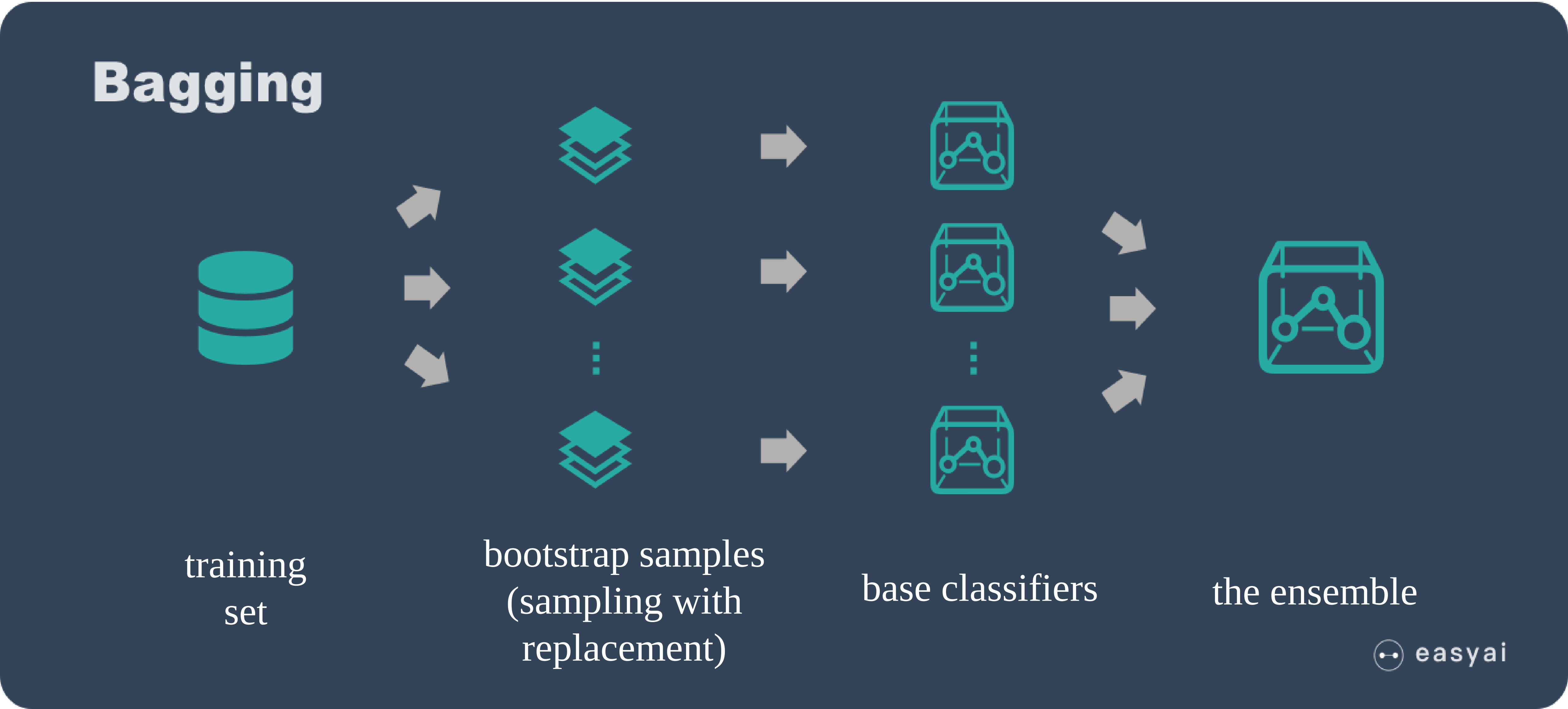
- RBF kernel function: $K_{Gaussian}(x_i, x_j) = e^{\frac{\|x_i - x_j\|^2}{2\sigma}}$ (σ is Gaussian sigma).

- Polynomial kernel function: $K_{poly}(x_i, x_j) = (x_i \cdot x_j + 1)^p$ (p is the degree of polynomial)

Classifier Ensembles



Bagging


Bagging
training set
bootstrap samples (sampling with replacement)
base classifiers
the ensemble
easyai

- Create Bootstrap samples of a training set using sampling with replacement
- Each bootstrap sample is used to train a different component of base classifier
- Classification is done by plurality voting

BAGGING

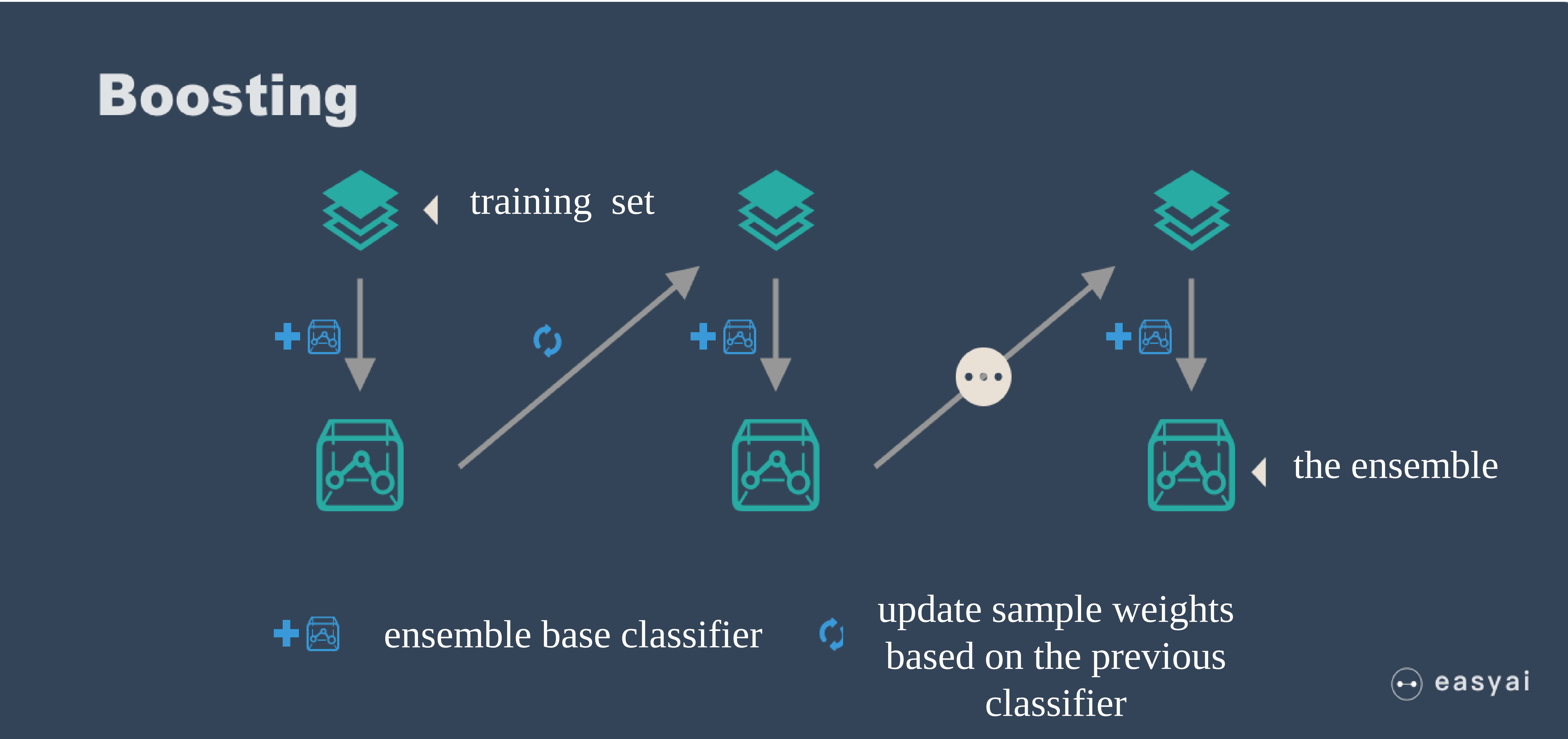
Training phase

1. Initialize the parameters
   - $\mathcal{D} = \emptyset$, the ensemble.
   - L, the number of classifiers to train.
2. For $k = 1, \ldots, L$
   - Take a bootstrap sample S_k from $\mathbf{Z}$.
   - Build a classifier D_k using S_k as the training set.
   - Add the classifier to the current ensemble, $\mathcal{D} = \mathcal{D} \cup D_k$.
3. Return $\mathcal{D}$.

Classification phase

4. Run $D_1, \ldots, D_L$ on the input $\mathbf{x}$.
5. The class with the maximum number of votes is chosen as the label for $\mathbf{x}$.

Boosting


Boosting
training set
the ensemble
ensemble base classifier
update sample weights
based on the previous
classifier
easyai

- Sequential training of weak learners
  - Each base classifier is trained on data that is weighted based on the performance of the previous classifier
- Each classifier votes to obtain a final outcome

Training phase

1. Initialize the parameters
   - Set the weights $\mathbf{w}^1 = [w_1, \ldots, w_N]$, $w_j^1 \in [0, 1]$, $\sum_{j=1}^{N} w_j^1 = 1$. (Usually $w_j^1 = \frac{1}{N}$).
   - Initialize the ensemble $\mathcal{D} = \emptyset$.
   - Pick L, the number of classifiers to train.
2. For $k = 1, \ldots, L$
   - Take a sample S_k from $\mathbf{Z}$ using distribution $\mathbf{w}^k$.
   - Build a classifier D_k using S_k as the training set.
   - Calculate the weighted ensemble error at step k by
     $$\epsilon_k = \sum_{j=1}^{N} w_j^k l_k^j,$$
     ($l_k^j = 1$ if D_k misclassifies $\mathbf{z}_j$ and $l_k^j = 0$ otherwise.)
   - If $\epsilon_k = 0$ or $\epsilon_k \geq 0.5$, ignore D_k, reinitialize the weights w_j^k to $\frac{1}{N}$ and continue.
   - Else, calculate
     $$\beta_k = \frac{\epsilon_k}{1 - \epsilon_k}, \quad \text{where} \quad \epsilon_k \in (0, 0.5),$$
   - Update the individual weights
     $$w_j^{k+1} = \frac{w_j^k \beta_k^{(1-l_k^j)}}{\sum_{i=1}^{N} w_i^k \beta_k^{(1-l_k^i)}}, \quad j = 1, \ldots, N.$$
3. Return $\mathcal{D}$ and $\beta_1, \ldots, \beta_L$.

Classification phase

4. Calculate the support for class ω_t by
   $$\mu_t(\mathbf{x}) = \sum_{D_k(\mathbf{x}) = \omega_t} \ln\left(\frac{1}{\beta_k}\right).$$
5. The class with the maximum support is chosen as the label for $\mathbf{x}$.

Comparison between Bagging and Boosting

- Sampling

  Bagging: training set is chosen with replacement and train sets of each round are independent.

  Boosting: train set does not change for each round.

- Weight

  Bagging: evenly

  Boosting: weight changes based on he loss function(error). Less the accuracy, larger the weight.

- classifier

  Bagging: classifiers weight evenly

  Boosting: every classifier has corresponding weight. Classifier which is more accurate has larger weight.

- computation

  Bagging: classifier generated simultaneously

  Boosting: classifier generated order by order because the parameters used in n-th model are produced by the (n-1) th model

- Variance

  Bagging: variance decreases as number of model increases

Compare SVM and SVM ensembles

- Classification accuracy
- ROC (receiver operation characteristic)

Hypothesized class \ True class	p	n
Y	True Positives	False Positives
N	False Negatives	True Negatives
Column totals:	P	N

$$\text{fp rate} = \frac{FP}{N} \qquad \text{tp rate} = \frac{TP}{P}$$

$$\text{precision} = \frac{TP}{TP+FP}$$

$$\text{accuracy} = \frac{TP+TN}{P+N}$$

$$\text{F-measure} = \frac{2}{1/\text{precision}+1/\text{recall}}$$

- F-measure
- Computational times of training

Experimental Results

- The performance of single SVM classifiers vs SVM classifier ensembles

SVM ensembles > single SVM classifier

- The performance of applying genetic algorithm (GA)

Small scale: with GA > without GA

Comparison of the the classification accuracy, ROC, and F-measure of the top 3 classifiers

Classification accuracy		ROC		F-measure	
Small scale dataset					
1	GA+RBF SVM ensembles (boosting) (98.28%)	1	GA+linear/poly SVM ensembles (bagging) (0.98)	1	GA+RBF SVM (0.988)
2	GA+linear SVM (96.85%)	2	GA+poly SVM ensembles (boosting) (0.979)	2	GA+linear SVM ensembles (bagging/boosting) (0.966)
3	GA+linear SVM ensembles (bagging/boosting) (96.57%)	3	GA+RBF SVM ensembles (boosting) (0.977)	3	GA+RBF SVM ensembles (boosting) (0.963)
Large scale dataset					
1	RBF SVM ensembles (boosting) (99.52%)	1	GA+linear SVM ensembles (boosting) (0.876)	1	RBF SVM ensembles (boosting) (0.995)
2	Poly SVM ensembles (bagging) (99.51%)	2	GA+RBF SVM ensembles (boosting) (0.875)	2	Poly SVM; poly SVM ensembles (bagging); GA+poly SVM ensembles (bagging); GA+RBF SVM ensembles (boosting) (0.994)
3	Poly SVM; GA+poly SVM; RBF SVM ensembles (bagging); GA+poly SVM ensembles (bagging) (99.50%)	3	RBF SVM ensembles (boosting) (0.869)		

Reference

- Huang M-W, Chen C-W, Lin W-C, Ke S-W, Tsai C-F (2017) SVM and SVM Ensembles in Breast Cancer Prediction. PLoS ONE 12(1): e0161501. doi:10.1371/journal.pone.0161501
- Fawcett T. An introduction to ROC analysis. Pattern Recognition Letters. 2006; 27: 861–874
- Oh I.S., Lee J.S., Moon B.R. Hybrid genetic algorithms for feature selection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004; 26(11): 1424–1437. PMID: 15521491
- Breiman L. Bagging predictors. Machine Learning. 1996; 24(2): 123–140.
- Freund Y. Schapire R.E. Experiments with a new boosting algorithm. International Conference on Machine Learning; 1996 July 3–6; Bari, Italy. p. 148–156.