Ensembles: Combining Label Outputs

Javier Sáez

Universidad de Granada

January 18, 2023

Index

What is a classifier ensemble?

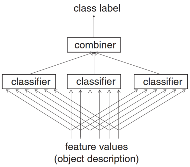


Figure: Classifier ensemble

What is a classifier ensemble?

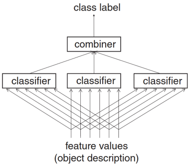


Figure: Classifier ensemble

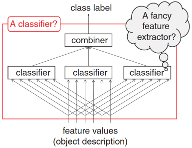


Figure: ¿ Ensemble = classifier ? (term)

What is a classifier ensemble?

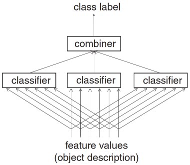


Figure: Classifier ensemble

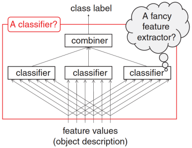


Figure: ¿ Ensemble = classifier ? (term)

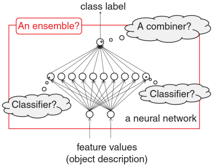


Figure: ¿ NN = ensemble ?

What is a classifier ensemble?

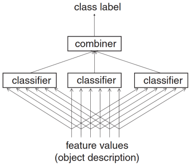


Figure: Classifier ensemble

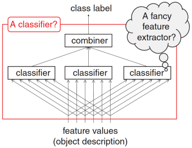


Figure: ¿ Ensemble = classifier ? (term)

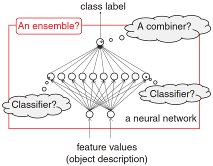


Figure: ¿ NN = ensemble ?

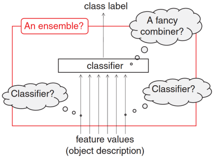
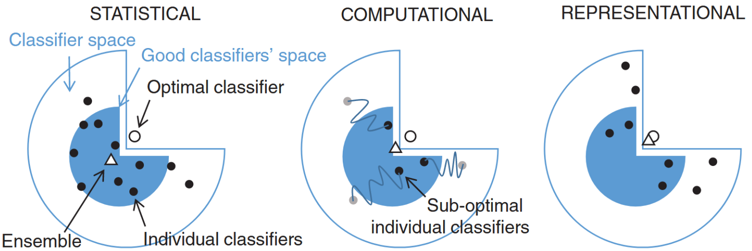


Figure: ¿ Any classifier = ensemble ?

Why do ensembles work?



Statistical

- Reduce randomness of data and training algorithms
- Improve generalization

Computational

- Avoid local optima
- Split data between classifiers
- ¿ Small amount of data ? $\Longrightarrow$ Resample
- Divide and conquer

Representational

- Approximate boundaries using simpler ones

Taxonomy

- **Combiner**: Non trainable/Trainable/Meta-classifier
- **Training the ensemble**: Independent/Incremental training
- **Diversity**:
  - Training of base classifiers
  - Resampling data
  - Partitioning data
  - Different base models
  - Different labels
- **Ensemble size**: Fixed in advance/during training/overproduce
- **Universality**: Specified/Any base classifier model

Combination Overview

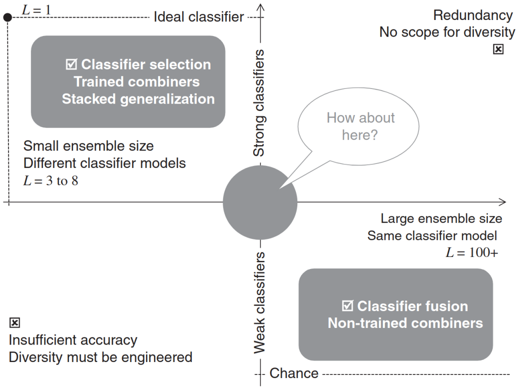


Figure: Summary of popular classifier combination approaches.

Types of outputs

Consider $\mathcal{D} = \{D_i\}_{i=1}^{L}$ and a set of classes $\mathbf{\Omega} = \{\omega_i\}_{i=1}^{c}$. Labels can be:

Types of outputs

Consider $\mathcal{D} = \{D_i\}_{i=1}^{L}$ and a set of classes $\mathbf{\Omega} = \{\omega_i\}_{i=1}^{c}$. Labels can be:

- Class label: $D_i : \mathbb{R}^n \to \mathbf{\Omega}$ and the L classifiers define a vector $\mathbf{s} = [s_1, \ldots, s_L]^T \in \mathbf{\Omega}^L$.

Types of outputs

Consider $\mathcal{D} = \{D_i\}_{i=1}^{L}$ and a set of classes $\mathbf{\Omega} = \{\omega_i\}_{i=1}^{c}$. Labels can be:

- Class label: $D_i : \mathbb{R}^n \to \mathbf{\Omega}$ and the L classifiers define a vector $\mathbf{s} = [s_1, \ldots, s_L]^T \in \mathbf{\Omega}^L$.
- Ranked class labels: $D_i : \mathbb{R}^n \to \mathbf{\Omega}^k$, suitable for large number of classes.

Types of outputs

Consider $\mathcal{D} = \{D_i\}_{i=1}^{L}$ and a set of classes $\mathbf{\Omega} = \{\omega_i\}_{i=1}^{c}$. Labels can be:

- Class label: $D_i : \mathbb{R}^n \to \mathbf{\Omega}$ and the L classifiers define a vector $\mathbf{s} = [s_1, \ldots, s_L]^T \in \mathbf{\Omega}^L$.
- Ranked class labels: $D_i : \mathbb{R}^n \to \mathbf{\Omega}^k$, suitable for large number of classes.
- Numerical support for classes: $D_i : \mathbb{R}^n \to [0,1]^c$. We create a matrix $\mathbf{D}$ where $d_{i,j}$ is the support value that classifier i assigns $\mathbf{x}$ to belong to class j.

Types of outputs

Consider $\mathcal{D} = \{D_i\}_{i=1}^{L}$ and a set of classes $\mathbf{\Omega} = \{\omega_i\}_{i=1}^{c}$. Labels can be:

- Class label: $D_i : \mathbb{R}^n \rightarrow \mathbf{\Omega}$ and the L classifiers define a vector $\mathbf{s} = [s_1, \ldots, s_L]^T \in \mathbf{\Omega}^L$.
- Ranked class labels: $D_i : \mathbb{R}^n \rightarrow \mathbf{\Omega}^k$, suitable for large number of classes.
- Numerical support for classes: $D_i : \mathbb{R}^n \rightarrow [0, 1]^c$. We create a matrix $\mathbf{D}$ where $d_{i,j}$ is the support value that classifier i assigns $\mathbf{x}$ to belong to class j.
- Oracle: We only consider if the classifier is correct or wrong: $D_i : \mathbb{R}^n \rightarrow \{1, 0\}$, where 1 indicates that D_i classified $\mathbf{x}$ correctly.

Probabilistic framework

Given $\mathbf{s} = [s_1, \ldots, s_L]^T$, we are interested in

$$p(\omega_k \mid \mathbf{s}), \quad k = 1, \ldots, c.$$

We **assume** that the classifier give **independent** decissions, **conditioned upon the class label**:

$$p(\mathbf{s} \mid \omega_k) = p(s_1 \mid \omega_k) \ldots p(s_L \mid \omega_k).$$

Probabilistic framework

Given $\mathbf{s} = [s_1, \ldots, s_L]^T$, we are interested in

$$p(\omega_k \mid \mathbf{s}), \quad k = 1, \ldots, c.$$

We **assume** that the classifier give **independent** decissions, **conditioned upon the class label**:

$$p(\mathbf{s} \mid \omega_k) = p(s_1 \mid \omega_k) \ldots p(s_L \mid \omega_k).$$

Using Bayes rule

$$\begin{aligned} p(\omega_k \mid \mathbf{s}) &= \frac{p(\mathbf{s} \mid \omega_k) p(\omega_k)}{p(\mathbf{s})} = \frac{p(\omega_k)}{p(\mathbf{s})} \prod_{i=1}^{L} p(s_i \mid \omega_k) \\ &= \frac{p(\omega_k)}{p(\mathbf{s})} \times \prod_{i \in I_+^k} p(s_i \mid \omega_k) \times \prod_{i \in I_-^k} p(s_i \mid \omega_k) \end{aligned}$$

Majority Vote

Unanimity	■	■	■	■	■	■	■	■	■	■
Simple majority	■	■	■	■	■	■	△	△	△	△
Plurality	■	■	■	■	△	△	△	×	×	×

Considering $d_{i,j} = 1$ if D_i labels $\mathbf{x}$ in ω_j and 0 otherwise, the **plurality vote** (*a.k.a. majority vote*) returns ω_k if

$$\sum_{i=1}^{L} d_{i,k} = \max_{j=1}^{c} \sum_{i=1}^{L} d_{i,j}.$$

Thresholded majority vote adds a needed confidence for the vote to be valid:

$$\begin{cases} \omega_k, & \text{if } \sum_{i=1}^{L} d_{i,j} \geq \alpha L \\ \omega_{c+1}, & \text{otherwise} \end{cases}$$

with $0 < \alpha \leq 1$.

Accuracy of Majority Vote

Assuming:

- The number of classifiers L is odd.
- Each classifiers assigns the correct class label with a probability p for any input.
- The classifier outputs are independent.

Majority vote will give an accurate class label if **at least** $\lfloor L/2 \rfloor + 1$ classifiers give correct answers. Then, the accuracy of the ensemble is:

$$p_{maj} = \sum_{\lfloor L/2 \rfloor + 1}^{L} \binom{L}{m} p^m (1-p)^{L-m} \tag{1}$$

Condorcet Jury Theorem

Theorem (Condorcet Jury Theorem)

In the previously presented conditions:

1. *If* $p > 0.5$, *then* p_{maj} *is monotonically increasing and*

$$p_{maj} \to 1 \quad as \quad L \to \infty.$$

2. *If* $p < 0.5$ *then* p_{maj} *is monotonically decreasing and*

$$p_{maj} \to 0 \quad as \quad L \to \infty.$$

3. *If* $p = 0.5$, *then* $p_{maj} = 0.5$ *for any* L.

Pattern of success

Intuitively: Best improvement over individual accuracy is achieved when exactly $\lfloor L/2 \rfloor + 1$ votes are correct. Extras are wasted.

Pattern of success

Intuitively: Best improvement over individual accuracy is achieved when exactly $\lfloor L/2 \rfloor + 1$ votes are correct. Extras are wasted.

Definition (Pattern of success)

The **pattern of success** is a distribution of the L classifier outputs such that:

- The probability of any combination of $\lfloor L/2 \rfloor + 1$ correct and $\lfloor L/2 \rfloor$ incorrect is α.
- The probability of all votes being incorrect is γ.
- Any other combination has probability 0.

Pattern of success

Intuitively: Best improvement over individual accuracy is achieved when exactly $\lfloor L/2 \rfloor + 1$ votes are correct. Extras are wasted.

Definition (Pattern of success)

The **pattern of success** is a distribution of the L classifier outputs such that:

- The probability of any combination of $\lfloor L/2 \rfloor + 1$ correct and $\lfloor L/2 \rfloor$ incorrect is α.
- The probability of all votes being incorrect is γ.
- Any other combination has probability 0.

In this scenario, the accuracy of the ensemble is:

$$p_{maj} = \min\left\{1, \frac{2pL}{L+1}\right\}$$

Pattern of failure

Definition (Pattern of failure)

The **pattern of failure** is a distribution of the L classifier outputs such that:

- The probability of any combination of $\lfloor L/2 \rfloor$ correct and $\lfloor L/2 \rfloor + 1$ incorrect is β.
- The probability of all votes being correct is δ.
- Any other combination has probability 0.

Pattern of failure

Definition (Pattern of failure)

The **pattern of failure** is a distribution of the L classifier outputs such that:

- The probability of any combination of $\lfloor L/2 \rfloor$ correct and $\lfloor L/2 \rfloor + 1$ incorrect is β.
- The probability of all votes being correct is δ.
- Any other combination has probability 0.

In this scenario, the accuracy of the ensemble is:

$$p_{maj} = \frac{(2p-1)L+1}{L+1}$$

Matan's bounds on the Majority Vote Accuracy

Consider that classifier D_i has accuracy p_i and $\{D_1, \dots, D_L\}$ are arranged so that $p_1 \leq p_2 \leq \dots \leq p_L$. Let $k = (L+1)/2$. Then the accuracy of the majority vote ensemble has the following lower and upper bounds:

$$\max\{0, \xi(k), \xi(k-1), \dots, \xi(1)\} \leq p_{maj} \leq \min\{1, \Sigma(k), \Sigma(k-1), \dots, \Sigma(1)\}$$

where

$$\Sigma(m) = \frac{1}{m} \sum_{i=1}^{L-k+m} p_i, \quad m = 1, \dots, k,$$

and

$$\xi(m) = \frac{1}{m} \sum_{i=k-m+1}^{L} p_i - \frac{L-k}{m}, \quad m = 1, \dots, k$$

Optimality of the Majority Vote Combiner

Theorem

Let $\mathcal{D}$ be an ensemble of L classifiers. Suppose that:

1. *The classifiers give their decisions independently, conditioned upon the class label.*
2. *The individual classification accuracy is p for all the classifiers, classes and datapoints.*
3. *The probability for incorrect classification is equally distributed among the remaining classes:*

$$P(s_i = \omega_j \mid \omega_k) = \frac{1-p}{c-1}, \quad i = 1, \ldots, L; \; k, j = 1, \ldots, c \; j \neq k$$

Then, the majority vote is the optimal combination rule.

Weighted Majority Vote (WMV)

If the classifiers in the ensemble do not have identical accuracy, it is reasonable to give the more competent classifiers more power in the final decision.

Using the previous $d_{i,j}$, the **class-support** function for class ω_j obtained through weighted voting is:

$$\mu_j(\mathbf{x}) = \sum_{i=1}^{L} b_i d_{i,j},$$

where b_i is a coefficient for classifier D_i.

The value of this function will be the sum of the weights for the classifiers of the ensemble whose output for $\mathbf{x}$ is ω_j.

Optimality of Weighted Majority Vote

Theorem

Let $\mathcal{D}$ be an ensemble of L classifiers. Suppose that:

- *The classifiers give their decisions independently, conditioned upon the class label.*
- *The individual classification accuracy is p_i for any class ω_k and any datapoint. (We relax the assumption about equal individual accuracies)*
- *The probability for incorrect classification is equally distributed among the remaining classes.*

Then the WMV is the optimal combination rule with weights:

$$b_i = \log\left(\frac{p_i}{1-p_i}\right), \quad 0 \leq p_i \leq 1 \tag{2}$$

Naïve Bayes Combiner

Definition (Naïve Bayes Classifier)

Given an unknown sample $\mathbf{x}^\star$, the *Naive Bayes (NB)* **classifier** assigns the sample a label according to the *maximum a posteriori* decision rule:

$$y^\star = \arg\max_k p(\omega_k) \prod_{i=1}^{L} p(x_i \mid \omega_k)$$

The Naïve Bayes **combiner** applies the NB classifier in the case where the input $\mathbf{x}^\star$ is a vector of the outputs of each of the classifiers.

Naïve Bayes Combiner

Definition (Naïve Bayes Classifier)

Given an unknown sample $\mathbf{x}^\star$, the *Naive Bayes (NB)* **classifier** assigns the sample a label according to the *maximum a posteriori* decision rule:

$$y^\star = \arg\max_k p(\omega_k) \prod_{i=1}^{L} p(x_i \mid \omega_k)$$

The Naïve Bayes **combiner** applies the NB classifier in the case where the input $\mathbf{x}^\star$ is a vector of the outputs of each of the classifiers.

Implementation detail: In practise, the probabilities are estimated by computing a confusion matrix for each of the classifiers using the whole dataset.

$$p(s_i \mid \omega_k) = \frac{cm^i_{k,s_i}}{N_k}, \quad p(\omega_k) = \frac{N_k}{N}$$

Optimality of the Naïve-Bayes Combiner

Theorem

Let $\mathcal{D}$ be an ensemble of L classifiers. Suppose that the classifiers give their decisions independently, conditioned upon the class label. Then, the Naïve Bayes combiner

$$\omega^{\star} = \max \left\{ p(\omega_k) \prod_{i=1}^{L} p(s_i \mid \omega_k) \right\}$$

is the optimal combination rule.

Behavior Knowledge Space (BKS)-[Huang and Suen, 1995]

A **Behaviour Knowledge Space** is a $L-$ dimensional space where each dimension correspond to the decision of one classifier.

Behavior Knowledge Space (BKS)-[Huang and Suen, 1995]

A **Behaviour Knowledge Space** is a $L-$ dimensional space where each dimension correspond to the decision of one classifier.
Consider $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{s_x} = [s_1, \dots, s_L]^T$ the labels asigned by each of the L classifiers to $\mathbf{x}$. Consider also:

- $\mathbf{E} \in \mathcal{M}_{N \times L}$, where the i-th row $\mathbf{E}_i$ contains the vector $\mathbf{s}_{\mathbf{x}_i}$ of the training set.
- $\mathbf{T} \in \mathcal{M}_{N \times 1}$, the vector of the true labels of the training set.
- ω_p is the most represented class in $\mathbf{T}$.
- $R(\mathbf{S})$ is the most represented class in the set $\mathbf{S}$. Also, $n_{\mathbf{S}}(k)$ is the number of times that class k is in $\mathbf{S}$.

BKS Algorithm

Algorithm 1 BKS Combiner

$\mathbf{S} = \{\emptyset\}$
Compute $\mathbf{r} = [D^1(\mathbf{x}^\star), \cdots, D^L(\mathbf{x}^\star)]$
for Each each row $\mathbf{E}_i$ of $\mathbf{E}$ **do**
  if $\mathbf{r} == \mathbf{E}_i$ **then**
    $\mathbf{S} = \mathbf{S} \cup \{\mathbf{T}_i\}$
  end if
end for
if $\mathbf{S} == \{\emptyset\}$ **then**
  return ω_p or ω_{C+1}
else
  return $R(\mathbf{S})$
end if

BKS: Confidence Variant

Let $|\mathbf{S}|$ be the cardinal of $\mathbf{S}$ and $E(\mathbf{x}^\star)$ be the output of the BKS method. Considering a confidence level $0 < \lambda < 1$, BKS can return:

$$E(\mathbf{x}^\star) = \begin{cases} R(\mathbf{S}) & \text{If } |\mathbf{S}| > 0 \text{ and } \frac{n_\mathbf{S}(R(\mathbf{S}))}{|S|} \geq \lambda \\ \omega_p \text{ or } \omega_{C+1} & \text{otherwise.} \end{cases}$$

BKS: Confidence Variant

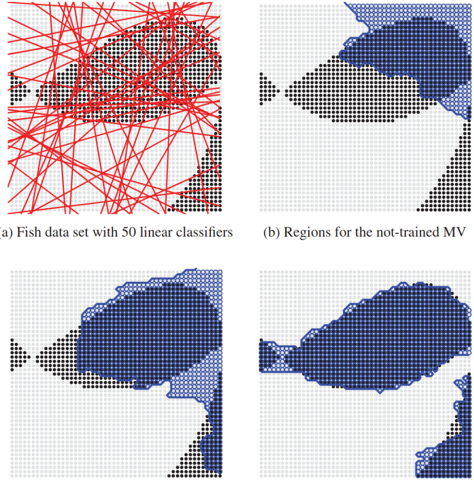
Let $|\mathbf{S}|$ be the cardinal of $\mathbf{S}$ and $E(\mathbf{x}^\star)$ be the output of the BKS method. Considering a confidence level $0 < \lambda < 1$, BKS can return:

$$E(\mathbf{x}^\star) = \begin{cases} R(\mathbf{S}) & \text{If } |\mathbf{S}| > 0 \text{ and } \frac{n_{\mathbf{S}}(R(\mathbf{S}))}{|S|} \geq \lambda \\ \omega_p \text{ or } \omega_{C+1} & \text{otherwise.} \end{cases}$$

Warning

This method needs a big dataset in order to create a representative BKS.

Comparison



(a) Fish data set with 50 linear classifiers (b) Regions for the not-trained MV

(c) Regions for the NB combiner (d) Regions for the BKS combiner

Comparison

Combiner	1	2	3	4	Number of parameters
Majority Vote	■				none
Weighted Majority Vote	■	■			$L + c$
Naive Bayes	■	■	■		$Lc^2 + c$
BKS	■	■	■	■	c^L

Columns mean **scopes of optimality**:

- Equal p
- Classifier specific p_i
- Full confusion matrix
- Independence not required

Thank you for your attention

Bibliography

Yea-Shuan Huang and Ching Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17:90–94, 01 1995. doi: 10.1109/34.368145.