Tab 1

Report date: 12/01/24

Paper report:

**Overview:**

The paper stems from a Kaggle competition started by Google on machine unlearning, the teams had to develop an unlearning algorithm to perform image recognition on a set of people images and had to assign the person to a set of age ranges.

The paper talks about 2 main topics:

* The evaluation metrics they developed to evaluate the unlearning algorithms performance
* The best algorithms approaches

The main topic is the first one.

The paper explains how evaluating the performance of unlearning is in itself an open research problem, as well as unlearning itself, especially for non convex loss functions.

The paper provides a definition of machine unlearning which also takes explicitly into consideration the model weights (it's the first paper in which I see it).

The whole idea is based on the fact that the unlearned model should be as close as possible to the retrained model (retrained without the data to forget), this closeness is expressed in terms of models weights distribution closeness.

They do refer to the distribution of the weights since they intend to get the weights distribution from multiple runs each having random seeds (in order to avoid issues described in Appendix Section 5).

Distribution distance can be measured with Kolmogorov-Smirnov test or KL divergence, but it would be complex to then measure the actual model closeness since it would require several runs and several model retraining due to the random seeds usage need.

They thereby define (ε,δ)-unlearning, a definition coming from the differential privacy world that requires two probabilities distributions (PD) to be almost identical, specifically:

PD[retrained algo weights] <= eε PD[unlearnt algo weights]+δ

and

PD[unlearnt algo weights] <= eε PD[retrained algo weights]+δ

In the paper (and Kaggle competition) they defined a small δ value and just computed each algorithm ε.

Just as it can be done in Differential Privacy, we can use empirical estimation of ε based on false positive and false negative rates. The main achievement of this paper is the final score formula made up by: forgetting quality formula coming from the ε value and the utility of the unlearning system (so how accurate is the model after it has unlearnt).

Apart from forgetnessful and utility the unlearning method should also be timely and take less than retraining the whole model (else, it doesn't make any sense), so the authors have decided to set a timeout to unlearning runs to 20% the time it takes to retrain. Sadly this approach doesn't allow them to take speed into consideration while evaluating the models.

The forgetting quality is measured via the hypothesis testing interpretation of (ε,δ)-unlearning, with some tricks to make the computation faster and more meaningful (ε value is not enough to define what algorithm is the best). Specifically they work on the scalar output distribution when given an example from the forgot set and by aggregating the ε values among different values in the forget set.

The false positive rate and false negative rate is computed based on a membership attacker trying to guess to which distribution a given result belongs to. They use different attack algorithms and take the best result among them in terms of FPR and FNR.

They do present different versions of the evaluation framework with different computing power requirements and reliability.

They quickly present the submitted algorithms and note that most of the top methods are comprised of a erase phase and a repair phase, in the erase phase they change weights and create chaos in the model in order to erase it, in the repair phase they retrain the model on the retain set.

Please remember that we are talking about non-convex loss functions and so there isn't any known exact unlearning method available.

The paper goes on by analyzing the results of the challenge without any revolutionary outcomes.

The appendix is very used to better understand the implementation and to get a better understanding of the best algorithms behaviour.

The also present alternative metrics for forgetting quality such as:

* Accuracy gap (accuracy difference of the unlearnt model vs the retrained model on the forget set only)
* Relearn time (how much time it takes for the algorithm to relearn the forget set, the more the better)
* L2 distance in the weight space of the models (not very meaningful actually)
* KL-divergence in the model weights distributions (but we would need to retrain the model on the forget set and see its weights distribution, and train everything multiple times to get a distribution)

Overall the most interesting chapters of this paper are 1-3 because of the definitions provided and the intuition on how to evaluate forgetfulness capability of a model.

**Repo:**

<https://github.com/google-deepmind/unlearning_evaluation>

<https://www.kaggle.com/competitions/neurips-2023-machine-unlearning>

**Authors:**

Triantafillou et alii.

**Year:**

2024