Dataset Condensation for Data Privacy

Eshan Gujarathi
Indian Institute of Technology Gandhinagar
eshan.rg@iitgn.ac.in

Hitarth Gandhi
Indian Institute of Technology Gandhinagar
hitarth.g@iitgn.ac.in

Abstract

There are a lot of black box models released by service providers like Amazon and Google which use the customer's data for training. Even these black box models are vulnerable to privacy attacks and it makes it crucial to make the black box models more private. In this work, we explore the use of dataset condensation, which provides two-fold benefits, more efficiency in training, and also making the data more private for free.

Keywords and phrases Dataset Condensation, Gradient Matching, Differential Privacy, Membership Inference Attack

Supplementary Material *https://github.com/eshan-rg/Dataset-Condensation-for-Data-Privacy*

Acknowledgements We are extremely grateful to Prof. Anirban Dasgupta for his guidance and support throughout the project. We are also thankful to IIT Gandhinagar for providing us with the resources needed to complete the project.

1 Introduction

We live in a world where all of us are dependent on various service providers like Amazon and Google. These service providers have a lot of our data that they use to train machine learning models for various downstream tasks. They open-source these models as black box models for others to use and we believe that our data here is not compromised. But in fact, these black box models can still be attacked to leak data they have been trained on. The attackers use various attacks like the Membership Inference Attacks to gain insights on the data used to train a model with only the black-box access to the model. This makes data privacy a very critical domain.

There have been various works to make these black-box models, also known as target models, more private, such as adding noise to the data or model while training the target model. But these techniques are not time efficient. We thus explore the use of Dataset Condensation in the domain of privacy. Dataset condensation not only benefits in terms of training efficiency but also provides privacy for free. It creates an extra layer between the attacker and the training data, which makes it difficult for the attacker to backtrack to the original member data of the training dataset.

2 Background and Related Work

2.1 Membership Inference Attack

For privacy analysis, we use the Membership Privacy as a measure of privacy as it directly relates to the personal privacy. In membership privacy, the attacker tries to identify whether a data point was a member of the training data or not. This type of attack is called a Membership Inference Attack(MIA).

Loss-Based MIA

The loss-based MIA infers membership by the predicted loss: if the loss is lower than a threshold τ , then the input is a member of the training data. Formally, the membership $M(x)$ of an input x can be expressed as:

$$M(x) = \mathbb{1}(l(x) \leq \tau)$$

where $M(x) = 1$ means that x is a member of the training data, $l(x)$ is the loss of the model on data point x and $\mathbb{1}(A) = 1$ if event A is true.
The intuition behind this kind of attack is that a trained model will give very different distributions of losses on the data it has already seen before vs the data it has not seen before. Thus we can find an intersection of the two kinds of distributions to find a threshold through which we can say weather the model has seen the sample before or not.

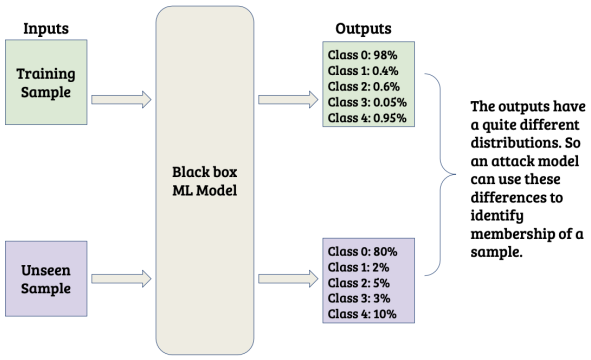

Figure 1 Difference in Softmax Predictions for training sample and non-training sample

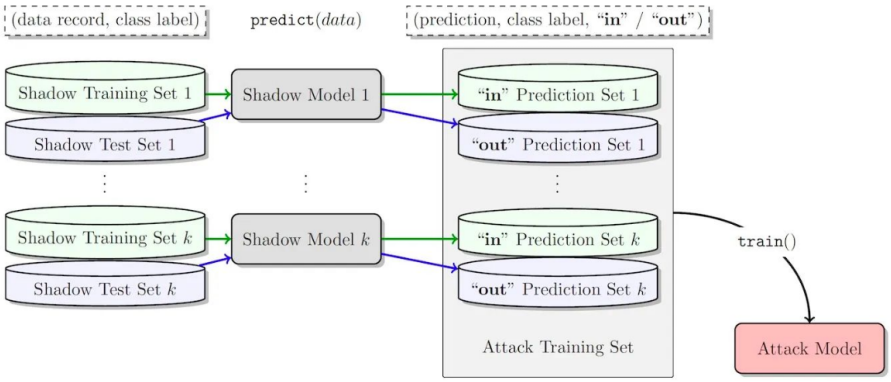

Figure 2 Attack model and Shadow Model training

We find the value for the threshold τ by locally trained shadow models [3]. This technique makes use of the fact that if the distribution of losses is very different, then the distribution of softmax values predicted by the model will also be quite different as seen in Figure 1. Thus we can make use of this difference to train an attack model to classify weather the data point was in training or not. The input to this attack model is created by training several shadow models for which we know which data points are in training and which are not.

2.2 Existing Methods to Counter MIA

Previously, GANs were used as an alternative to data sharing. But the aforementioned privacy risks are not mitigated by GANs as they create a synthetic dataset which is very similar to the real dataset and it is easy to match the fake data generated by GANs with real data.
Current approaches [4, 1] apply differential privacy to develop differentially private data generators which work by adding noise to different parts of the data to preserve privacy. Because of this noise, the data generated by DP-generators is of very low quality. Thus the training accuracy of models trained with these data is very low. Thus a huge amount of data is needed to train the model which inevitably increases the training time.

2.3 Dataset Condensation

Recently, the technique of dataset condensation [5] has emerged which aims to create a small synthetic dataset which can give comparable performances as the original dataset on training Deep Neural Networks. This works by creating a small informative dataset which represents the whole original dataset as opposed to the existing data generation techniques which aim to create data just like the real data with high fidelity. Since this synthesizes a small dataset, training efficiency also increases. Recent works [2] have shown that models trained using the synthetic dataset generated by dataset condensation are less vulnerable to MIA than ones trained on original dataset.

3 Experiments

3.1 Dataset

We conduct our experiments on the MNIST dataset of handwritten digits. We divide the training data of the dataset into two disjoint parts, the target data and the attacker data. The target data is used to train the target model, and the attacker data is the data used by the attacker to train the shadow models. Both the data parts thus contain 30,000 data points.

3.2 Pipeline

The following pipeline will be followed to train the attack model and the target model.

Attack model

The attacker only has access to the attacker's data. We train 10 shadow models with this data. We choose a dataset size *ds* for which each shadow model will be trained. We randomly choose *ds* samples for training and evaluation, which we call member and non-member data respectively. We use a simple custom CNN model architecture for the shadow models. We train each model for 100 epochs with a learning rate of 0.001 and batch size equal to 128.

We store the output of the shadow models on the training and evaluation data and whether the data points belong to member or non-member data, and this becomes the training data for the attack model.

We then use the training data created by the shadow models to train the attack models. We use Support Vector Machines (SVM) as the attack model. We create a different attack model for each class in our data. We use grid search to find the best parameters for the SVM model for each class.

Target Model

We use the target data here. We split it into two equal halves, member and non-member data, and use the member data for training the target model. The model architecture for the target model is the same custom CNN that we use to train the shadow models. We have the following two approaches for training that we compare:

1. Training using the entire original dataset
2. Training using the condensed dataset created by using only the member data.

We train the target model for 100 epochs with a learning rate of 0.001 and batch size of 128. We store the outputs of the target model and whether the data points belong to member or non-member data, and this becomes the test data for evaluating our attack model.

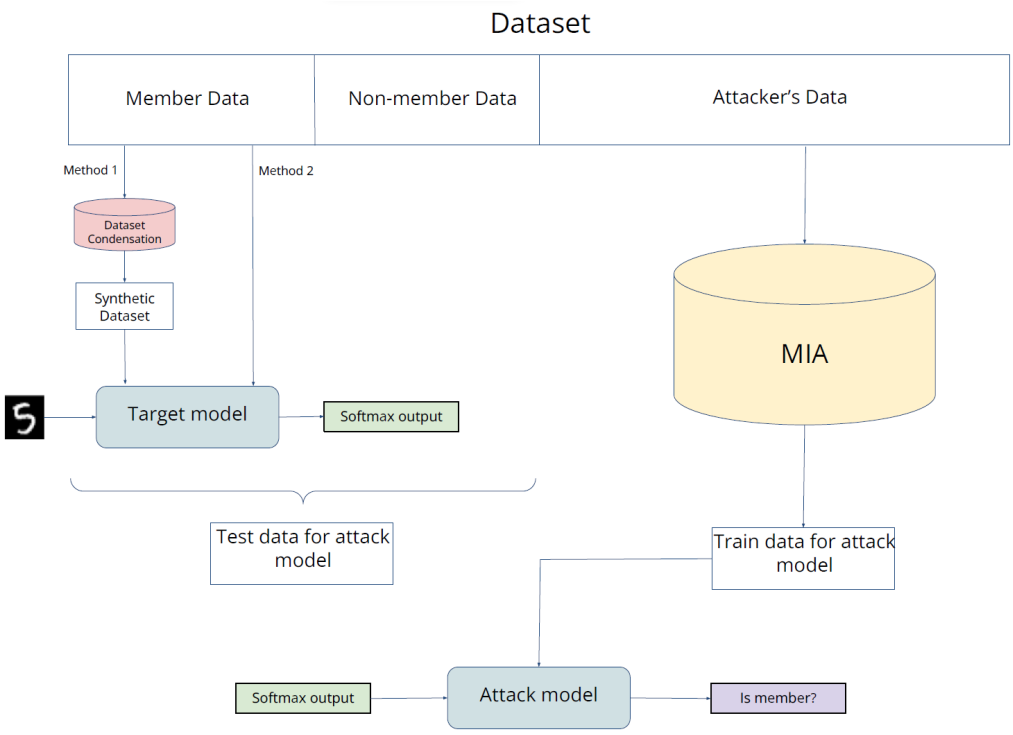

Figure 3 Pipeline for evaluating techniques against membership inference attacks

4 Results

Dataset Condensation

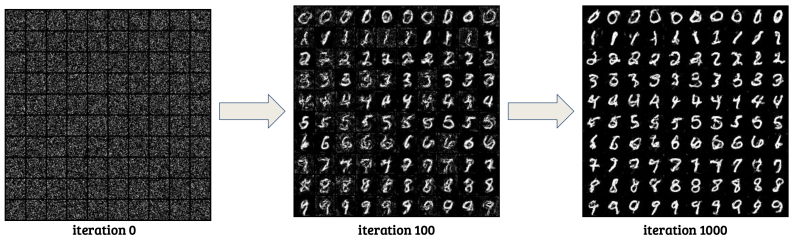

Figure 4 Dataset Condensation for MNIST dataset

Training Efficiency of Condensed Dataset vs Original Dataset

Table 1 Comparison of classification accuracy on simple CNN

Dataset	Accuracy
Original	∼99%
Original(Subset)	∼87%
Condensed (10ipc)	∼90%
Condensed (50ipc)	∼92%

Privacy Analysis for Condensed dataset vs Original Dataset

Table 2 Attacker's Metrics for different datasets

Dataset	Attacker Precision	Attacker Recall	Attacker Accuracy
Original	50.78%	79.58%	51.22%
Condensed (10ipc)	50.26%	27.66%	50.14%
Condensed (50ipc)	50.28%	39.49%	50.22%

Conclusion

The condensed dataset losses some accuracy on the downstream task as the size of the condensed dataset is much less than the size of the original dataset. But, it still performs better than a subset of the dataset as it contains features from the entire member dataset.

We can notice that there is not much decrease in the attacker's accuracy. But the relevant metric that we need to compare here is the recall. Recall answers the following question, how many of the true positives were identified correctly? We can infer from the results that the attacker was able to identify very few of the actual members of the training dataset after applying dataset condensation, thus making the member data safer. This is what we want in a real-life scenario. For example, if the target model was trained on 3 out of 100 samples, an attack model that predicts negative for all the samples would give 97% accuracy, but its recall would be 0%, preserving the privacy of the member data.

5 Shortcomings and Future Work

The attack model does not show good performance, and we suspect the following reasons for the same.

1. MNIST is a very small dataset with minor differences between two images of the same class. Thus, it becomes difficult for the model to predict whether a data point was a member of the training data. It will likely return that a data point was a member even if a similar data point was used in training. Thus, we plan to evaluate our experiments on other real-life datasets like the celeb dataset or cifar10 dataset.
2. We train the target and shadow models with a different architecture than the one used while dataset condensation. This is likely affecting the performance of the condensed dataset on the target model. We plan to use the same architecture for creating the condensed data.
3. We consider the worst-case scenario for the attacker by taking disjoint data for the target and attack models, which reduces the attack accuracy. We plan to evaluate on a more

real-life scenario where the attacker can have access to all the data, which makes it difficult to create a more private target model.

Once we prove our hypothesis, we plan to evaluate different techniques to condense data and find the most optimum technique that gives the most private data while suffering very less difference in training and performance.

References

1 Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, and Karsten Kreis. Don't generate me: Training differentially private generative models with sinkhorn divergence. 2021. URL: https://arxiv.org/abs/2111.01177, doi:10.48550/ARXIV.2111.01177.

2 Tian Dong, Bo Zhao, and Lingjuan Lyu. Privacy for free: How does dataset condensation help privacy?, 2022. URL: https://arxiv.org/abs/2206.00240, doi:10.48550/ARXIV.2206.00240.

3 Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models, 2016. URL: https://arxiv.org/abs/1610.05820, doi:10.48550/ARXIV.1610.05820.

4 Yuxin Wang, Zeyu Ding, Yingtai Xiao, Daniel Kifer, and Danfeng Zhang. Dpgen: Automated program synthesis for differential privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 393–411, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3460120.3484781.

5 Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. 2020. URL: https://arxiv.org/abs/2006.05929, doi:10.48550/ARXIV.2006.05929.