Course Name: Machine Learning and Deep learning - Fundamentals and Applications

Professor Name: Prof. M. K. Bhuyan

Department Name: Electronics and Electrical Engineering

Institute Name: Indian Institute of Technology, Guwahati

Week-6

Lecture-25

Welcome to NPTEL online course on machine learning and deep learning fundamentals and applications. In my last class, I explained the concept of ensemble classifiers. I highlighted three important principles. The first I explained the concept of stacking. After this, I explained the concept of bagging and finally, I explained the concept of boosting. In case of the boosting, the principle is from number of weak learners, I can make a strong classifier.

The samples which are not correctly classified are given maximum importance. So that means, I can give the importance to the samples which are not correctly classified and that is the fundamental concept of the boosting and one important algorithm is adaboost classifier. Today, I am going to discuss the concept of boosting and also I will explain the concept of adaboost classifier. So let us begin this class.

So in my last class, I have shown this figure and this is regarding boosting. So this is the boosting. So here you can see I am considering a training dataset and with the help of this training dataset, I am training the model, the model number 1. The model number 1 cannot perfectly classify all the samples.

The samples which are misclassified that we are considering and I am giving the importance to the samples which are misclassified by the model number 1.

So that is why we are considering weighted sample 1 that means I am considering another training dataset considering weighted samples and based on these weighted samples, I am training the model number 2 that is the another classifier and model number 2 also it cannot perfectly classify all the samples. So that is why I have to give the importance to the samples which are not correctly classified by the model number 2. So that means weighted sample 2. So I have another training dataset considering the weighted sample 2 and with the help of this new training dataset, I am training the model 3.

So like this I can consider number of weak learners.

So here this model number 1, model number 2, model number 3, these I can consider as weak classifiers. So these are weak classifiers, the model number 1, model number 2, these are weak classifiers. And finally the output from model 1, model 2, model 3 from all the models and they are combined and after combination I am getting the output. So that means this is the strong classifier.

So this is the fundamental concept of boosting that means from number of weak learners, I am making a strong classifier and you can understand the concept of the weighted samples.

The samples which are not correctly classified, I am giving the importance and they are weighted. So this is the fundamental concept of the boosting. So let us write the algorithm for the boosting. So move to the next slide. So what is this boosting?

So first step is number 1, train a weak model on some training data.

So first we are considering a weak model on some training data. After this, I am computing error, so that means compute the error of the model on each training samples. example. After this give higher importance to the examples on which the model mistakes. So that means I am giving the importance to the examples which are not correctly classified by the model.

After this what I can do? Re-train the model using importance weighted training examples and after this go back to step 2. So step 2 is this. So this is the algorithm for boosting. So this is the fundamental algorithm for boosting. Now let us discuss the concept of another algorithm related to the boosting and that is the Adaboost classifier.

So Adaboost is a boosting technique and with the help of this Adaboost classifier, we can do perfect classification. So let us discuss the concept of the Adaboost algorithm. So now I am discussing Adaboost. This is a boosting algorithm.

So first suppose I have some training samples given training data.

So suppose I have the training data (x_1, y_1). So N number of training data x_n that is the input and output is y_n with this $y_n \in \{-1, +1\} \forall n$. So output may be minus 1 or plus 1. So the given training dataset and you can see I have the training dataset (x_1, y_1), (x_2, y_2) like this and output is y_n and it may be minus 1 or plus 1.

After this in the second step initialize weight of each example.

So the example is (x_n, y_n) and I am initializing the weights that is I am considering equal weights. So I am initializing the weights and I am considering equal weights. So it is $D_1(n) = \frac{1}{N} \forall n$. So initialization of the weights for each example. After this we start iteration.

So for round $t = 1:T$ so we are starting the iteration. Learn a weak classifier so suppose the weak classifier is $h_t(x_\cdot)$ and output already I told you it is it may be minus 1 or plus 1. Using training data weighted as per D_t. So now we are considering the learning of a weak classifier using the training data weighted as per D_t. After this we have to determine the error because already I told you that all the samples may not be correctly classified.

So that is why we have to determine the error. So compute the weighted fraction of errors of the weak classifier. Weak classifier is the h_t on this training data. So we are computing the error. So what is the error? $\epsilon_t = \sum_{n=1}^{N} D_t(n) \forall [h_t(x_n) \neq y_n]$

So only for this case we are determining the error. So error is nothing but $\sum_{n=1}^{N} D_t(n)$. And we are considering the error for the samples which are not correctly classified. That is $h_t(x_n) \neq y_n$. So that means we have computed the error.

After computation of the error what I need to do so move to the next slide. Next point is we have to set importance of h_i. So how to set the importance because the samples which are misclassified we have to give the weightage maximum weightage the importance.

So this $\alpha_t = \frac{1}{2} log(\frac{1-\epsilon_t}{\epsilon_t})$.

So we are considering the set the importance like this. So it is the importance. So this value actually gets larger as the error gets smaller. So we are setting the importance. So after this I have to update the weight of each of the samples.

So how to update the weights of the samples. So update the weight of each sample because already I told you the samples which are not correctly classified I have to give the importance. So in the next iteration I am getting D_{t+1} and how to give the importance to the misclassified samples. So $D_{t+1}(n) = D_t(n) exp(-alpha_t)$ if $h_t(x_{\cdot n}) = y_n$. So that means the samples which are correctly classified I have to decrease the weight.

So this actually the meaning is corresponding to this and this is actually the correct prediction. So since it is a correct prediction what I need to do decrease weight. Now I let us consider the samples which are not correctly classified by the weak classifier. So

$D_{t+1}(n) = D_t(n)exp(alpha_t)$ if $h_t(x_n) \neq y_n$.

So I am giving the importance. So what is the meaning of this? The meaning of this is incorrect prediction and corresponding to this what I am doing now increasing the weight. So that is the fundamental principle of the adaboost filter. So these two conditions I can combine like this $D_{t+1}(n) = D_t(n)exp(-alpha_t)$. So I can write like this D_t. So D_{t+1} I can determine like this that is the training data set in the next iteration.

So I am getting this one and also I have to do normalization. So how to do the normalization? Normalize D_{t+1} so that it sums to 1.

So that means D_{t+1} I can write like this $D_{t+1} = \frac{D_{t+1}(n)}{\sum_{m=1}^{n} D_{t+1}(m)}$.

So just I am doing the normalization. So after the normalization I have to consider the output because I have to combine all the outputs of the weak learners to get the strong classifier. So what is the output? Output is nothing but I can consider $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$. So this is the final output I am getting. So signum function is considered and that means I am combining all the weak classifiers to get the strong classifier and that is actually the boosted.

So this is actually the boosted output. So this is the boosted output. So I am getting the output like this. So this is the fundamental concept of the AdaBoost algorithm. Now let us consider one illustration or I can consider one example of the AdaBoost classifier.

So move to the next slide. So in this case we are considering a binary classification problem having 10 training examples. So this is the training dataset and suppose the dataset is D_1. So initial weight distribution initial weight distribution I can consider like this $\frac{1}{10}$ because we have 10 training examples and if I consider the initial weight as $\frac{1}{10}$ that means each point has equal weight and we are considering the weak classifiers we are considering weak classifiers for this problem for this classification problem and what weak classifier we are considering this axis parallel linear classifier we can consider linear classifier. So you can see here we are considering 10 training examples 10 and what I am considering first we are considering initial weight distribution. So it is $\frac{1}{10}$ so each point has equal weight that is equal to $\frac{1}{10}$ and we are considering axis parallel linear classifiers that is the weak classifiers.

So next we are considering suppose round 1 that is the after round 1. So we are considering the classifier suppose classifier is h_1 and this is a linear classifier. So if I consider this

classifier you can see some of the samples are misclassified. So like this sample and these samples these are misclassified. So what is the principle of the adaboost filter I have to give the importance to the misclassified samples.

So I have to give importance to the misclassified samples. So based on this I am getting the second data set that is D_2. So in this case suppose the error rate of the classifier h_1 is suppose $\epsilon_1 = 0.3$. So the weight of h_1 so that is $\alpha_1 = \frac{1}{2} ln(\frac{1-\epsilon_1}{\epsilon_1}) = 0.42$.

So each misclassified point up weighted that is the weight is multiplied by exponential of α_1. So I can write is misclassified point up weighted so how to do the up weighted this weight is multiplied by that means how to do the up weight the weight is multiplied by exponential of α_1. So weight is multiplied by exponential of α_1 and similarly for a correctly classified samples correctly classified points classified point down weighted down weighted. So how to do the down weight the weight is multiplied by exponential of $-\alpha_1$. So you can see I am giving importance to the misclassified sample because the weight is multiplied by exponential of α_1 and for the correctly classified samples what we are doing we are down weighted that means weight is multiplied by exponential of $-\alpha_1$.

And after this I am getting the new data set the new data set is D_2. So this is after round 1 after this I move to the round number 2. So this is after round 2. So in the round number 2 we are considering another weak classifier that is a linear classifier that is h_2 previously we considered h_1 now we are considering h_2. So in this case also you can see I have some misclassified samples.

So what are the misclassified samples? So this sample this sample and also this sample these are misclassified. So I have to give the importance to these misclassified samples. So here you can see I am giving the importance to these samples I am giving the importance to these samples I am giving the importance to these samples. So in this case also I can determine error rate.

So error rate of h_2 $\epsilon_2 = 0.21$ and we have to determine the weight. Weight of h_2 that is $\alpha_2 = \frac{1}{2} ln(\frac{1-\epsilon_2}{\epsilon_2}) = 0.65$. And after this each misclassified point I have to give the importance. So I can write is misclassified point up weighted that is the weight is multiplied by exponential of α_2.

This is for the misclassified samples and each correctly classified points correctly classified points down weighted. So how to do the down weight this weight is simply multiplied by exponential of $-\alpha_2$. So this is the concept. So this is about the round 2.

So after round 2 I am getting this one. Now move to the round number 3 in the next slide. So after round 3 so you can see first we considered h_1 after this we considered h_2 now we are considering this one h_3. These are all weak classifiers and corresponding to h_3 also you can see there are some misclassifications. So this is this is the misclassification and in this case also we can determine the error rate.

So error rate of h_3 is equal to 0.14 and weight because we have 10 number of samples from this we can determine the error rate. So how many samples are misclassified based on this we can determine the error rate. So weight of h_3 now I can determine the weight of h_3 is α_3. So $\alpha_3 = \frac{1}{2} ln(\frac{1-\epsilon_3}{\epsilon_3}) = 0.92.$

So after this suppose I want to stop so our ensemble consist of 3 classifiers. So finally I have the 3 classifiers and these are h_1, h_2 and h_3. So that is the ensemble classifier. So finally I have 3 classifiers and this is the ensemble one.

So we are combining these 3. So what will be the final classifier so let us move to the next slide. So this is the final classifier because we have to combine all these 3 classifiers. So how to combine the combination formula already I have explained. So I have shown the combination formula. The combination formula if you see in my previous slide it is nothing but $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$.

So we consider this formula for final output. So the final classifier is a weighted linear combination of all the classifiers. So we are considering 3 classifiers. So the final classifier is a weighted linear combination of all these 3 classifiers. So classifier h_1 gets a weight of α_1.

So what is my h_{final}? h_{final} is equal to signum. So we have to combine all these classifiers.

So it is $0.42 + 0.65 + 0.92$. So we are combining all these classifiers 3 classifiers one is h_1 another one is h_2 and this is h_3. So we are combining with the help of the signum function. So that means the final classifier is a linear combination of all the classifiers. So earlier we computed α_1 if you see my previous slide α_1 I computed like this it is 0.42, α_2 the value was 0.65 and α_3 I computed like this 0.92. So that means after combination I am getting this output. So this is the final classifier. So this is the final strong classifier. So h_1,h_2 and h_3 these are combined and finally I am getting this classifier the final classifier I am getting. So multiple weak linear classifiers are combined to get a strong nonlinear classifier.

So here you can see I am getting a nonlinear classifier and h_1,h_2 and h_3 all these are linear classifiers. So multiple I can write the multiple weak linear classifiers are combined to get a strong nonlinear classifier. So this is the concept of the AdaBoost filter. So if I want to compare these two principles one is the bagging and another one is the boosting.

So I can say bagging versus boosting. So first point is actually they are data dependent. So who is the winner? I do not know but it depends on data. The first point is the computational complexity. So bagging is computationally more efficient than boosting. So that means the bagging is better than boosting if I consider this point the computational complexity.

Second is the both the techniques can reduce the variance. So both can reduce variance. Variance means overfitting by combining different models. So the resulting model has higher stability as compared to the individual ones. So this is the comparison.

Another important point is so bagging actually cannot reduce the bias. So I can write bagging usually cannot reduce the bias but boosting can. And the bagging usually performs better than boosting if we do not have high bias. If I want to reduce only the variance then I think the bagging is better. So I can write the bagging performs better than boosting if we do not have high bias then I want to only reduce the variance and only want to reduce variance.

So these are the comparisons between bagging and the boosting. In this class I explained the concept of boosting and also I explained the concept of AdaBoost classifier. In case of the boosting I can combine number of weak classifiers to get a strong classifier and that is the fundamental concept of the AdaBoost classifier. I can give the importance to the samples which are not correctly classified by the weak classifiers. So I can combine all the weak classifiers to get a strong classifier.

So this is about the AdaBoost or the boosting principle. If I want to compare the bagging and the boosting the both can reduce variance that means the overfitting problem can be reduced with the help of boosting and the bagging. And if I want to compare the computational complexity boosting is more computationally complex than bagging. And if I want to reduce bias then bagging is not a good principle I have to consider boosting. So that is the distinction between bagging and the boosting. So let us stop here today. Thank you.