Lecture 17: Ensemble Learning

COMP90049

Semester 2, 2021

Qiuhong Ke, CIS

Acknowledgement: Jeremy Nicholson, Tim Baldwin & Karin Verspoor

Classification

So far:

- Classification algorithms in isolation
- Training and testing one classifier
- Remedies for Overfitting and underfitting

Today:

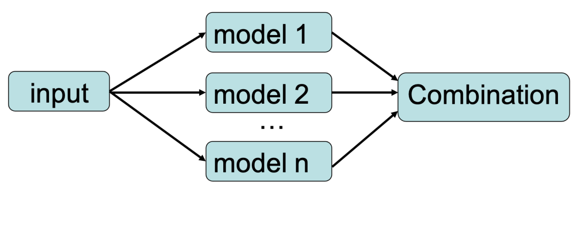
- Introduction of Ensemble learning
- Stacking
- Bagging
- Boosting

THE UNIVERSITY OF MELBOURNE

Introduction of Ensemble learning

What is Ensemble Learning

Ensemble learning (aka. Classifier combination): constructs a set of base classifiers from a given set of training data and aggregates the outputs (e.g., using majority voting) into a single meta-classifier.



Approaches for Ensemble Learning

- **Instance manipulation**: generate multiple training datasets through sampling, and train a base classifier over each dataset
- **Feature manipulation**: generate multiple training datasets through different feature subsets, and train a base classifier over each dataset
- **Class label manipulation**: generate multiple training datasets by manipulating the class labels in a reversible manner
- **Algorithm manipulation**: semi-randomly “tweak” internal parameters within a given algorithm to generate multiple base classifiers over a given dataset

Why Ensemble Learning

- **Intuition 1**: the combination of lots of weak classifiers can be at least as good as one strong classifier
- **Intuition 2**: the combination of a selection of strong classifiers is (usually) at least as good as the best of the base classifiers

When does ensemble learning work? I

- The base classifiers should not make the same mistakes
- The base classifiers are reasonably accurate

	t_1	t_2	t_3
C_1	√	√	x
C_2	x	√	√
C_3	√	x	√
C^*	√	√	√

	t_1	t_2	t_3
C_1	√	√	x
C_2	√	√	x
C_3	√	√	x
C^*	√	√	x

	t_1	t_2	t_3
C_1	√	x	x
C_2	x	√	x
C_3	x	x	√
C^*	x	x	x

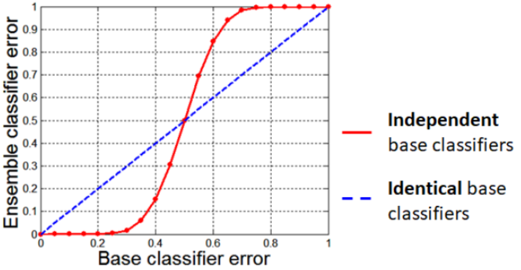
When does ensemble learning work? II

- Given 25 binary base classifiers, each with an error rate of $\epsilon = 0.35$.
- Ensemble by majority voting
  - if the base classifiers are identical, after ensemble, $\epsilon = 0.35$.
  - If the base classifiers are independent, after ensemble,

$$\sum_{i=13}^{25} \binom{25}{i} \epsilon^i (1-\epsilon)^{25-i} \approx 0.06$$

When does ensemble learning work? II

- When does ensemble learning work?



Quiz

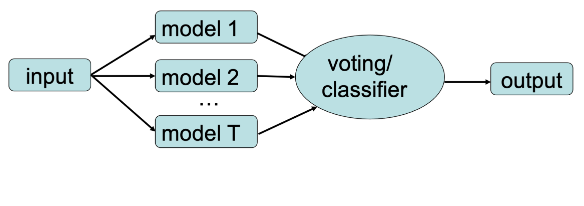
Which of the following statement(s) are TRUE about ensemble learning?

- (a) An ensemble of classifiers may not be able to outperform any of its individual base learners.
- (b) Combining significantly diverse base learners (suppose each produces meaningful predictions) typically yields bad results.

Stacking

Stacking

- **Intuition**: “smooth” errors over a range of algorithms with different biases
- **Method**: use different algorithms to train multiple base classifiers on the dataset.



- **Inputs for second-level classifier (meta-learner):** use base classifiers to generate predictions on unseen samples (using cross-validation).

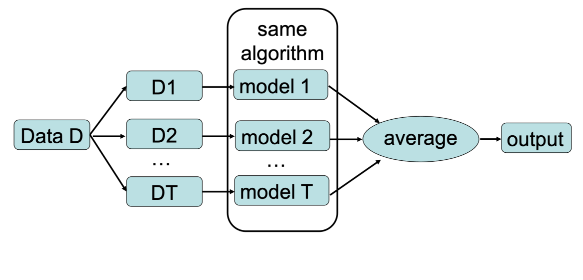
Stacking II

- Mathematically simple but computationally expensive method
- Able to combine heterogeneous classifiers with varying performance
- Generally, stacking results in as good or better results than the best of the base classifiers

Bagging

Bagging I

- **Intuition**: Average multiple models can lower the model variance.
- **Method**: Create multiple new training sets for training multiple classifiers base on the same algorithm and average the predictions.



Bagging II

Dataset generation: randomly sample the original dataset (N instances) N times, with replacement. Any individual instance is absent with probability $(1 - \frac{1}{N})^N$

Example:

- Original dataset:

1	2	3	4	5	6	7	8	9	10

- Bootstrap Samples

7	2	6	7	5	4	8	8	1	10

1	3	8	10	3	5	8	10	1	9

2	9	4	2	7	9	3	10	1	10

$\vdots$

Bagging III

- Possibility to parallelise computation of individual base classifiers
- Highly effective over noisy datasets (outliers may vanish)
- Generally produces the best results on unstable models that have high variance and low bias

Random Forest I

- A “Random Forest” is an ensemble of Random Trees (many trees = forest)
- A “Random Tree” is a Decision Tree where at each node, only some of the possible attributes are considered
- Use random trees instead of decision trees to increase diversity of base classifier

Random Forest II

Practical Properties of Random Forests:

- Embarrassingly parallelisable
- Robust to overfitting
- Interpretability sacrificed

Quiz

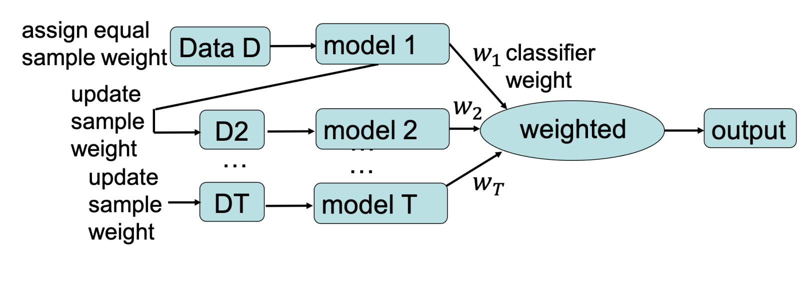
Which of the following statement(s) are TRUE about Random Forest?

- (a) Random Forest provides higher interpretability over the logic behind the predictions than a single random tree.
- (b) Random Forest adopts both feature manipulation and instance manipulation approaches.
- (c) Random Forest minimizes the bias by having multiple random trees trained on different versions of the dataset.

Boosting

Boosting I

- **Intuition**: Build a strong model from several weak models to reduce model bias.
- **Method**: Iteratively change the weights of training instances to train next base classifier and combine the base classifiers via weighted voting



Boosting II

- Original dataset:

1	2	3	4	5	6	7	8	9	10

- Boosting samples:

Iteration 1:	7	2	6	7	5	4	8	8	1	10
Iteration 2:	1	3	8	4	3	5	4	10	1	4
Iteration 3:	4	9	4	2	4	4	3	10	1	4

$\vdots$

AdaBoost I

- Input: Training instances $(x_j, y_j) | j = 1, 2, \ldots, N$
- Initial equal sample weights $w_j^{(0)} = \frac{1}{N} | j = 1, 2, \ldots, N$
- For $i = 1 \cdots T$
    - Construct classifier C_i in iteration i:
        - apply sample weights to the loss or
        - use the weights to re-sample data to train model
    - Calculate weight of the classifier α_i
    - Update the sample weights
- Final classification via weighted voting: multiply vote of each classifier with its weight.

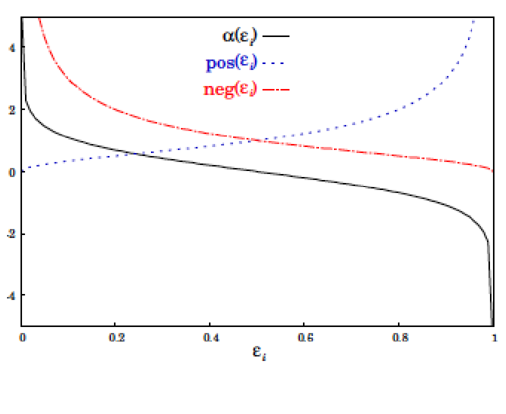
AdaBoost II

- Error rate for C_i:

$$\epsilon_i = \sum_{j=1}^{N} w_j^{(i)} \delta(C_i(x_j) \neq y_j)$$

- "Importance" of C_i (i.e. the weight associated with the classifiers' votes):

$$\alpha_i = \frac{1}{2} \log_e \frac{1 - \epsilon_i}{\epsilon_i}$$



AdaBoost III

- If $\alpha_i > 0$, adjust weights for instance j ($i > 0$):

$$w_j^{(i+1)} = w_j^{(i)} \times \begin{cases} e^{-\alpha_i} & \text{if} \quad C_i(x_j) = y_j \\ e^{\alpha_i} & \text{if} \quad C_i(x_j) \neq y_j \end{cases}$$

$$Z_i = \sum_{j=1}^{N} w_j^{(i+1)}$$

$$w_j^{(i+1)} = w_j^{(i+1)} / Z_i$$

AdaBoost IV

- Classification:

$$C^*(x) = \underset{y}{argmax} \sum_{i=1}^{T} \alpha_i \delta(C_i(x) = y)$$

- Base classification algorithm: decision stumps (OneR) or decision trees

Quiz

Which of the following statement(s) are TRUE about Boosting?

- (a) Boosting adopts feature manipulation approach to train multiple base learners
- (b) Boosting assigns higher weights to better-performing base learners
- (c) Boosting iteratively learns base learners while emphasizing the samples that can be easily classified

Bagging vs. Boosting

Bagging/Random Forests	Boosting/AdaBoost
Builds base models in parallel	Builds base models sequentially
Parallel sampling: Resamples data points with replacement	Iterative sampling: Reweights data points (modifies their distribution)
Base classifiers have the same weight	Base classifiers have the different weight
Reduce variance	Reduce bias
Not prone to overfitting	Prone to overfitting

THE UNIVERSITY OF MELBOURNE

Summary

Summary

- What is classifier combination?
- What is bagging and what is the basic thinking behind it?
- What is boosting and what is the basic thinking behind it?
- What is stacking and what is the basic thinking behind it?
- How do bagging and boosting compare?

Quiz

What are the techniques that use instance manipulation approach to combine classifiers?

- (a) Bagging
- (b) Boosting
- (c) Random Forest
- (d) Stacking

References

- Leo Breiman. Random forests. Machine Learning, 45(1):5–32, 2001.
- Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Introduction to Data Mining. Addison Wesley, 2006.
- Ian H. Witten and Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco, USA, second edition, 2005.