Boosting Algorithms: *AdaBoost*

Tutorial for ELEC-E7260 Machine Learning for Mobile and Pervasive System

January 18th, 2019

Viet Ba Hirvola
Shuhao Que
Yunfei Xue

Ensemble methods: Boosting

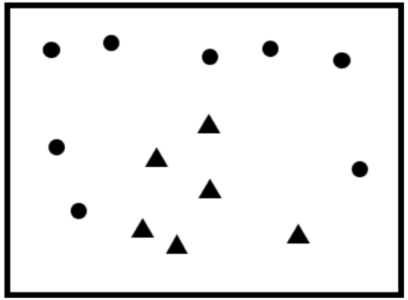
- **Ensemble methods** - use multiple learning algorithms into a single predictive model to obtain improved predictive performance than could be achieved from each separately.
    - **Stacking** - improves predictions
    - **Bagging** - decreases variance
    - **Boosting** - decreases bias
- **Boosting algorithms** - combines multiple weak learners to create a stronger model.
    - **Weak learner** - a classifier that produces prediction slightly better than random guessing.
    - Weak learners are trained **sequentially**.
- Examples of boosting methods: **AdaBoost**, Gradient Boosting

AdaBoost

- Stands for *Adaptive Boosting*
- Combines weak learners **linearly**
- Most commonly used to boost **decision trees** with one level, aka **decision stumps**
- Iteratively adapts to the errors made by weak learners in previous iterations.
- **Re-weighting** scheme:
    - Higher weight is assigned to incorrectly classified data points
    - Lower weight assigned to correctly classified data points

Initialisation

Training data



Initialise the **weight matrix** of data points as uniform distribution, i.e. assigning same weight to all data points:

$$\alpha_i = 1 / N$$

where N is the number of data points.

Iterations

For $i = 1, \ldots, T$ or until low enough error is achieved:

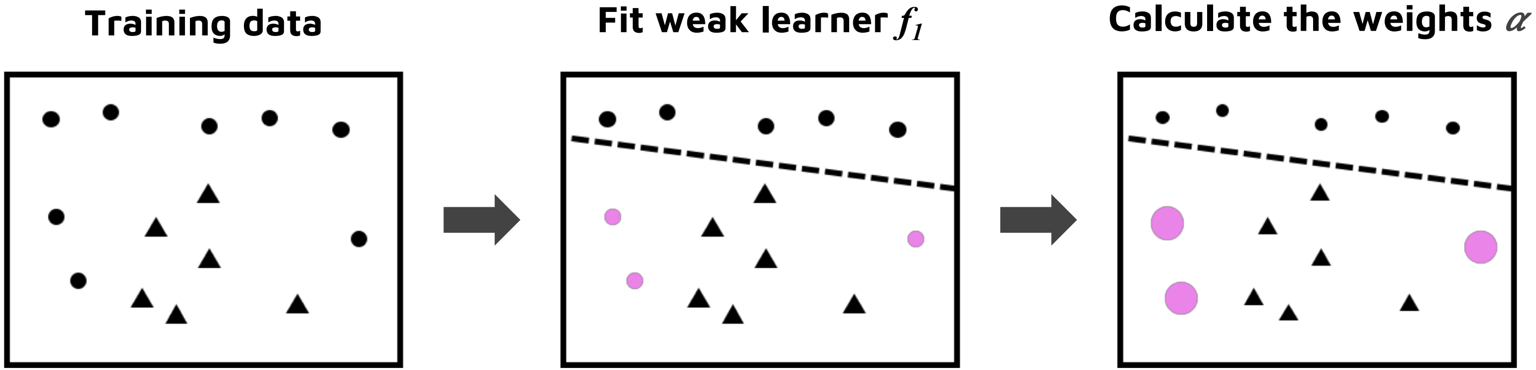
- Fit weak learner $f_t(x)$ to data points with data point weights α_i

- Compute weak learner's weight $\quad w_t = \frac{1}{2}\ln\left(\frac{1 - weighted\_error(f_t)}{weighted\_error(f_t)}\right)$

- Recompute data point weights $\quad \alpha_i = \begin{cases} \alpha_i e^{-w_t} & if\ f_t(x_i) = y_i \\ \alpha_i e^{w_t} & if\ f_t(x_i) \neq y_i \end{cases}$

- Normalize weights α_i

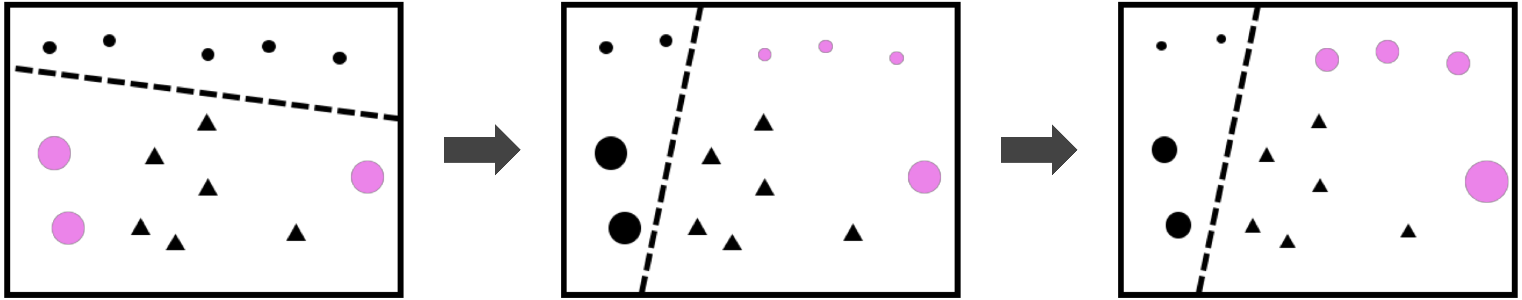
Iteration 1



- initially all data points have the same weight *1/N*
- whatever the class is correctly classified will be given less weights in the next iteration, and higher weights for misclassified classes

Iteration 2

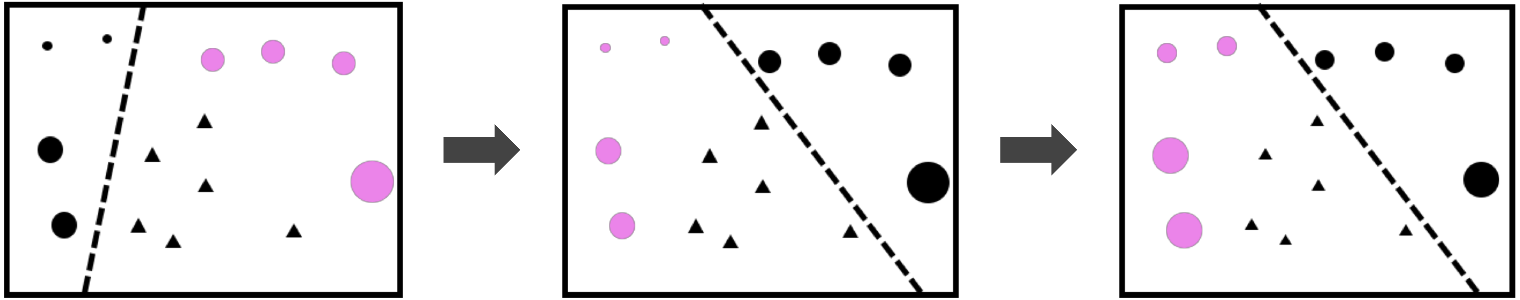
Weights after 1st iteration | **Fit weak learner f_2** | **Calculate the weights α**



- Weak learner forms a decision boundary which classifies better the data points with higher weights

Iteration 3

Weights after 2nd iteration | **Fit weak learner f_3** | **Calculate the weights α**

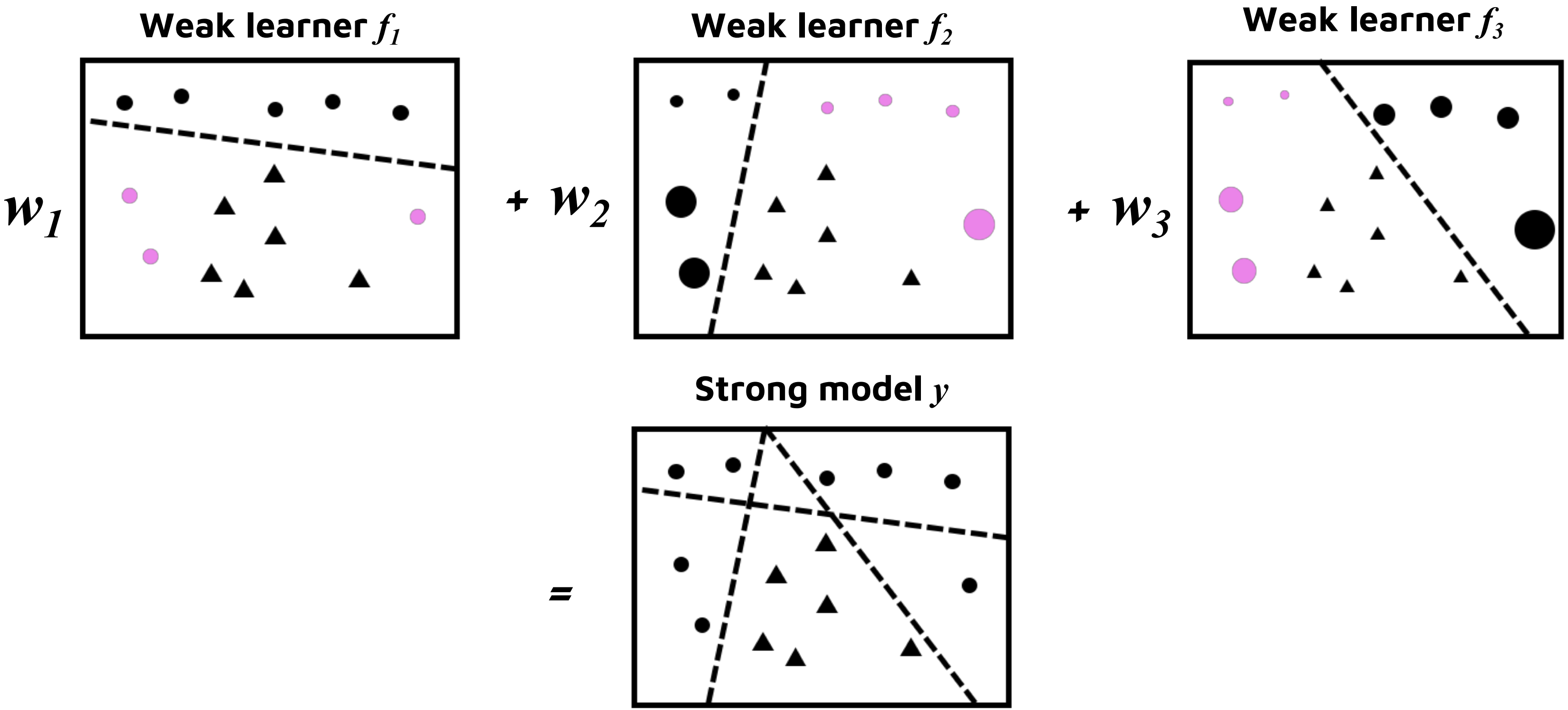


… continue iterating until either:

- Sufficiently low training error is achieved (with enough iterations the algorithm can reach 100% accuracy)
- A predefined number of weak learners was added

Final model

$$\hat{y} = sign\left(\sum_{t=1}^{T} w_t f_t(x)\right)$$



Why use AdaBoost?

- Needs only a simple classifier as a weak learner
- Can achieve prediction results similar to powerful classifiers
- Can combine with any learning algorithm
- Requires little parameter tuning (usually only T)
- Selects only features known to improve predictive power
  - Relatively simple classifier, easy to program
  - Reduced dimensionality
  - Improved execution time
- Has been extended to problems beyond binary classification

However…

- Performance depends on the input data and weak learner
- Can fail if the weak classifiers are
  - Too complex (overfits)
  - Too weak (underfits)
- Sensitive to noisy data and outliers
- The ensemble is optimised based on the currently known estimates, not globally

Summary

- AdaBoost (i.e. Adaptive boosting) is one of the most popular and powerful **ensemble methods**
- Shifts algorithm's focus on the data points that are erroneous by **adapting weights**
- Simple to implement... depending on the weak learner you choose
- Vulnerable to noisy data

References

- Freund, Y., & Schapire, R. E. (1997). *A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences*, 55(1), 119-139.
- Avinash Navlani. (2018). *AdaBoost Classifier in Python*. DataCamp Tutorials.
- Brownlee Jason. (2016). *Boosting and AdaBoost for Machine Learning*. Machine Learning Mastery.
- Wyner, A. J., Olson, M., Bleich, J., & Mease, D. (2017). Explaining the success of adaboost and random forests as interpolating classifiers. The Journal of Machine Learning Research, 18(1), 1558-1590.
- Jung Haebichan. (2018). *Adaboost for Dummies: Breaking Down the Math (and its Equations) into Simple Terms*. Towards Data Science.
- Matas Jiri & Sochman Jan. (2009). *AdaBoost*. Centre for Machine Perception, Czech Technical University. C4B Computer Vision Lectures Michaelmas.
- Tommi Jaakkola. (2012). Introduction To Machine Learning - The AdaBoost algorithm. Source: http://people.csail.mit.edu/dsontag/courses/ml12/slides/lecture13.pdf