Discussion 3
Probabilistic Machine Learning,Fall 2016

1. Boost

Concepts

(a) In AdaBoost, if a weak learner gets example i wrong, then on the next round, the weight of example i will be:

Solution Higher, so that the next weak learner is more likely to get example correct

(b) AdaBoost will eventually give zero training error regardless of the type of weak classifier it uses, provided enough iterations are performed.

Solution False. Not if the data in the training set cannot be separated by a linear combination of the specific type of weak classifiers we are using. For example consider the XOR example with decision stumps as weak classifiers. No matter how many iterations are performed zero training error will not be achieved.

Practice

1. Consider the following 2-class binary problem in Fig. 1. Using the error and weight update rule of the discrete AdaBoost, answer the following question

(a) What are the first 2 decision stumps, and what are their corresponding thresholds?

Solution First decision stump: $\phi_1(x) = 1, \text{if } x > -2, \quad \text{otherwise} - 1$

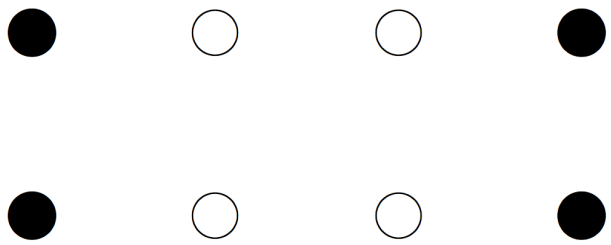
Second decision stump: $\phi_2(x) = 1, \text{if } x < 2, \quad \text{otherwise} - 1$

(b) Are these sufficient to obtain perfect classification?

Solution Not with Discrete AdaBoost. The final vote is obtained by linear combination of the weak learners. Regardless of the choice of weights assigned to each learner, the vote for the negative (black) samples on one of the two sides will be positive (the classifier function will be equal to 0 if all the weak learner weights are equal for instance). You need at minimum 3 weak learners to correctly classify this problem so that the third weight can play the role of a bias (a decision stump with all the points on one side of it, with a weight equal to the bias).

2. Selecting the proper model of weak classifier is often a very important step when tackling new classification problems. Fig. 2 present a number of such problems; for each dataset: a) Propose a weak learner that will excel in classifying the data b) Estimate the number of weak learners necessary to perform

$$C_{y=-1} = \{(-3,-1),(-3,1),(3,-1),(3,1)\}$$
$$C_{y=1} = \{(-1,-1),(-1,1),(1,-1),(1,1)\}$$



Training

Figure 1: Toy data.

the task optimally Using the error and weight update rule of the discrete AdaBoost, answer the following question

Solution :

(a) Figure 1: a) The easiest solution would be using random rectangles or decision stumps, as they will perfectly model the straight boundaries between the two classes. However, if the number of weak learners generated is too small, no rectangles might fit perfectly the boundaries of each 'cell'. In this case a "Random Circles" might better do the job as it is more robust to the initial random generation

b) Assuming that a sufficient number of weak learners have been generated, the system could perfectly classify the example using 12 random rectangles (there are 12 red squares and 13 white square, if you put equal weight one for each weak classifier on the red squares, there is no need for a bias w.l.) or 9 decision stumps (4 for each axis and one bias.)

(b) Figure 2: a) Random Circles are perfect for this task as they perfectly model the radial expansion of the classes

b) Three boundaries need to be modeled and 3 weak learners are sufficient to properly classify this data (on the contrary to Exercise 1, there is already an odd number of weak learner therefore no other one is required to add a bias.)

(c) Figure 3: a) Random Projection would allow to 'cut the edges' of the negative (white) class while maintaining perfect classification of the positive class. Random rectangles are another possible choice.

b) A minimum of 3 weak learners would be necessary to solve correctly this problem (see Exercise 1). One random rectangle is also enough.

(d) Figure 4: a) Random rectangles would be a good starting point, if sufficient examples are generated. Otherwise, Decision stumps are well fitted to this problem.

b) A single random rectangle, if placed perfectly would do the job. Alternatively, 5 decision stumps would also suffice (2 to classify horizontally, 2 to classify vertically, 1 bias, see Exercise 1).

3. AdaBoost can be used in two ways. The first way is to combine a set of weak classifiers (such as stumps) where the set is determined before we run AdaBoost. The second way to run AdaBoost is as a meta algorithm on top of a weak learning algorithm like CART or C4.5. In that case, we can never actually

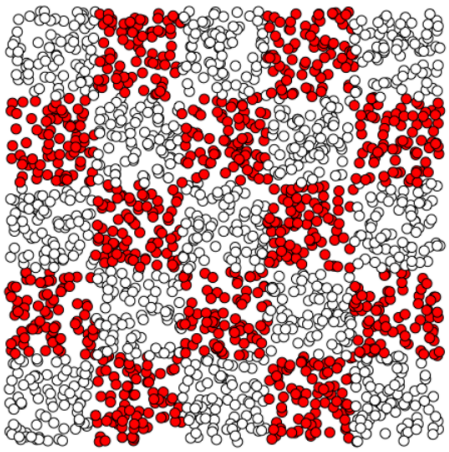


Figure 1: Checkerboard

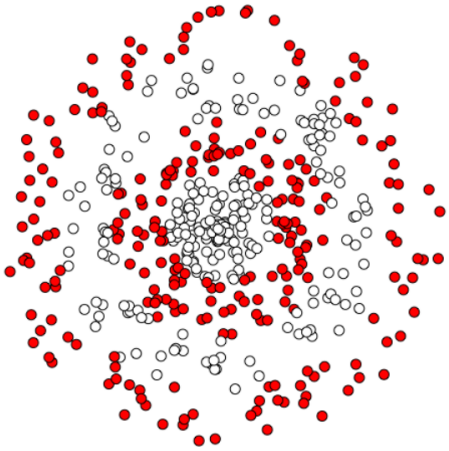


Figure 2: Concentric rings

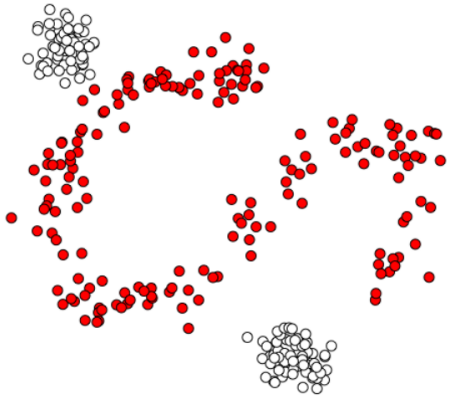


Figure 3: The serpent

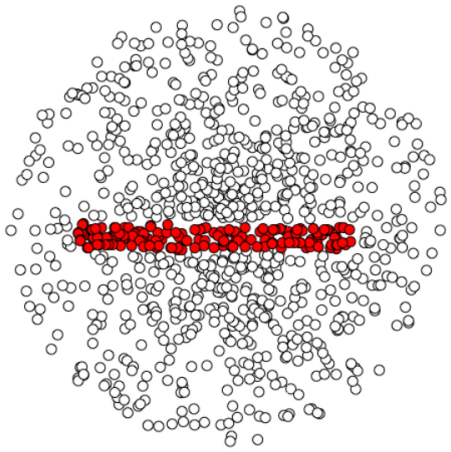


Figure 4: Stop sign

Training

Figure 2: Toy data.

enumerate all the weak classifiers. Look through all the steps of the AdaBoost algorithm and show that no step in the algorithm requires us to actually enumerate the set of weak classifiers. This means we never need to evaluate the matrix of margins $\mathbf{M}$. You will need to know that we do not actually need to obtain the best possible weak classifier (the argmax) in each round of AdaBoost in practice. It is sufficient to get a good weak classifier.

Solution: For a data point x, we have

$$f(x) = \sum_{j=1}^{n} \lambda_{j,t} h_j(x) \tag{1}$$

$$= \sum_{j=1}^{n} \left(\sum_{t=1}^{T} \alpha_t \mathbf{1}_{[j_t=j]} \right) h_j(x) = \sum_{t=1}^{T} \alpha_t \left(\sum_{j=1}^{n} \mathbf{1}_{[j_t=j]} h_j(x) \right) \tag{2}$$

$$= \sum_{t=1}^{T} \alpha_t h_{j_t}(x) \tag{3}$$

We can calculate $f(x)$ based on either Eq. 1 or Eq. 3. If Eq. 1 is chosen, we have to compute $\mathbf{M}$. In practice, we can alternatively choose Eq. 3 to obtain $f(x)$, where we avoid computing $\mathbf{M}$.

4. Assume the weak learning assumption holds. Asymptotically, as AdaBoost iterates over rounds, it is possible to determine what the values of $f(\mathbf{x}_i)$ will converge to?

Solution: Note that because of the weak learning assumption, the data is separable, and α_t is always bounded below by $\alpha_{\min}$, where $\alpha_{\min}$ is a function of the minimum edge $\gamma_{\min}$ that comes from the weak learning assumption. This means α_t is always fairly large. This means some of the λ's grow by at least a constant at every iteration.

$f(x_i)$ is a sum over λ's of the h_j's, which take on values -1 or 1. This means that $f(x_i)$'s get larger and larger, and since the data are correctly classified, $f(x_i)$ for the positive examples goes to infinity and $f(x_i)$ for the negative examples goes to negative infinity.

A simple proof can largely depend on the Theorem that *the training error of Adaboost decays exponentially fast*. For the weak learning assumption, the each classifier performs better than "random guessing": $\epsilon_t = \frac{1}{2} - \gamma_t$, with $\gamma_t > \gamma_{\text{WLA}}$, the Theorem shows that:

$$LH \triangleq R_{\text{train}}(\lambda_T) \leq \exp(-2\sum_{t=1}^{T} \gamma_t^2) \leq \exp(-2\sum_{t=1}^{T} \gamma_{\text{WLA}}^2 T) \triangleq RH \ ,$$

where $R_{\text{train}}(\lambda_T) = \frac{1}{m} \sum_{i=1}^{m} \exp(-(\mathbf{M}\lambda_T)_i)$, and $(\mathbf{M}\lambda_t)_i = y_i f(x_i)$.

When $T \to +\infty$, we have $RH \to 0$, therefore $LH \leq 0$. Note that every term in LH is nonnegative, we need $\exp(-(\mathbf{M}\lambda_t)_i) \to 0$ or $y_i f(x_i) \to +\infty$ to satisfy this. This means, for x_i with label $y_i = 1$, its $f(x_i) \to +\infty$, and for x_i with label $y_i = -1$, its $f(x_i) \to -\infty$.