HW 6

Quyen Dang

11/15/2024

What is the difference between gradient descent and *stochastic* gradient descent as discussed in class? (*You need not give full details of each algorithm. Instead you can describe what each does and provide the update step for each. Make sure that in providing the update step for each algorithm you emphasize what is different and why.*)

In both gradient descent and stochastic gradient descent, we are updating a set of parameters iteratively to minimize an error function. However, in gradient descent, the parameters are updated in each iteration by computing the gradient using the entire training dataset. In comparison, stochastic gradient descent updates the parameters using only one data point or subset per iteration. The update step for this algorithm is similar to gradient descent, with the only difference being that it uses X_i and Y_i, denoting the single data point/subset instead of X and Y, which denotes the entire dataset.

Gradient Descent Update Step: $\theta_{i+1} = \theta_i \nabla f(\theta_i, X, Y)$

Stochastic Gradient Descent Update Step: $\theta_{i+1} = \theta_i \nabla f(\theta_i, X_i, Y_i)$

Consider the `FedAve` algorithm. In its most compact form we said the update step is $\omega_{t+1} = \omega_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} \nabla F_k(\omega_t)$. However, we also emphasized a more intuitive, yet equivalent, formulation given by $\omega_{t+1}^k = \omega_t - \eta \nabla F_k(\omega_t); w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$.

Prove that these two formulations are equivalent.
(*Hint: show that if you place* ω_{t+1}^k from the first equation (of the second formulation) into the second equation (of the second formulation), this second formulation will reduce to exactly the first formulation.)

Substitute $\omega_{t+1}^k = \omega_t - \eta \nabla F_k(\omega_t)$ into $w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$. Result: $w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} (\omega_t - \eta \nabla F_k(\omega_t))$.

Then, distribute out $\mathrm{w_t}$ and η: $w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} \omega_t - \sum_{k=1}^{K} \frac{n_k}{n} \eta \nabla F_k(\omega_t))$.

Take out $\mathrm{w_t}$ and η outside of summation:

$w_{t+1} = \omega_t \sum_{k=1}^{K} \frac{n_k}{n} - \eta \sum_{k=1}^{K} \frac{n_k}{n} \nabla F_k(\omega_t))$

Recognize $\sum_{k=1}^{K} \frac{n_k}{n} = 1$. Therefore, $w_{t+1} = \omega_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} \nabla F_k(\omega_t))$, which is our first formulation.

Now give a brief explanation as to why the second formulation is more intuitive. That is, you should be able to explain broadly what this update is doing.

The second formulation is more intuitive because it more clearly shows how the local client first takes one step size of SGD using current local data, followed by taking a weighted average of the resulting local models.

Prove that randomized-response differential privacy is ϵ-differentially private.

In a randomized response, a user has a true response ("Yes" or "No") to a sensitive question. To protect privacy, they report their answer in a randomized way. To prove that this is epsilon-differentially private, let's first define a few variables. Let A be the randomized response mechanism, D be the true answer of

Y/N, and S be the output Y/N answer.For a mechanism A to be epsilon-differentially private if for any tow possible inputs D1 and D2 and any possible output S: $\frac{Pr[A(D1)\in S]}{Pr[A(D2)\in S]} <= e^{\epsilon}$

For the rnadomized response, consider both S = Yes and S = No. For S = Yes, $\frac{Pr[A(Yes)=Yes]}{Pr[A(No)=Yes]} = \frac{3/4}{1/4} = 3 = e^{ln(3)}$. For S = No, $\frac{Pr[A(Yes)=No]}{Pr[A(No)=No]} = \frac{1/4}{3/4} = 1/3 = e^{-ln(3)}$.

Since $e^{-ln(3)} <= \frac{Pr[A(D1)\in S]}{Pr[A(D2)\in S]} <= e^{ln(3)}$, we can conclude that the randomized response satisfies epsilon differential privacy with $\epsilon = ln(3)$

Define the harm principle. Then, discuss whether the harm principle is *currently* applicable to machine learning models. (*Hint: recall our discussions in the moral philosophy primer as to what grounds agency. You should in effect be arguing whether ML models have achieved agency enough to limit the autonomy of the users of said algorithms.*)

The harm principle, described by JS Mill, describes how individuals should have their personal autonomy restricted only when using autonomy would result in objective moral harm.

To figure out whether the harm principle is currently applicable to machine learning models, we have to consider whether machine learning models possess agency, or the ability to understand, make decisions, and be held accountable for its actions. Currently, I argue that ML models lack true agaency because they operate based on algorithms and finding patterns in datasets without having much understanding or moral reasoning at all. While they can indirectly impact user autonomy and cause harm, such as through biased HR decisions, they do not independently choose to do so. Therefore, the harm principle cannot apply to ML models themselves. Instead, accountability for potential harm from ML models lies with those who develop and use them.