The notes of
RELAXLOSS: DEFENDING MEMBERSHIP INFERENCE ATTACKS WITHOUT LOSING UTILITY

(Published as a conference paper at ICLR 2022)

Key words: Relax Loss, Confidence Suppression, Gradient Ascent

Index/Abstract:

RelaxLoss: Blurring the member/non-member loss distribution by increasing the mean of the training target loss.
“**Relaxing Loss Target with Gradient Ascent**
“In order to bypass these issues and reduce the distinguishability between the member and non-member loss distributions, we propose to simplify the problem by considering the mean of the loss distributions, and subsequently set a more achievable mean value for the target loss, where the loss target is relaxed to a level that is easier to be achieved for the non-member data. Algorithmically, instead of pursuing zero training loss of the target model, we relax the target mean loss value α to be larger than zero and apply a gradient ascent step as long as the average loss of the current batch is smaller than α.””

Posterior Flattening: Keep the confidence of the ground-truth while equally distributing the remaining probabilities to other classes.
“**Posterior Flattening: Improve Model Utility**
“To address this issue, we propose to encourage a large margin between the prediction score of the ground-truth-class and the others by flattening the target posterior scores for non-ground-truth classes. Specifically, we dynamically construct softlabels during each epoch by: (i) retaining the score of the groundtruth class, i.e., the current predicted value p gt , and (ii) re-allocating the remaining probability mass evenly to all non-ground-truth classes.””

Introduction

RelaxLoss is a new idea and a unique strategy introduced by Dingfan Chen, Ning Yu, and Mario Fritz to defend against membership inference attacks, particularly those relying on loss-based metrics. The core components can be categorized into two parts: relaxed loss functions(Gradient descent) and posterior flattening soft labels.

Typically, an attacker heavily relies on the loss distribution of samples output by the model, where member samples usually exhibit lower loss values, while non-member samples have higher losses. The core idea of RelaxLoss is to introduce a relaxed target loss threshold, denoted as α, in order to blur the boundary between member and non-member loss distributions. By dynamically adjusting the loss values during training, it reduces their distinguishability, thereby mitigating the effectiveness of MIA attacks.

However, relaxing the loss can reduce the model’s confidence in the ground-truth class, which may increase misclassification risk and degrade predictive accuracy. To address this, the method introduces another key mechanism called posterior flattening.

Posterior flattening softens the label distribution by retaining the predicted score of the ground-truth class, and evenly redistributing the remaining probability mass across all other classes. This ensures that the model preserves its predictive accuracy while obfuscating the output structure to resist attacks. This label transformation is one of the foundational strategies in soft-label-based defense.

According to the authors, this method aims to balance both utility (i.e., predictive accuracy) and privacy (i.e., resistance to inference attacks). While some existing state-of-the-art privacy defenses offer strong theoretical guarantees, their practicality remains under question.

The core parts can be categorized as two kinds: Relax Loss functions and Posterior Flattening soft labels.

Usually, the attacker will highly rely on the sample loss distribution that output from the models, where member samples typically exhibit lower loss values and non-member samples higher ones. The core idea of RelaxLoss is to introduce a relaxed target loss threshold, denoted as α, to blur the boundary between member and non-member loss distributions. By dynamically adjusting the loss values during training, it reduces their distinguishability, thereby mitigating the effectiveness of MIA attacks.

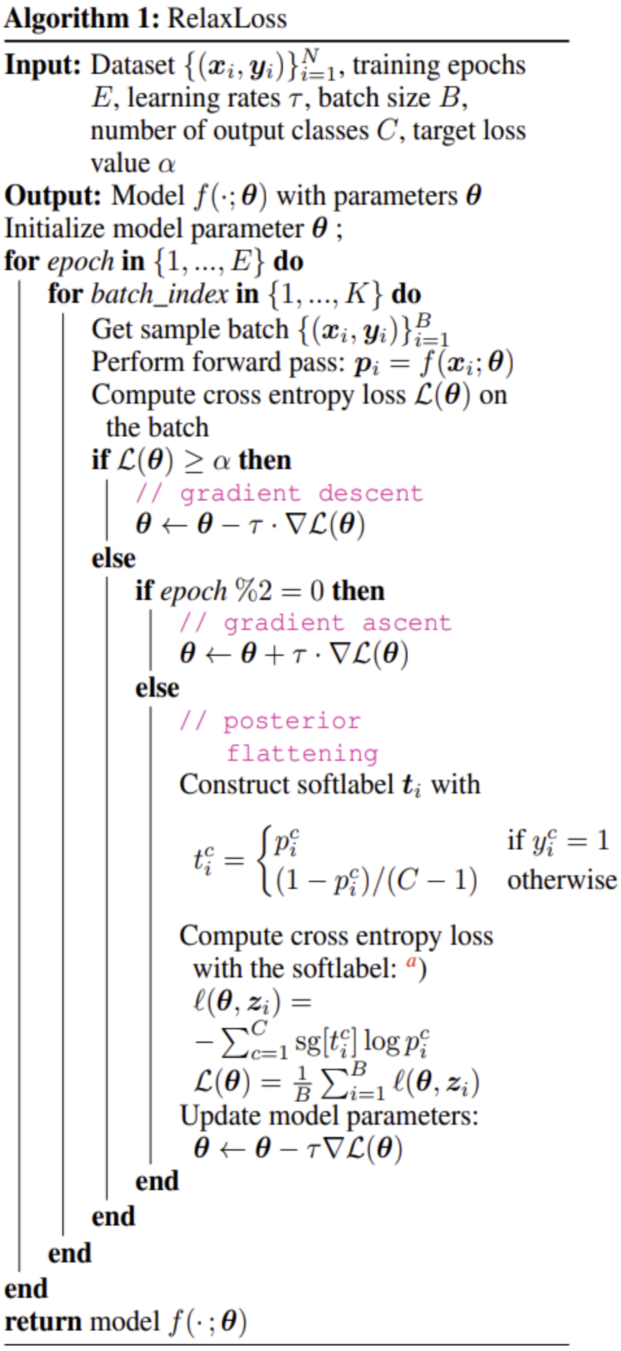
But however, the strategy of relaxing loss to defend against the MIA attack (target on loss) will sometimes mistakenly distribute the loss to the false labels. To fix this problem will introduce another significant function called posterior flattening.

Such as
Relax Loss: [0.80, 0.05, 0.15] -> [0.40, 0.35, 0.45]

This represents a softened output distribution where the model no longer assigns overly confident scores to the predicted class.

Such as
posterior flattening: [0.40, 0.35, 0.45] -> [0.40, 0.30, 0.30]

Posterior flattening retains the confidence score for the predicted (ground-truth) class while evenly redistributing the remaining probability mass among the other classes. This helps preserve accuracy while mitigating overconfident patterns that can be exploited by attacks.



```
Algorithm 1: RelaxLoss
Input: Dataset $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$, training epochs
       $E$, learning rates $\tau$, batch size $B$,
       number of output classes $C$, target loss
       value $\alpha$
Output: Model $f(\cdot;\boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$
Initialize model parameter $\boldsymbol{\theta}$ ;
for epoch in $\{1, ..., E\}$ do
    for batch_index in $\{1, ..., K\}$ do
        Get sample batch $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^B$
        Perform forward pass: $\boldsymbol{p}_i = f(\boldsymbol{x}_i;\boldsymbol{\theta})$
        Compute cross entropy loss $\mathcal{L}(\boldsymbol{\theta})$ on
          the batch
        if $\mathcal{L}(\boldsymbol{\theta}) \geq \alpha$ then
            // gradient descent
            $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \tau \cdot \nabla\mathcal{L}(\boldsymbol{\theta})$
        else
            if epoch $\%2 = 0$ then
                // gradient ascent
                $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \tau \cdot \nabla\mathcal{L}(\boldsymbol{\theta})$
            else
                // posterior
                   flattening
                Construct softlabel $\boldsymbol{t}_i$ with
                  $t_i^c = \begin{cases} p_i^c & \text{if } y_i^c = 1 \\ (1 - p_i^c)/(C-1) & \text{otherwise} \end{cases}$
                Compute cross entropy loss
                  with the softlabel: [a])
                  $\ell(\boldsymbol{\theta}, \boldsymbol{z}_i) =$
                  $-\sum_{c=1}^C \mathrm{sg}[t_i^c] \log p_i^c$
                  $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{B}\sum_{i=1}^B \ell(\boldsymbol{\theta}, \boldsymbol{z}_i)$
                Update model parameters:
                  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \tau\nabla\mathcal{L}(\boldsymbol{\theta})$
            end
        end
    end
end
return model $f(\cdot;\boldsymbol{\theta})$
```

The pseudo code:

Gradient adjustment: If the batch loss exceeds the target threshold α, gradient descent is applied to reduce it. If it falls below α and the epoch is even-numbered, gradient ascent is used to intentionally increase the loss and spread its variance.

(Gradient descent only happened when epoch is even, if it's not (epoch is odd), then go posterior flattening)

Posterior flattening: A soft label is constructed by keeping the predicted probability for the ground-truth class, and distributing the remaining probability uniformly across all other classes. This ensures smoother outputs while preserving accuracy.

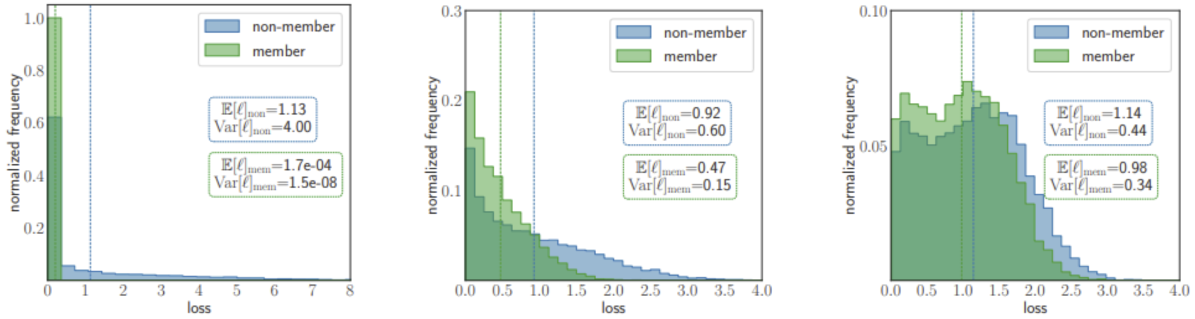
If:
L(θ)≥α **then** gradient descent (keep learning)
L(θ)<α and epoch is even **then** gradient ascent (will lower the gradient)
L(θ)<α and epoch is odd **then** posterior flattening（construct the soft label, smooth output, ensure the accuracy）

Gradient Descent is an optimization technique that minimizes the loss function by updating model parameters in the direction of the negative gradient.
It is used to help the model learn and improve prediction accuracy.
Goal: Reduce loss → Improve model performance

Gradient Ascent is the opposite process—it increases the loss value by updating model parameters in the direction of the positive gradient.
In RelaxLoss, it is used to intentionally enlarge the loss (especially for member samples with very low loss) to reduce the distinguishability between member and non-member samples.

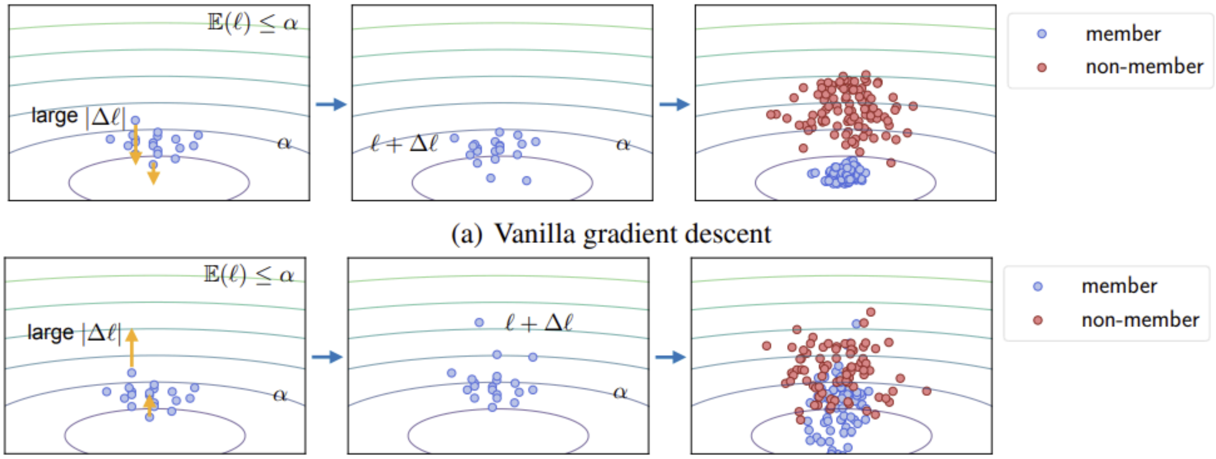
Goal: Increase loss → Blur the difference between member and non-member losses → Defend against membership inference attacks

In the experiment (Loss histograms on CIFAR-10 with ResNet20), when α is in different values then:



(a) Vanilla (b) Ours ($\alpha = 0.5$) (c) Ours ($\alpha = 1.0$)

Vanilla: original model
When α is equal to 1.0, the generalization gap is reduced and the distribution increases and becomes wider, the member and non-member are mostly matched, hard to identify.



(a) Vanilla gradient descent

(b) Gradient ascent

Advantages: Changing the behavior of the model (its operation on loss and its sensitivity to various samples) without changing the underlying operation of the model.

Disadvantages: Passive defense, defense against a specific attack mode. If the attack comes from other directions, the defense effect is limited.

Initial new ideas (maybe)
Using DP-SGD to boost the privacy performance to construct a hybrid defence, meet the definition and rules of privacy protection.

Per-sample adaptive loss perturbation, additional relax loss by using a new setting of α that if Apply stronger perturbations to samples with “extremely high confidence” and apply light or no perturbations to samples with “originally unstable predictions”

Per-sample adaptive α + variance-aware gradient ascent

OR

The α of RelaxLoss does not need to be set too high (for example, α = 1.0 will overly disturb the loss distribution and damage the utility). We can use a smaller α (such as 0.5) + slight DP-SGD noise to combine and achieve a softer balance between privacy protection and model accuracy.

To calculate the variance (the dataset loss distribute)

$$\mathrm{Var}(\ell + \Delta\ell) = \mathrm{Var}(\ell) + \mathrm{Var}(\Delta\ell) + 2\mathrm{Cov}(\ell, \Delta\ell)$$

The final **loss** variance = the original variance of the **loss** + the variance of how much each sample's **loss** is increased (Δloss variance) + the additional variance caused by the trend that "samples with higher original **loss** tend to be increased more" (the covariance term).

Variance in RelaxLoss refers to the spread of training loss values. By increasing variance (via gradient ascent), the model breaks the tight clustering of member losses, thus confusing attackers and reducing the effectiveness of membership inference attacks.

In the experiment, the author excluded the “label only attack” since it is usually identified as a weak attack.
Use white box and black box attack with the “strongest attacker model” (Grad-x and Grad-w)
In white box, the attacker could access the model structure and gradient compute.
In black box, the attacker could access the loss, entropy and logistic values.

Grad-x (whitebox attack)
Grad-x is a white-box membership inference attack. The principle is that the attacker computes the model loss and then calculates its gradient with respect to the input of a given sample to determine whether the sample belongs to the training set. Since training data has been optimized by the model, a member sample typically results in a small input gradient norm, whereas a non-member sample (unseen by the model) can cause a larger gradient magnitude. Based on this observation, the attacker computes the input gradient norm and compares it with a threshold: if the norm is smaller, the sample is likely a member; otherwise, it is likely a non-member.

Grad-w (whitebox attack)
Grad-w is a white-box membership inference attack. The attacker computes the model loss on a given sample and then calculates the gradient of the loss with respect to the model’s parameters. Since member samples have already been used to optimize the model, their corresponding parameter gradients are usually smaller. In contrast, non-member samples tend to produce larger gradients, indicating the model is less familiar with them. By measuring the norm of the parameter gradients and comparing it with a threshold, the attacker can infer whether the sample is likely a member (smaller norm) or a non-member (larger norm).

Norm
In this case, the norm value is used to quantify the magnitude of the gradient change. The gradient norm for training samples is usually smaller, while non-training samples can lead to larger gradient changes.