Machine learning

Ensemble Learning

Hamid Beigy

Sharif University of Technology

April 24, 2023

Table of contents

Introduction

1. In our daily life
    1.1 Asking different doctors' opinions before undergoing a major surgery
    1.2 Reading user reviews before purchasing a product.
    1.3 There are countless number of examples where we consider the decision of mixture of experts.
2. Ensemble systems follow exactly the same approach to data analysis.

Problem (Ensemble learning)

- *Given training data set* $S = \{(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)\}$ *drawn from common instance space* X, *and*
- *A collection of inductive learning algorithms,*
- *Return a new classification algorithm for* $x \in X$ *that combines outputs from collection of classification algorithms*

3. Desired Property

    Guarantees of performance of combined prediction.

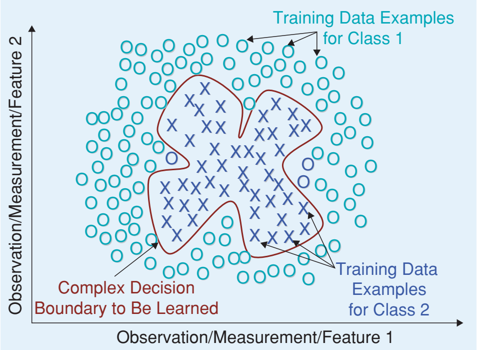
Why we combine classifiers?

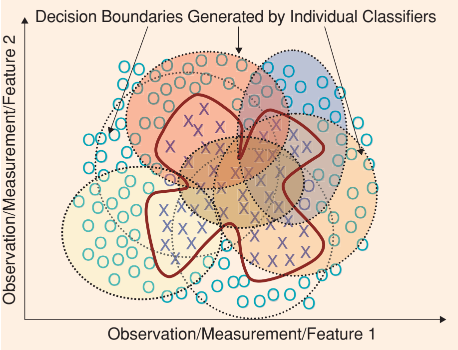
Reasons for using ensemble based systems

1. Statistical reasons
   1.1 A set of classifiers with similar training data may have different generalization performance.
   1.2 Classifiers with similar performance may perform differently in field (depends on test data).
   1.3 In this case, averaging (combining) may reduce the overall risk of decision.
   1.4 In this case, averaging (combining) may or may not beat the performance of the best classifier.
2. Large volumes of data
   2.1 Usually training of a classifier with a large volumes of data is not practical.
   2.2 A more efficient approach is to
       Partition the data into smaller subsets
       Training different Classifiers with different partitions of data
       Combining their outputs using an intelligent combination rule
3. To little data
   3.1 We can use resampling techniques to produce non-overlapping random training data.
   3.2 Each of training set can be used to train a classifier.

Reasons for using ensemble based systems

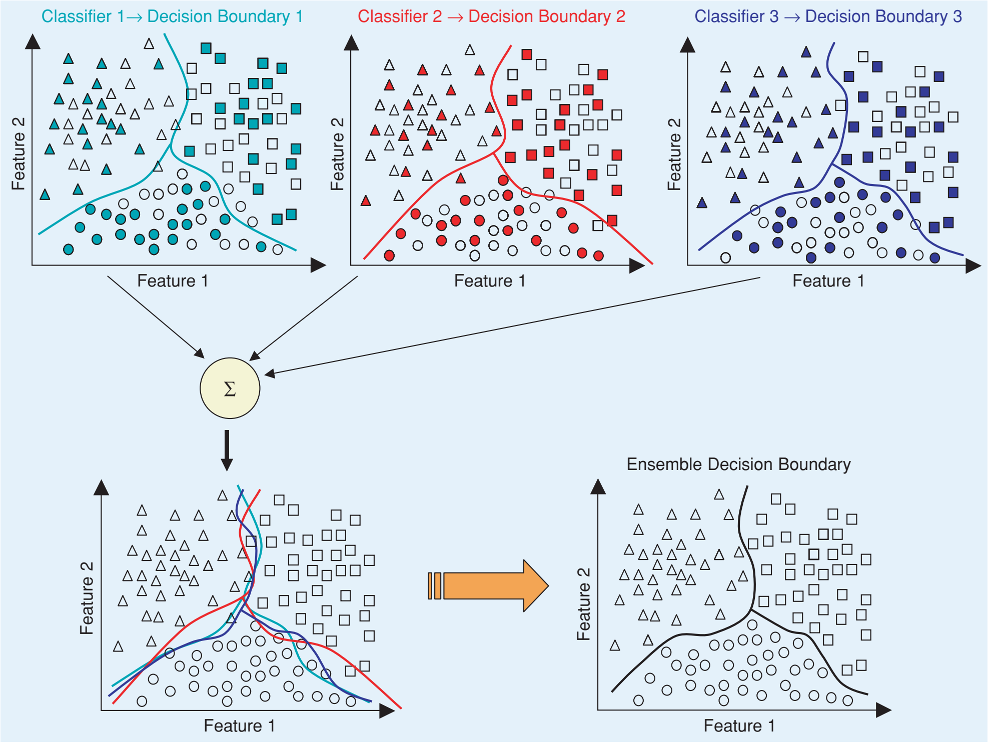
1. Data fusion
   1.1 Multiple sources of data (sensors, domain experts, etc.)
   1.2 Need to combine systematically, for example a neurologist may order several tests
       MRI scan, EEG recording, Blood test
   1.3 A single classifier cannot be used to classify data from different sources (heterogeneous features).
2. Divide and conquer
   2.1 Regardless of the amount of data, certain problems are difficult for solving by a classifier.
   2.2 Complex decision boundaries can be implemented using ensemble Learning.





Diversity measures

1. Strategy of ensemble systems
   Creation of many classifiers and combine their outputs in a such a way that combination improves upon the performance of a single classifier.
2. Requirement
   The individual classifiers must make different errors on different inputs.
3. If errors are different then strategic combination of classifiers can reduce total error.
4. Solution
   We need classifiers whose decision boundaries are adequately different from others.
   Such a set of classifiers is said to be diverse.
5. Classifier diversity can be obtained
   - Using different training datasets for training different classifiers.
   - Using unstable classifiers.
   - Using different training parameters(such as different topologies for NN).
   - Using different feature sets (such as random subspace method).
6. Reference
   G. Brown, J. Wyatt, R. Harris, and X. Yao, ''Diversity creation methods : a survey and categorization'', Information fusion, Vo. 6, pp. 5-20, 2005.


Classifier 1→ Decision Boundary 1
Classifier 2 → Decision Boundary 2
Classifier 3 → Decision Boundary 3
Feature 2
Feature 1
Σ
Ensemble Decision Boundary

Diversity measures

1. Pairwise measures (assuming that we have T classifiers)
   We can calculate $\frac{T(T-1)}{2}$ pair-wise diversity measures.

	h_j is correct	h_j is incorrect
h_i is correct	a	b
h_i is incorrect	c	d

   For a team of T classifiers, the diversity measures (d_{ij}) are averaged over all pairs

$$D_{ij} = \frac{2}{T(T-1)} \sum_{i=1}^{T-1} \sum_{j=1}^{T} d_{ij}$$

2. Pairwise diversity measures
   2.1 Correlation diversity is measured as the correlation between two classifier outputs.

$$\rho_{ij} = \frac{ad - bc}{\sqrt{a+b)(c+d)(a+c)(b+d}}$$

   When classifiers are uncorrelated, maximum diversity is obtained and $\rho = 0$.

   2.2 Q-Statistic defined as

$$Q_{ij} = (ad - bc)/(ad + bc)$$

   Q is positive when the same instances are correctly classified by both classifiers; and is negative, otherwise.
   Maximum diversity is, once again, obtained for $Q = 0$.

1. Pairwise measures (assuming that we have T classifiers)
   We can calculate $\frac{T(T-1)}{2}$ pair-wise diversity measures, and average them.

	h_j is correct	h_j is incorrect
h_i is correct	a	b
h_i is incorrect	c	d

2. Pairwise diversity measures
   2.1 Disagreement measure is the probability that the two classifiers will disagree,
   $$D_{ij} = b + c$$
   The diversity increases with the disagreement value.
   2.2 Double fault measure is the probability that both classifiers are incorrect,
   $$DF_{ij} = d.$$
   The diversity increases with the double fault value.

1. Non-pairwise measures (assuming that we have T classifiers)
   1.1 Entropy measure makes the assumption that the diversity is highest if half of the classifiers are correct, and the remaining ones are incorrect.

$$E = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{T - \left\lceil \frac{T}{2} \right\rceil}\min\{\xi_i, (T - \xi_i)\}$$

   where ξ_i is the number of classifiers that misclassify instance x_i.
   Entropy varies between 0 and 1, where 1 indicates highest diversity.

   1.2 Kohavi–Wolpert variance

$$KW = \frac{1}{NT^2}\sum_{i=1}^{N}\xi_i(T - \xi_i)$$

   Kohavi–Wolpert variance follows a similar approach to the disagreement measure.

   1.3 Measure of difficulty is

$$\theta = \frac{1}{T}\sum_{t=0}^{T}(z_t - \bar{z})$$

   where $z = \left[0, \frac{1}{T}, \frac{2}{T}, \ldots, 1\right]$ and $\bar{z}$ s mean of z.
   z is the fraction of classifiers that misclassify x_i.
   How Measure of difficulty shows the diversity?

- Comparison of different diversity measures

Name		↑ / ↓	P	S	Reference
Q-statistic	Q	(↓)	Y	Y	(Yule, 1900)
Correlation coefficient	ρ	(↓)	Y	Y	(Sneath & Sokal, 1973)
Disagreement measure	D	(↑)	Y	Y	(Ho, 1998; Skalak, 1996)
Double-fault measure	DF	(↓)	Y	N	(Giacinto & Roli, 2001)
Kohavi-Wolpert variance	kw	(↑)	N	Y	(Kohavi & Wolpert, 1996)
Interrater agreement	κ	(↓)	N	Y	(Dietterich, 2000b; Fleiss, 1981)
Entropy measure	Ent	(↑)	N	Y	(Cunningham & Carney, 2000)
Measure of difficulty	θ	(↓)	N	N	(Hansen & Salamon, 1990)
Generalised diversity	GD	(↑)	N	N	(Partridge & Krzanowski, 1997)
Coincident failure diversity	CFD	(↑)	N	N	(Partridge & Krzanowski, 1997)

Note: The arrow specifies whether diversity is greater if the measure is lower (↓) or greater (↑). 'P' stands for 'Pairwise' and 'S' stands for 'Symmetrical'.

- Reference

  L. I. Kuncheva and C. J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Machine Learning, Vol. 51, pp. 181-207, 2003.

Design of ensemble systems

- Two key components of an ensemble system
  1. Creating an ensemble by creating weak learners.
     1.1 Bagging
     1.2 Boosting
     1.3 Stacked generalization
     1.4 Mixture of experts
  2. Combination of classifiers' outputs (trainable vs. fixed rule).
     2.1 Majority Voting
     2.2 Weighted Majority Voting
     2.3 Averaging
     2.4 Error correcting codes
- What is weak learners?

Definition (Weak learner)

A weak learner does not guarantee to do better than random guessing.

In ensemble learning, a rule is needed to combine outputs of classifiers.

1. Classifier selection
    1.1 Each classifier is trained to become an expert in some local area of feature space.
    1.2 Combination of classifiers is based on the given feature vector.
    1.3 Classifier that was trained with the data closest to the vicinity of the feature vector is given the highest credit.
    1.4 One or more local classifiers can be nominated to make the decision.
2. Classifier fusion
    2.1 Each classifier is trained over the entire feature space.
    2.2 Classifier Combination involves merging the individual weak classifier design to obtain a single Strong classifier.

Building ensemble based systems

- Bootstrap Aggregating (Bagging)
  1. Create T bootstrap samples $S[1], S[2], \ldots, S[T]$.
  2. Train distinct inducer on each $S[t]$ to produce T classifiers.
  3. Classify new instance by classifier vote (majority vote).
- Application of bootstrap sampling
  1. Given set S containing N training examples
  2. Create $S[t]$ by drawing N examples at random with replacement from S
  3. $S[t]$ of size N: expected to leave out $75\% - 100\%$ of examples from S. (show it)
- Variations
  1. Random forests

     Can be created from decision trees, whose certain parameters vary randomly.
- Pasting small votes (for large datasets)
  1. RVotes : Creates the data sets randomly
  2. IVotes : Creates the data sets based on the importance of instances, easy to hard

Consider the set of k regression models

1. Each model i makes error ϵ_i on each example
2. Errors drawn from a zero-mean multivariate normal with variance $\mathbb{E}[\epsilon_i^2] = v$ and covariance $\mathbb{E}[\epsilon_i \epsilon_j] = c$
3. Error of average prediction of all ensemble models: $\frac{1}{k}\sum_i \epsilon_i$
4. Expected squared error of ensemble prediction is

$$\mathbb{E}\left[\frac{1}{k}\sum_i \epsilon_i\right]^2 = \frac{1}{k}v + \frac{k-1}{k}c$$

5. If errors are perfectly correlated, $c = v$, and mean squared error reduces to v, so model averaging does not help.
6. If errors are perfectly uncorrelated and $c = 0$, expected squared error of ensemble is only $\frac{v}{k}$ and Ensemble error decreases linearly with ensemble size

1. Let $S = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$ be the training set, where $\mathbf{x}_i \in \mathbb{R}^D$.
2. Random Forests algorithm follows thefollowing steps:
   2.1 Create m bagged samples of size n, with $n < N$.
   2.2 Train a decision tree with each of the m bagged data sets as input using the following procedure.
      2.2.1 When doing a node split, don't explore all features in S.
      2.2.2 Randomly select a smaller number, $d \ll D$ features, from all the features in S.
      2.2.3 Then pick the best split using impurity measures, like Gini, Impurity or Entropy.
   2.3 Aggregate the results of the individual decision trees into a single output.
      2.3.1 Average the values for each observation, produced by each tree, if you're working on a Regression task.
      2.3.2 Do a majority vote across all trees, for each observation, if you're working on a Classification task.

- Schapire proved that a weak learner can be turned into a strong learner that generates a classifier that can correctly classify all but an arbitrarily small fraction of the instances.
- In boosting, the training data are ordered from easy to hard. Easy samples are classified first, and hard samples are classified later.
- Boosting algorithm
  1. Create the first classifier same as Bagging
  2. The second classifier is trained on training data only half of which is correctly classified by the first one and the other half is misclassified.
  3. The third one is trained with data that two first disagree.
- Variations
  1. AdaBoost.M1
  2. AdaBoost.R
- Reference

  Robert E. Schapire, The strength of weak learnability, Machine Learning, Vol. 5, pp. 197-227 (1990).

$\textsc{AdaBoost}(S = ((x_1, y_1), \ldots, (x_m, y_m)))$

for $i \leftarrow 1$ **to** m **do**

  $D_1(i) \leftarrow \frac{1}{m}$

for $t \leftarrow 1$ **to** T **do**

  $h_t \leftarrow$ base classifier in H with small error $\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$

  $\alpha_t \leftarrow \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$

  $Z_t \leftarrow 2[\epsilon_t(1-\epsilon_t)]^{\frac{1}{2}}$ ▷ normalization factor

  for $i \leftarrow 1$ **to** m **do**

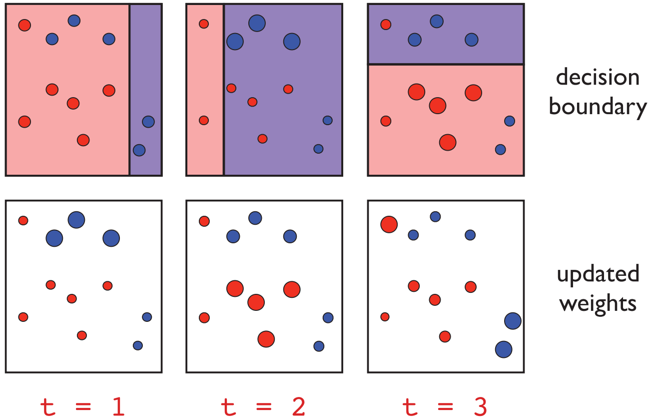
    $D_{t+1}(i) \leftarrow \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$

$g \leftarrow \sum_{t=1}^{T} \alpha_t h_t$

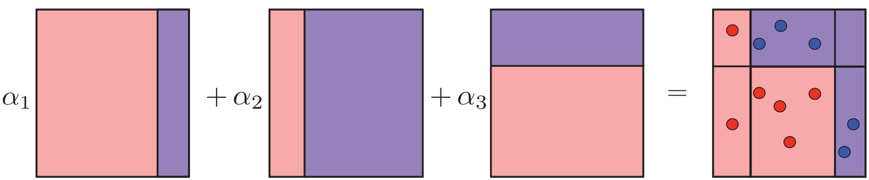
return $h = \operatorname{sgn}(g)$

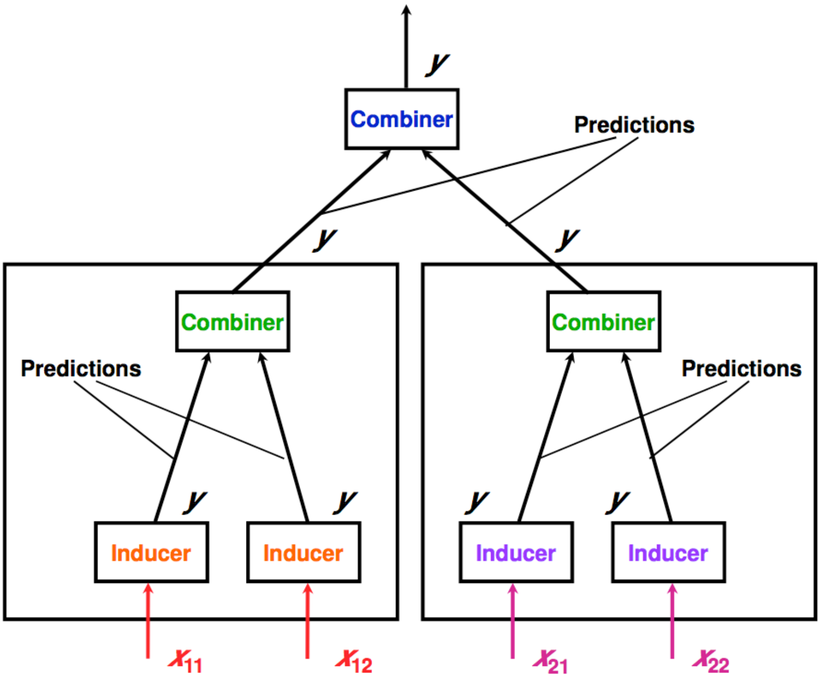
Freund, Yoav; Schapire, Robert E.A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences, Vol. 55, pp. 119–139 (1997).

AdaBoost (example)

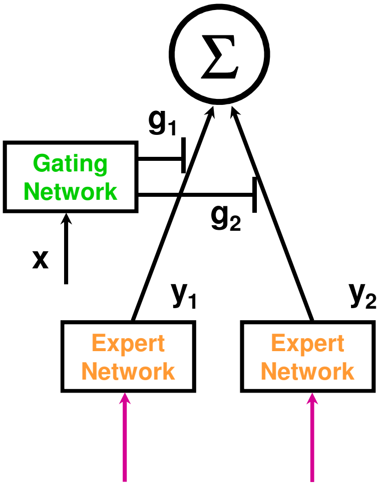


(a)

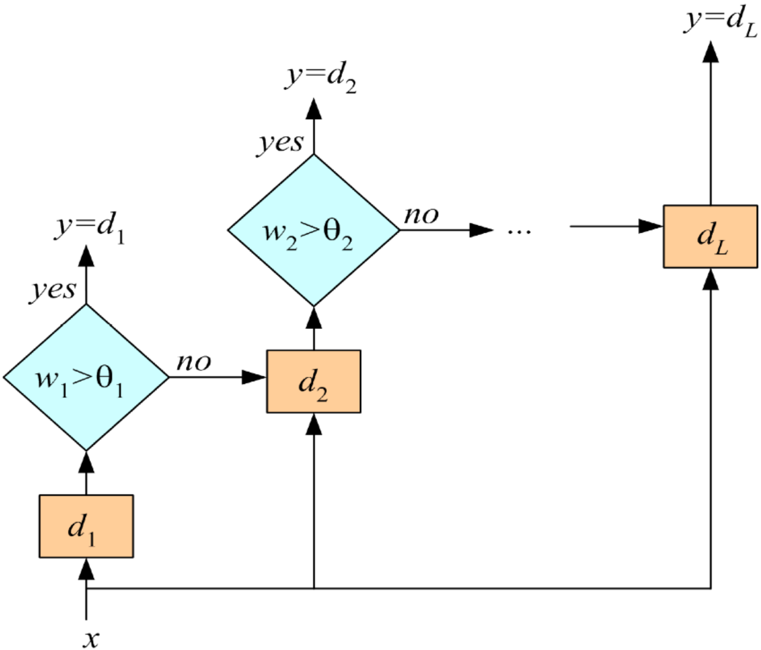



y
Combiner
Predictions
y
y
Combiner
Combiner
Predictions
Predictions
y
y
y
y
Inducer
Inducer
Inducer
Inducer
x11
x12
x21
x22

1. Train multiple learners
   1.1 Each uses subsample of S
   1.2 May be ANN, decision tree, etc.
2. Gating Network usually is NN



Cascade learners in order of complexity



Reading

1. Sections 14.1, 14.2 & 14.3 of Pattern Recognition and Machine Learning Book (Bishop 2006).
2. Robi Polikar, Ensemble based system in decision making, IEEE Circuits and Systems Magazine, Vol. 6, No. 3, pp. 21 - 45 (2006).
3. T. G. Dietterich, Machine Learning Research: four current directions, AI Magazine. 18(4), 97-136 (1997).
4. T. G. Dietterich, Ensemble Methods in Machine Learning, Lecture Notes in Computer Science, Vol. 1857, pp 1-15 (2000).
5. Ron Meir, Gunnar Ratsch, An introduction to Boosting and Leveraging, Lecture Notes in Computer Science, Vol. 2600, pp 118-183 (2003).
6. David Opitz, Richard Maclin, Popular Ensemble Methods: An Empirical Study, journal of artificial intelligence research, pp. 169-198 (1999).
7. L.I. Kuncheva, Combining Pattern Classifiers, Methods and Algorithms, Second edition. New York, NY: Wiley Interscience, 2014.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag.

Questions?