Boosting : Intuition and Advantages

Saiprasad Koturwar
Sr. Data Scientist
Dream11, Mumbai

Learners

1. Learner : An algorithm (mathematical, rule based etc.), which takes a data point as input (explanatory variables) and outputs an inference (response) on the input (e.g. Good/Bad, stock market price etc.)
2. Weak learner: A learner which performs poorly(in terms of a desired performance metric, e.g accuracy, mse etc.) over the given data
3. Problem statement: Given a set of weak learners, can we come up with a better learner (which performs better than the existing learners)?

Bagging : Improving over the weaker learners

1. Let us consider a binary learner, B_1 tasked with deciding whether a particular situation is good or bad (e.g. medical diagnosis, investment strategies)
2. We somehow know (e.g. through historical evidence), that it is accurate only 60% of the times. (i.e. $P(B_1 \text{ is right}) = 0.6$)
3. If we have three such learners, can we improve?
   a. P(all three are wrong) = $0.4*0.4*0.4 = 0.064 = P_1$
   b. P(Two of them are right and one of them is wrong) = $3*(0.6*0.6*0.4) = 0.432 = P_2$
   c. P(all three are correct) = $0.6*0.6*0.6 = 0.216 = P_3$

Voting

1. What if we ask all the weak learners about their opinion, and choose the option which majority of learners prefer?
2. P(majority learner is correct) $= \mathrm{P}_2 + \mathrm{P}_3 = 0.432 + 0.216 = 0.648 > 0.6$
3. The majority learner performs better than our weaker learners

Bagging in Real Life

1. Elections in democratic countries
2. Shopping (electronic appliances, home buying etc.)

Bagging Shortcoming

1. Each classifier has equal say (may not be ideal always, e.g. if $P(B_1$ is right$) = 0.95$?)
2. Performance of a learner is a function of the data point, basic voting does not tackle this

Boosting : AdaBoost

Adaptive Boosting(a.k.a AdaBoost): Intuition

1. Weak learners can be domain experts (i.e. they may perform well for certain data points and not so well for other)
2. More relevant examples,
    a. Should a finance minister have the equal say in a matter if the problem at hand concerns with the country's poor performance in a sports tournament?
    b. Would you consult your tech geek friend if you want to buy a new fashion outfit?
    c. Would you consult a specialist batsman if you are bowling in death overs?
3. Instead of just taking majoritarian opinion, can we sophisticate the process?

Improving over basic bagging

1. Loss function:
   a. $loss = \sum_{i=0}^{N} e_i$
2. Let us make the loss, a function of data points
   a. $weighted\ loss = \sum_{i=0}^{N} w_i * e_i$

      where , w_i is the normalized weight of the i^{th} data point.
3. What if, the next learner concentrates heavily on the data points where our current state of the art fails?
4. We can build a sequential learner which improves at each stage
   a. $F(x) = \sum_{i=0}^{M} \alpha_i * F_i(x)$

AdaBoost : Algorithm

1. Choose M (the number of stages/learners to be learnt)
2. Assign $w_i = 1/N$ for i^{th} data point, for i=1,2,..N
3. Fit a learner on the given data points with the following loss function
   a. With e_i being the loss on i^{th} data point $weighted\ loss = \sum_{i=0}^{N} w_i * e_i$
4. Compute the overall loss/error rate for the given learner
   $$\frac{\sum_{i=0}^{N} w_i * e_i}{\sum_{i=0}^{N} w_i}$$
5. For a binary classifier the error bound is minimized if $\alpha_i = \frac{1}{2} * \log(\frac{1 - error\ rate_i}{error\ rate_i})$
6. Update the weights of data points $w_i = w_i * \exp(\alpha_t * e_i)$
7. Repeat steps 2 to 6 for M iterations

Boosting: Gradient Boosting

Gradient Boosting: Working

1. AdaBoost tries to improve based on concentrating on the data points where the current state of our system is failing
2. Can we do something different?
3. What if, instead of concentrating on the failed data points, we concentrate on the error at each point?
4. Gradient boosting, is a variant of sequential boosting approach, where we are trying to find the error vector at each stage

Gradient Boosting: Algorithm

1. Choose M (no. of stages)
2. Given a set of data points $(\boldsymbol{x_i}, y_i)$, fit a base learner $\mathrm{F} = f_0$
3. For every data point calculate the error (e_i) using base learner F
4. Create new set of data points $(\boldsymbol{x_i}, e_i)$, fit a learner on the residual function h_t
5. Update the base learner using equation $\mathrm{F} = \mathrm{F} + h_t$
6. Repeat the steps 3 to 5 for M iterations

Thank You