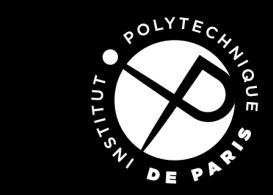
Leakage of Sensitive Data from Deep Neural Networks

Ganesh del Grosso Guzman



INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAX159

Thèse de doctorat

Leakage of Sensitive Data from Deep Neural Networks

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École polytechnique

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Mathematics and Informatics

Thèse présentée et soutenue à Palaiseau, le 7 Novembre 2023, par

GANESH DEL GROSSO

Composition du Jury :

Daniel Augot Senior research scientist, Inria, LIX École polytechnique (Grace)	Président
Sonia Ben Mokhtar Director of research, CNRS Blaise Pascal (INSA)	Rapporteur
Mark Dras Professor, Shool of Computing, Macquarie University	Rapporteur
Catuscia Palamidessi Director of research, Inria	Directeur de thèse
Pablo Piantanida Director of International Laboratory on Learning Systems (ILLS)	Co-directeur de thèse
Georg Pichler Researcher, Technical University of Vienna	Examinateur
Michaël Perrot Researcher, Inria	Examinateur
Giovanni Cherubin Segnior researcher, Microsoft Research	Examinateur

Acknoledgements

I offer my most sincere thanks to Georg, for his teaching and his good company during the four years of my thesis. Your humbleness and willingness are exemplar. Likewise, I thank Pablo and Catuscia, for their guidance, patience, teaching and motivation.
I thank my colleagues Marco Romanelli and Federica Granese. Your presence in the lab made these years a great experience. Thank you for the coffee, the talks about music and cinema and for always being willing to help. Moreover, I thank my other colleagues, Sergio, Santiago, Renan, Filippo, Sayan, Ruta, Carlos and all the others for giving life to the lab.
I thank my good colleagues at Ericsson, especially Hassan and Illyyne for welcoming me into the team and for teaching me so much.
I thank my parents, Blanca and Jose, for their unconditional and unending love and support.
I thank Catherine and Grandmere for all the nourishment and for being my family on this side of the ocean.
I thank Ghost and Stormy, just for being.
I thank my friends Marta and Ansgar for going through this journey with me.
I thank Edson for always being there for me.
Finally, but most importantly, I thank my dearest Marine, for her love, motivation and support during the hardest times.

Abstract

It has been shown that Machine Learning (ML) models can leak information about their training sets. This is a critical issue in the case where the training data is of a sensitive nature, e.g., medical applications where the data belongs to patients.

A popular approach for measuring the leakage of information from ML models is to perform inference attacks against the models. The goal of this approach is to measure the privacy of the system as the robustness to inference attacks. These attacks are mainly categorized into Membership Inference Attacks (MIAs) and Attribute Inference Attacks (AIAs). The goal of a MIA is to determine if a sample or group of samples are part of the training set of the model, while an AIA tries to infer or reconstruct a sample from the trained model.

Although there exist other methods for measuring privacy in ML, such as Differential Privacy (DP), the main focus of this thesis is on inference attacks.

This work is divided in three big chapters. The first chapter provides the motivation for our work, problem statement, review of the state-of-the-art and sets the notation and theoretical framework to be used in future chapters. The second chapter contains our main theoretical results and provides a taxonomy of membership and attribute inference attacks. The third chapter provides and thorough description of our experiments and a discussion on the results.

Our theoretical findings regarding inference attacks are described as follows: First, we derive theoretical bounds on the success rate of an attacker. This result provides an upper bound on the success probability of an inference attack in the specific case where the attacker has access to the model parameters of the trained model, and therefore in any other scenario where the attacker possesses less information. Second, we derive bounds that link the generalization gap of a ML model to the success rate of an attacker against this model. This result suggests that a ML that generalizes poorly will be susceptible to MIAs. However, the converse is not always true, as we prove with a pertinent example. Third, we derive a list of results that relate the mutual information between the trained model and its training set to the generalization gap and the success rate of the attacker.

We use our theoretical framework to describe the existing MIA strategies in the literature and we propose several novel strategies. We explore the use of Out of Distribution (OOD) techniques and diversity measures for MIAs. We also propose a technique based on the norm of the minimum perturbation necessary to make a model change its prediction using an adversarial attack. Additionally, we use our framework to describe a set of AIAs.

Our theoretical results are illustrated in a toy scenario. The lower bound relating the generalization gap to the success rate is tested and compared to state of the art MIAs in a more realistic scenario.

The bulk of our experiments are dedicated to benchmark the performance of different MIAs strategies against state of the art image classification models. We describe and categorize the existing state of the art strategies. We compare the effectiveness of the novel strategies proposed in this work to the state of the art. We empirically show that having access to additional samples that can be used as training data for the attacker does not provide an advantage over strategies that do not require additional data. We

rank different strategies based on their performance against state of the art image classification models. This result provides guidelines on how to measure the privacy robustness of a ML model.

Finally, we test the effectiveness of AIAs against a model trained to classify handwritten digits. The data set contains the identity of the writers, and we use this as the sensitive information to be determined by the AIAs.

We show with mathematical rigour and also empirically that Deep Neural Networks are susceptible to MIAs, even when they generalize well. Empirically, we show that resource hungry MIA strategies are not more effective than strategies that simply query the target ML model one time. This result suggests that the most relevant information to determine membership is contained in the last layers of the target model.

Résumé

Il a été démontré que les modèles d'apprentissage automatique (ML) peuvent divulguer des informations sur leurs ensembles d'apprentissage. Il s'agit d'un problème critique lorsque les données d'apprentissage sont de nature sensible, par exemple dans les applications médicales où les données appartiennent à des patients.

Une approche populaire pour mesurer la fuite d'informations des modèles de ML consiste à effectuer des attaques d'inférence contre les modèles. L'objectif de cette approche est de mesurer la confidentialité du système en fonction de sa robustesse aux attaques par inférence. Ces attaques sont principalement classées en attaques d'inférence de membres (MIA) et en attaques d'inférence d'attributs (AIA). L'objectif d'une MIA est de déterminer si un échantillon ou un groupe d'échantillons fait partie de l'ensemble d'apprentissage du modèle, tandis qu'une AIA tente de déduire ou de reconstruire un échantillon à partir du modèle d'apprentissage.

Bien qu'il existe d'autres méthodes pour mesurer la confidentialité en ML, comme la confidentialité différentielle, cette thèse se concentre principalement sur les attaques par inférence.

Ce travail est divisé en trois grands chapitres. Le premier chapitre présente la motivation de notre travail, l'énoncé du problème, l'examen de l'état de l'art et définit la notation et le cadre théorique qui seront utilisés dans les chapitres suivants. Le deuxième chapitre contient nos principaux résultats théoriques et fournit une taxonomie des attaques d'inférence de membres et d'attributs. Le troisième chapitre fournit une description détaillée de nos expériences et une discussion sur les résultats.

Nos résultats théoriques concernant les attaques par inférence sont décrits comme suit: Tout d'abord, nous dérivons des limites théoriques sur le taux de réussite d'un attaquant. Ce résultat fournit une limite supérieure à la probabilité de succès d'une attaque par inférence dans le cas spécifique où l'attaquant a accès aux paramètres du modèle entraîné, et donc dans tout autre scénario où l'attaquant possède moins d'informations. Deuxièmement, nous dérivons des limites qui relient l'écart de généralisation d'un modèle ML au taux de réussite d'un attaquant contre ce modèle. Ce résultat suggère qu'un modèle ML qui se généralise mal sera susceptible de faire l'objet de MIA. Cependant, l'inverse n'est pas toujours vrai, comme nous le prouvons à l'aide d'un exemple pertinent. Troisièmement, nous dressons une liste de résultats qui relient l'information mutuelle entre le modèle entraîné et son ensemble d'entraînement à l'écart de généralisation et au taux de réussite de l'attaquant.

Nous utilisons notre cadre théorique pour décrire les stratégies de MIA existant dans la littérature et nous proposons plusieurs nouvelles stratégies. Nous explorons l'utilisation de techniques de détection de distribution et de mesures de diversité pour les MIA. Nous proposons également une technique basée sur la norme de la perturbation minimale nécessaire pour qu'un modèle modifie sa prédiction à l'aide d'une attaque adversariale. En outre, nous utilisons notre cadre pour décrire un ensemble d'AIA.

Nos résultats théoriques sont illustrés à l'aide d'un scénario fictif. La limite inférieure reliant l'écart de généralisation au taux de réussite de l'attaquant est testée et comparée à l'état de l'art des MIAs dans un scénario plus réaliste.

La majeure partie de nos expériences est consacrée à l'évaluation comparative des performances des différentes stratégies de MIA contre des modèles de classification d'images les plus récents. Nous décrivons et classons les stratégies existantes dans l'état de l'art. Nous comparons l'efficacité des nouvelles stratégies proposées dans ce

travail à l'état de l'art. Nous montrons empiriquement que le fait d'avoir accès à des échantillons supplémentaires pouvant être utilisés comme données d'entraînement pour l'attaquant n'offre pas d'avantage par rapport aux stratégies qui ne nécessitent pas de données supplémentaires. Nous classons les différentes stratégies en fonction de leurs performances contre les modèles de classification d'images les plus récents. Ce résultat fournit des indications sur la manière de mesurer la robustesse d'un modèle de ML en matière de protection de la vie privée.

Enfin, nous testons l'efficacité des AIA contre un modèle entraîné à classer les chiffres manuscrits. L'ensemble de données contient l'identité des auteurs et nous l'utilisons comme information sensible à déterminer par les AIA.

Nous montrons avec rigueur mathématique et de manière empirique que les réseaux neuronaux profonds sont sensibles aux attaques d'inférence de membres, même lorsqu'ils généralisent bien. Nous montrons empiriquement que les stratégies de MIA coûteuses en ressources ne sont pas plus efficaces que les stratégies qui interrogent une seule fois le modèle ML cible. Ce résultat suggère que les informations les plus pertinentes pour déterminer l'appartenance sont contenues dans les dernières couches du modèle cible.

Contents

Chapter 1

Introduction

Machine learning has seen astounding progress in the last few decades thanks to the critical increase of computational power and the deluge of data available in recent times. This improvement has brought the development of techniques in many fields such as computer vision [30, 83, 51, 34], time series [102, 66, 26] and natural language processing [13, 7, 33, 25, 107]. In turn, these techniques have been applied in a wide range of societal applications ranging from industry [16, 54] to modern medicine [70, 103, 53, 69, 55] to art [47, 35], all of which may be considered *sensitive* domains, given the nature of the data involved.

These applications have the potential to impact our society and the lives of individuals, raising concerns about the functioning of the technology behind them [4]. Moreover, in some of these applications, the data used to train the ML models belongs to individuals or is the intellectual property of a company or institution. This raises concerns not only about the anonymity of individuals present in the data [43], but also about the leakage of intellectual property. In view of this, new regulations have emerged, such as the General Data Protection Regulation (GDPR) in Europe and the California Consumer Privacy Act in the United States [84, 1], and with them new challenges for designing and training ML models [96]. According to the GDPR, in order for an ML algorithm to be private, it must be impossible to *single out* any individual from the training set. To impose these principles in practice, there exists several practical definitions of privacy in ML.

One such definition is DP [28] which provides that the output of an algorithm should not reveal whether or not a particular individual was part of the input data. This definition provides formal guarantees for privacy. However, satisfying differential privacy has proven to be taxing for the utility of ML models [2, 111], thus we resort to other measures of privacy.

Inference attacks measure the privacy of a system through their vulnerability [90]. If there exists no attack that is capable of extracting sensitive information from a ML model, then the model is considered private. In addition to measuring the privacy of the model, inference attacks can be used to validate the model and adjust its training in order to ensure privacy. We distinguish two main kinds of inference attacks.

First, MIAs try to determine if a sample or group of samples were part of the training set of a ML model. These attacks can represent a risk when the membership information is sensitive. For example, consider a study about patients with cancer;

identifying an individual in the data implies that the individual has cancer. Moreover, these attacks can serve as a gateway for more elaborated and dangerous attacks; having identified an individual in the training set, the attacker might try to extract additional information about this individual from the trained model. In a MIAs, the attacker has access to a sample (or a group of samples) and to the target model. Being robust to such attack guarantees that the model will be robust to other attacks that try to extract more information or that have access to less resources.

Second, AIAs try to determine a target concept or attribute from the target model. This definition covers a wide range of attacks. Some have partial access to a sample and try to determine missing features from a target model that used this sample as part of its training set. Others try to reconstruct a sample representative of a target class present in the classification task of the target model. This represents a high risk for the individuals present in the training set of the target model, as their data could potentially be reconstructed by the attacker. Even when the data is not sensitive to individuals, the data could be the intellectual property of a company, and risk being stolen by AIAs.

1.1 Overview

The goal of this work is to develop methods to guarantee the privacy of ML models, and we focus on inference attacks to achieve this goal. The first stage was to develop a formalism to describe inference attacks (Section 2.1). This formalism is used to derive bounds on the success rate of inference attacks (See Theorem 1), and link the success rate of those attacks to the generalization gap of the target model (See Theorem 2). We show an upper bound on the success rate of an attacker having access to the target model's parameters. Such an upper bound provides a measure of the privacy of the system. If the upper bound is low, the success rate of any attacker is guaranteed to be low, ensuring privacy of the system. We show that the success rate of the attacker achieving the upper bound is lower bounded by the generalization gap of the target model. Intuitively, this means that a model with a large generalization gap will be vulnerable to inference attacks. Nevertheless, the converse does not necessarily hold, i.e. having a low generalization gap is not enough to guarantee the safety of a model against inference attacks. We show this by a suitable example in Section 2.1.3.

These concepts are illustrated using a toy example in Section 3.2.1. Additionally, the link between the success rate of the attacker and the generalization gap of the target model is demonstrated in a more realistic scenario in Section 3.2.2, where the target model is a generic model for image classification.

The second part of this work focuses on practical methods for launching inference attacks. We present a thorough revision of the existing methods for MIAs in Section 2.2. Additionally,

- We propose a method that leverages the magnitude of a perturbation necessary to make the target model change its prediction under an adversarial attack. The intuition behind this is that during training, the value of the loss function is minimized over samples in the training set, while adversarial attacks attempt to maximize this same quantity; thus, it should be more costly to maximize the value of the loss for samples in the training set than for it is for samples outside the training set (See Section 2.2.3).

- We propose methods inspired in the field of OOD detection. This methods include the use of the Mahalanobis distance, Fisher-Rao distance, Renyi-divergence and a learned diversity metric. The hypothesis is that there is a significant statistical difference between the distributions of the model parameters resulting from a particular sample being inside or outside the training set. Since we are therefore dealing with a problem of a shift of distributions, OOD detection techniques appear to be suitable to detect if a sample is in the distribution of the training set or not (See Section 2.3).

We show empirically that there is no gain to be had over simple methods by exploiting additional resources, such as a training set for the attacker, or additional information, such as having white-box access to the model. These results are presented in Section 3.3.

Our experiments on state-of-the-art models for image classification show a vulnerability to very simple MIAs, such as the *loss* attack. These attacks are simple in terms of the resources and information they require. For example, the loss attack computes the loss of the target sample on the target model, and thus requires only the ground truth for the target sample and one query to the model plus the loss computation. This

attack, despite being extremely cheap in terms of computational resources, achieves state-of-the-art performance and shows a significant privacy breach in the target models. On the other hand, we were unsuccessful in finding better performing attacks when using additional resources. Our results suggest that additional resources fail to improve the performance of inference attacks. Despite having tried a wide array of attack strategies, we cannot formally conclude that there does not exists a strategy that uses additional resources to produce more effective attacks. These results are presented in Section 3.3.

AIAs are described in the context of our framework in Section 2.4 and their performance is empirically verified using a target model trained to classify hand-written digits encoded in the form of time series in Section 3.4. The sensitive information in this setup is the identity of the writer, which is used by the target model in its classification task. It is shown that simple strategies provide a significant gain over a random guess in trying to determine the identity of the writer.

The rest of this chapter is organized in the following manner: Section 1.2 provides an overview of inference attacks against ML models and other related security issues in ML. Section 1.3 introduces the general notation and definitions used throughout this work.

1.2 State of the Art

Connection between Privacy Leakage and Generalization: The authors of [109] study the interplay between generalization, DP, attribute and membership inference attacks. Our work investigates related questions, but offers a different and complementary perspective. While their analysis considers only bounded loss functions, we extend the results to the more general case of tail-bounded loss functions. They consider a membership inference strategy that uses the loss of the target model, yielding an equivalence between generalization gap and success rate of this attacker. In contrast, we consider a Bayesian attacker with white-box access, yielding an upper bound on the probability of success of all possible adversaries and also on the generalization gap.

Consequently, a large generalization gap implies a high success probability for the attacker. The converse statement, i.e., "*generalization implies privacy*" has been proven false in previous works, such as [18, 67, 109]. Our work also provides a counter proof, giving an example where the generalization gap tends to 0 while the attacker achieves perfect accuracy.

In this line of work, the authors of [86] derived an attack strategy for membership inference that is optimal to their setup. However, their results rely on randomness during training and assume a specific form in the distribution of network parameters given the training set. In this sense, our Bayesian attacker can be specialized to their framework and models.

The authors of concurrent work [95] studied the trade-off between the size of the target model (number of model parameters) and the success rate of an optimal attacker within their framework. That setup differs from ours mainly in terms of the capabilities of the attacker; while our attacker has access to the model parameters and full information on the target sample, their attacker only has access to the target sample data and corresponding model output. The work [95] presents a formal relation between the over-parametrization of the model and the success rate the Bayesian attacker against a linear regression model trained on Gaussian data. Differences in the definition of the sample-space, target model and attacker capabilities lead to orthogonal results, but similar conclusions.

Membership Inference: The authors of [90] utilize MIAs to measure privacy leakage in deep neural networks. Their attacks consist in training a classifier that distinguishes members from non-members. While their first work covers the case of black-box attacks, subsequent work [74] considers white-box attacks, where the adversary has access to the model parameters. Later, in [99] the influence of model choice on the privacy leakage of ML models via membership inference was studied.

Recent works [50, 85, 93, 17] revise new and old membership inference strategies under the light of new evaluation metrics. In particular, the work in [17] takes inspiration from [86], developing an attack strategy based on estimating the distribution of the loss. Further work [64] proposes to use learned differences in distribution between outputs of intermediate layers to predict membership. In [23], a new MIA strategy is proposed, which is based on the magnitude of the perturbation necessary to successfully make the target model change its prediction. It is compared to state-of-the-art methods [85, 74].

The use of shadow models is prevalent in the MIA literature. These models mimic the behavior of the target model, while allowing an attacker access to the training set and model parameters. Many of the aforementioned MIAs require the training of an attacker model (e.g. [85]), while others require the training of shadow models

[90, 74, 64, 88] in addition to training an attacker. The attacks in [93] require only black box access to the model and no additional information, while the attacks in [23] require white box access.

Recent work [20], applies the Modified Entropy strategy proposed by [93] to launch MIAs against poisoned target models. This setup differs from previous works in the sense that the attacker plays an active role in the training by poisoning part of the training data. The work in [20] shows that the effectiveness of MIAs is highly increased against poisoned models.

Typically, when studying the privacy leakage of ML models, classifiers are considered as the target to privacy attacks. In contrast, the authors of [41] were the first to consider MIAs against generative models. A comprehensive study of MIAs against GANs and other generative models is provided in [19].

Attribute Inference/Model Inversion: A more severe violation of privacy is represented by attribute inference attacks. Mainly two forms of these attacks have been considered in the literature. The first consists in inferring a sensitive attribute from a partially known record plus knowledge of a model that was trained using this record, e.g. [32, 44, 92, 110, 72, 77]. The second consists in generating a representative sample of one of the members of the training set, or one of the classes in a classification problem, by exploiting knowledge of the target model, e.g. [31, 10, 11, 108, 45, 87]. Our framework is applicable to both forms, but in this work we focus on the former, i.e., inferring sensitive information from a partially known record. The authors of [104] propose a framework that generalizes to both types of attribute inference attacks and connects them to several cryptographic notions. The notion of attribute inference is also formalized by [109]. While their work defines the advantage of an adversary as the difference between the information leaked by the model and the information present in the underlying probability distribution of the data, our formalism only allows the adversary to gain advantage from the target model. Furthermore, we consider and compare different attack strategies, while their work only focuses on the attack introduced by [32], and an attacker with oracle access to a membership inference algorithm.

Model Extraction: A third class of privacy violation consists in stealing the functionality of a model, when the model and its parameters are considered sensitive information, e.g., [97], but this setup is out of the scope of our work.

Unintended Memorization: Leakage of sensitive information might be caused by unintended memorization by the model. The authors of [18] study unintended memorization by generative sequence models. They prove that unintended memorization is persistent and hard to avoid; moreover, they find that a model can present exposure even before overfitting. This is an instance in which a model can leak sensitive information even while generalizing well.

Differential Privacy in Machine Learning: DP, introduced in [28, 29], is a widely used definition of privacy, which guarantees the safety of individuals in a database while releasing general information about the group. There have been several works in ML that use DP as a measure for privacy or use DP mechanisms for defense against inference attacks. The work [2] proposes a Differentially Private Stochastic Gradient Descent method for training neural networks. Their analysis allows them to estimate the privacy budget when successively applying noise to the model parameters during training. Later, in [111] the authors presented a comprehensive analysis of DP in ML by considering the different stages in which noise can be added to make an ML model differentially private. [49] evaluates the effectiveness and cost of DP methods for ML

in the light of inference attacks. The authors of [98] propose *Bayesian DP*, which takes into account the data distribution to provide more practical privacy guarantees, achieving the same accuracy as DP while providing better privacy guarantees on several models and datasets. Recent work [68] proposes an algorithm to "audit" the privacy of ML models, accurately computing the privacy budget necessary to prevent attacks with minimal impact on the utility of the target model. We do not consider the connection between DP and MIAs, as this is thoroughly analyzed in [109].

Federated Learning: Inference attacks that target federated systems have been investigated by [45, 6]. Privacy preserving methods specific for federated learning have been proposed by [91, 14, 89, 58]. The work [73] provides a comprehensive study of MIAs against Federated Learning models. In these setups the attacker can influence other entities during training. In our framework the attacker directly obtains the trained model; thus, our framework does not cover such cases.

Adversarial Examples and Privacy: There have been several works that combine the topics of privacy and adversarial examples. E.g. the work [94] studies the impact that securing a Machine Learning model against Adversarial Attacks has on the privacy of the model. The authors of [52] make use of Adversarial Examples as part of a defense mechanism against MIAs. The authors of [78] were the first to simultaneously address the issues of robustness and privacy, providing a complete analysis of both aspects of Deep Neural Networks (DNNs).

1.3 Preliminaries

In this section we introduce the general notation and definitions that are used throughout this work and we set up the formalism for membership, attribute and adversarial attacks. Section 1.3.1 introduces the general notation and provides a quick reference of the mathematical symbols used in this work. Section 1.3.2 introduces the learning and inference framework and necessary definitions. Section 1.3.3 defines the attack model, assumptions and capabilities of the attacker for inference attacks against machine learning models. Section 1.3.4 defines adversarial examples.

1.3.1 Notation

A random variable is indicated by upper case (e.g., X). Lower case letters indicate realizations, while calligraphic case denotes the alphabet (e.g., X takes values in $\mathcal{X}$ and x is a realization of X). A probability density function (pdf) is denoted by p (e.g., the pdf of X is denoted by p_X). A random variable, its alphabet and realizations are denoted all by the same letter unless otherwise indicated.

Bold face quantities denote vectors (e.g. $\mathbf{x}$). For a vector $\mathbf{x}$, x_i indicates its i-th component.

Expectations $\mathbb{E}[\cdot]$ are taken over all random variables inside the square brackets.

Tables 1.1 and 1.2 provide a quick reference to some of the notation used throughout the manuscript. The list is not exhaustive, but contains the symbols most used throughout this work.

1.3.2 Learning and Inference

We assume a fully Bayesian framework, where $Z = (X, Y) \sim p_{XY} \equiv p_Z$ denotes data X and according labels Y, drawn from sets $\mathcal{X}$ and $\mathcal{Y}$, respectively. The training set consists of n i.i.d. copies $\mathbf{Z} \triangleq \{Z_1, \ldots, Z_n\}$ drawn according to $\mathbf{Z} \sim p_Z^n$.

Let $\mathcal{F} \triangleq \{f_\theta \,|\, \theta \in \Theta\}$ be a *hypothesis class* of (possibly randomized) decision functions parameterized with θ, i.e., for every $\theta \in \Theta$, $f_\theta(\,\cdot\,; x)$ is a probability distribution on $\mathcal{Y}$. We will abuse notation and let $f_\theta(y; x)$ be a probability mass function (pmf) or a pdf in y for every $x \in \mathcal{X}$, depending on the context. The symbol $\hat{Y}_\theta(x)$ will be used to denote the random variable on $\mathcal{Y}$ distributed according to $f_\theta(\,\cdot\,; x)$. For a deterministic decision function $f_\theta(y; x) \in \{0, 1\}$ is a one-hot pmf for every $\theta \in \Theta$, $x \in \mathcal{X}$. We write $\hat{y}_\theta(x) \in \mathcal{Y}$, to denote a realization of $\hat{Y}_\theta(x)$.

A *learning algorithm* is a (possibly randomized) algorithm $\mathcal{A}$ that assigns to every training set $\mathbf{z} \in (\mathcal{X} \times \mathcal{Y})^n$ a probability distribution on the parameter space Θ (and, thus, also on the hypothesis space $\mathcal{F}$). We have $\mathcal{A}\colon \mathbf{z} \mapsto \mathcal{A}(\,\cdot\,; \mathbf{z})$, where $\mathcal{A}(\,\cdot\,; \mathbf{z})$ is a probability distribution on Θ. The symbol $\widehat{\theta}(\mathbf{z})$ is used to denote a random variable on Θ, distributed according to $\mathcal{A}(\,\cdot\,; \mathbf{z})$. In case of a deterministic learning algorithm, we have a pmf $\mathcal{A}(\theta; \mathbf{z}) \in \{0, 1\}$ for every training set $\mathbf{z}$ and can thus define the function $\widehat{\theta}(\mathbf{z}) = \arg\max_{\theta \in \Theta} \mathcal{A}(\theta; \mathbf{z})$, yielding the (possibly random) decision function $f_{\widehat{\theta}(\mathbf{z})}$. Let us define for use in later sections the Softmax function with temperature scaling.

Definition 1 (Softmax with temperature scaling)**.** *Let f_θ be a decision function which maps an input $x \in \mathcal{X}$ into a probability distribution on $\mathcal{Y}$. f_θ is parameterized by θ.*

The Softmax function with temperature scaling of f_θ with input x is defined by,

$$f_{\theta,\delta}(\,\cdot\,;x) = \frac{\exp\left(f_\theta(\,\cdot\,;x)/\delta\right)}{\sum_{y'\in\mathcal{Y}}\exp\left(f_\theta(y';x)/\delta\right)}\,, \tag{1.1}$$

where $\delta > 0$ is the temperature parameter.

The temperature parameter has the effect of smoothing out the distribution defined by $f_\theta(\cdot\,;x)$ when $\delta \geq 1$ or making the distribution closer to a Dirac delta when $0 < \delta < 1$. $\delta = 1$ describes the usual Softmax function.

To judge the quality of a decision function $f \in \mathcal{F}$ we require a loss function $\ell\colon \mathcal{Y}\times\mathcal{Y}\to\mathbb{R}$. We naturally extend this definition to vectors by an average over component-wise application, i.e., $\ell(\mathbf{y},\mathbf{y}') = \frac{1}{n}\sum_{i=1}^{n}\ell(y_i,y_i')$.

Definition 2 (Expected risk)**.** *We define $\varrho(\theta,(x,y)) \triangleq \mathbb{E}[\ell(\hat{Y}_\theta(x),y)]$ as the expected loss between $f_\theta(x)$ and y. This notation is naturally extended to vectors as*

$$\varrho(\theta,\mathbf{z}) \triangleq \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\ell(\hat{Y}_\theta(x_i),y_i)]. \tag{1.2}$$

The expected risk *and* empirical risk *of a learning algorithm $\mathcal{A}$ at training set $\mathbf{Z}$ are respectively defined as*[1] [2]

$$\mathcal{R}_{\exp}(\mathcal{A}) \triangleq \mathbb{E}\big[\varrho\big(\widehat{\theta}(\mathbf{Z}),(X,Y)\big)\big]\,,\ \mathcal{R}_{\mathrm{emp}}(\mathcal{A}) \triangleq \varrho\big(\widehat{\theta}(\mathbf{Z}),\mathbf{Z}\big)\,, \tag{1.3}$$

where the training set $\mathbf{Z}$ and (X,Y) are independent. The difference between expected and empirical risk is the generalization gap *$\mathcal{G}_{\mathrm{G}}(\mathcal{A})$. The expectation of this quantity is denoted by $\mathcal{E}_{\mathrm{G}}(\mathcal{A})$. These quantities are respectively defined as,*

$$\mathcal{G}_{\mathrm{G}}(\mathcal{A}) \triangleq \mathcal{R}_{\exp}(\mathcal{A}) - \mathcal{R}_{\mathrm{emp}}(\mathcal{A})\,,\ \mathcal{E}_{\mathrm{G}}(\mathcal{A}) \triangleq \mathbb{E}[\mathcal{G}_{\mathrm{G}}(\mathcal{A})]\,. \tag{1.4}$$

1.3.3 Attack Model and Assumptions

In order to make privacy guarantees for an algorithm $\mathcal{A}$, we need to specify an attacker model and the capabilities of an attacker. We will adopt a point of view of information-theoretic privacy and will not make assumptions about the computation power afforded to an attacker. We will also assume that the attacker has perfect knowledge of the underlying data distribution p_Z, as well as the algorithm $\mathcal{A}$.

In general, the goal of the attacker is to infer some property of $\mathbf{z}$ from $\widehat{\theta}(\mathbf{z})$. However, in general the attacker may have access to certain side information. This may include the specific potential member of the training set that is queried (in case of a MIA) or any additional knowledge gained by the attacker. This side information is modeled by a random variable S taking values in $\mathcal{S}$, dependent on $\mathbf{Z}$, the value of which is known to the attacker. The attacker is interested in a target (or concept) property denoted by a random variable T taking values in $\mathcal{T}$, which we assume to be discrete, which is dependent on $(\mathbf{Z},S)$. A (white box) *attack strategy* is a (measurable) function $\varphi\colon \Theta\times\mathcal{S}\to\mathcal{T}$.

[1] Note that the expectation is taken over all random quantities, i.e., $\mathbf{Z}\sim p_Z^n$, $\widehat{\theta}(\mathbf{Z})\sim\mathcal{A}(\,\cdot\,;\mathbf{Z})$, and $(X,Y)\sim p_Z$.

[2] Note that the empirical risk is computed using the training data of the algorithm.

We shall assume that S and T are independent, but not necessarily conditionally independent given $\mathbf{Z}$. This natural assumption ensures that knowledge of the side-information S does not change the prior $p_T = p_{T|S}$ of the attacker.

Definition 3. *The Bayes* success probability *of a (randomized) attack strategy φ is*

$$\mathcal{P}_{\mathrm{Suc}}(\varphi) = \mathrm{P}\{\varphi(\widehat{\theta}(\mathbf{Z}), S) = T\}. \tag{1.5}$$

We may additionally define the success probability conditioned on side information $S = s$ *as*

$$\mathcal{P}_{\mathrm{Suc}}(\varphi|s) = \mathrm{P}\{\varphi(\widehat{\theta}(\mathbf{Z}), s) = T|S = s\}. \tag{1.6}$$

Definition 4 (Membership Inference Attack)**.** *In a MIA, T is a Bernoulli variable on $\mathcal{T} = \{0, 1\}$ and J is independently, uniformly distributed on $\{1, 2, \ldots, n\}$. Then set $S = TZ_J + (1 - T)Z$, where Z_J is a random element of the training set and $Z \sim p_Z$ is independently drawn. Thus, an attacker needs to determine if $T = 1$ or $T = 0$, i.e., whether S is part of the training set or not.*

From a practical perspective, we can consider a MIA as a binary hypothesis test, in which the attacker tries to determine T according to the trained model parameters $\widehat{\theta}(\mathbf{z})$ and to the side-information S. This approach will be used later in Section 2.2 to describe a wide array of attack strategies.

Definition 5 (Attribute inference attack)**.** *We model the non-sensitive attribute by a random variable $V \in \mathcal{V}$. In this context, the input to the model is formed by the sensitive and non-sensitive attributes $X \equiv (V, T)$. Thus $\mathcal{X} \subseteq \mathcal{V} \times \mathcal{T}$. The side information given to the attacker can consist of $S = V$ or $S = (V, Y)$, depending on the attack strategy considered.*

This definition describes the case in which the attacker has access to only some features from a target sample, and aims to determine the missing features via the target model which was trained using the target sample.

For later use we define the random variable $R \triangleq \varrho(\widehat{\theta}(\mathbf{Z}), S)$, i.e., the (random) loss function evaluated at S (cf. Definition 2). A MIA using an arbitrary strategy is illustrated in Fig. 1.1.

Although, in practice, the prior distribution of the target attribute T is usually unknown, we define the optimal rejection region of an idealized attacker, having access to all other involved distributions.

Definition 6 (Most powerful test according to Neyman-Pearson lemma)**.** *In a membership inference setup (Definition 4), define, for a threshold $0 < \gamma < \infty$, the decision region*

$$\widehat{\mathcal{T}}(\gamma) \triangleq \left\{(\theta, s) \in \Theta \times \mathcal{S} : p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|1) > \gamma \cdot p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|0)\right\}. \tag{1.7}$$

By the Neyman-Pearson lemma [75], the most powerful test at threshold γ is then given by $\varphi(\theta, s) = 1$ if and only if $(\theta, s) \in \widehat{\mathcal{T}}(\gamma)$.

In Proposition 1 we will provide lower bounds on the error achieved by this decision region and make the connection to the fully Bayesian case.

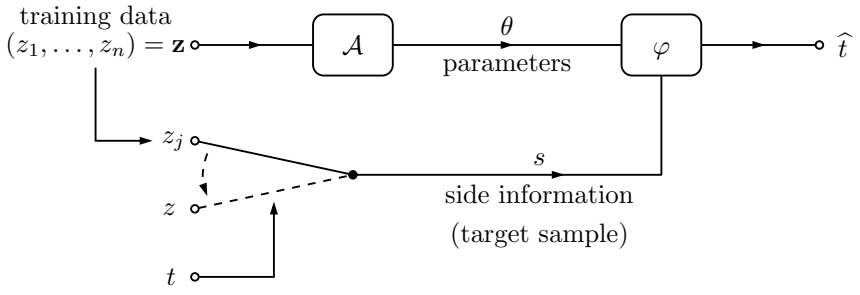


Figure 1.1: Schematic of a MIA. If $t = 1$, the target sample is drawn from the training set $\mathbf{z} = (z_1, \ldots, z_n)$ used by $\mathcal{A}$ to train the target model. If $t = 0$, the target sample is independently drawn from the data distribution. The attacker φ then uses the parameters θ at the output of $\mathcal{A}$ and the side information s to provide an estimate $\hat{t}$ of t.

1.3.4 Adversarial Examples

The framework of untargeted adversarial examples can be set as follows: Given an input $x \in \mathcal{X}$, where we assume $\mathcal{X}$ to be a vector space, and a target model $f_{\widehat{\theta}(\mathbf{z})}$, the goal of adversarial strategy $\psi_{p,\epsilon}$ is to produce some perturbation $\nu \in \mathcal{X}$ such that the prediction provided by $f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,; x + \nu)$ changes from that provided by $f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,; x)$. Additionally, we require that the perturbation ν is small.

Definition 7 (Untargeted adversarial attack)**.** *Let an untargerted adversarial strategy for a classifier $f_\theta \in \mathcal{F}$ be defined as a function $\psi_{p,\epsilon} \colon \mathcal{X} \times \mathcal{F} \to \mathcal{X}$ on the input space $\mathcal{X}$, such that for any $x \in \mathcal{X}$ and $f_\theta \in \mathcal{F}$ it obtains $\nu \triangleq \psi_{p,\epsilon}(x, f_\theta) \in \mathcal{X}$ with*

$$\arg\max_{y' \in \mathcal{Y}} f_\theta(y'; x + \nu) \neq \arg\max_{y' \in \mathcal{Y}} f_\theta(y'; x) \text{ and} \tag{1.8}$$

$$\|\nu\|_p \leq \epsilon\,, \tag{1.9}$$

i.e., the constrained perturbation ν changes the prediction of the target model to a different class.

Adversarial examples are computed constraining $\|\nu\|_p < \epsilon$, with $\epsilon > 0$ and the l_p norm $\|\cdot\|_p$ (see [3] for an extensive review on adversarial strategies). The purpose of this constraint is twofold: to perturb the original image in a way that is imperceptible for the human eye and to control the power of the attacker. In our case, the goal is not to produce subtle perturbations, the adversarial examples may be significantly different from their original counterparts. Indeed, our goal is to observe the size of the perturbation necessary to force the target model to drastically change its prediction, and use it as a criteria to distinguish members from non-members of the training set. Since $\psi_{p,\epsilon}$ will tend to compute the smallest perturbation possible such that $f_{\widehat{\theta}(\mathbf{z})}$ changes its prediction, arbitrarily high ϵ can be allowed while still observing a significant difference between the size of the perturbation of samples in and outside the training set.

For our experiments we use *Auto-Attack* to build adversarial examples[3] [22]. The Auto-Attack library offers an ensemble of different strategies to compute adversarial ex-

[3]Code available at `https://github.com/fra31/auto-attack`.

amples. Particularly, we use auto Projected Gradient Descent (Auto-PGD). Given an objective function for the adversary $\ell_a : \mathcal{X} \mapsto \mathbb{R}$ and a constraint in the form $\mathcal{S} \subset \mathcal{X}$, Auto-PGD iteratively solves $\max_{x \in \mathcal{S}} \ell_a(x)$ by applying $x^{(k+1)} = P_{\mathcal{S}}\left(x^{(k)} + \eta^{(k)} \nabla_{x^{(k)}} \ell_a(x^{(k)})\right)$, for $k = [1, \ldots, N_{\text{iter}}]$, where $P_{\mathcal{S}}$ is the projection onto the surface of $\mathcal{S}$, and typically $\ell_a(x) = \ell(y, \hat{y}_{\widehat{\theta}(\mathbf{z})}(x))$. In the original algorithm introduced by [61, 71], the step size $\eta^{(k)}$ is fixed, while Auto-PGD uses an adaptive step size which improves the performance and makes the algorithm model-agnostic.

General notation

$\mathbb{R}$	Set of real numbers
$\exp(\,\cdot\,)$	Exponential function. $\exp : \mathbb{R} \to \mathbb{R}$. Defined by the rule $\exp(x) = e^x$
$\mathrm{P}\{\,\cdot\,\}$	Probability measure. $\mathrm{P} : \mathcal{E} \to [0,1]$ is a real valued function that assigns a probability to an event in an event space $\mathcal{E}$.
$\mathbb{1}\{\,\cdot\,\}$	Indicator function. $\mathbb{1} : \mathcal{E} \to \{0,1\}$ is a random variable for an event that assigns the value 1 when the event happens and 0 when it does not happen
$\mathbb{E}[\,\cdot\,]$	Expected value. Consider a continuous random variable X with a pdf given by a function p in the real numbers. The expectation of X is given by the integral $\int_{\mathbb{R}} xp(x)dx$
$d_{KL}(\,\cdot\,\|\,\cdot\,)$	Kullback Leibler divergence. Consider pdfs p and g of a continuous random variable X. The Kullback Leibler divergence of p from g is defined by, $d_{KL}(p\|g) = \int_{\mathbb{R}} p(x)\log(p(x)/g(x))dx$
$I(\,\cdot\,;\,\cdot\,)$	Mutual information. For a jointly continuous pair of random variables (X,Y) with joint distribution p_{XY} and marginal distributions p_X and p_Y, the mutual information between X and Y is defined by $I(X;Y) = d_{KL}(p_{XY}\|p_X p_Y)$
$\|\,\cdot\,\|_p$	l_p norm. Given a vector $\mathbf{x}$ in a vector space $\mathcal{X}$ of dimension N, the l_p norm of $\mathbf{x}$ is given by $\|\mathbf{x}\|_p = (\sum_{i=1}^{N} \|x_i\|^p)^{\frac{1}{p}}$
$\mathcal{N}(\,\cdot\,;\mu,\sigma)$	Normal distribution with mean μ and variance σ^2. For a continuous random variable x, the normal distribution with mean μ and σ^2 is defined by the pdf, $\mathcal{N}(x;\mu,\sigma) = \frac{\exp(-(x-\mu)^2/2\sigma^2)}{\sqrt{2\pi\sigma^2}}$
ess sup	Essential supremum. Let f be a real valued function defined on a space $\mathcal{X}$. Let $(\mathcal{X},\Sigma,\mu)$ be a measure space with measure μ. Define $f^{-1}(a,\infty) = \{x \in \mathcal{X} : f(x) > a\}$. Define $U_f^{\text{ess}} = \{a \in \mathbb{R} : \mu(f^{-1}(a,\infty)) = 0\}$ the set of essential upper bounds. The essential supremum of f is defined by $\operatorname{ess\,sup} f = \inf U_f^{\text{ess}}$
ess inf	Essential infimum. Similarly to the essential supremum, the essential infimum of f is defined by $\operatorname{ess\,sup} f = \sup\{b \in \mathbb{R} : \mu(\{x \in \mathcal{X} : f(x) < b\}) = 0\}$

Table 1.1: Quick reference for general notation.

Learning and Inference

$Z = (X, Y)$	Data X according to label Y with X and Y taking values in $\mathcal{X}$ and $\mathcal{Y}$, respectively
$\mathbf{z}$	Realization of the training set; n i.i.d. copies $\mathbf{z} \triangleq \{z_1, \ldots, z_n\}$ drawn according to $\mathbf{Z} \sim p_Z^n$
f_θ	Decision function f parameterized by θ. For every $\theta \in \Theta$ and $x \in \mathcal{X}$ $f_\theta(\,\cdot\,; x)$ is a probability distribution on $\mathcal{Y}$
$f_{\widehat{\theta}(\mathbf{z})}$	Decision function trained on $\mathbf{z}$
$f_{\theta,\delta}$	Softmax function with temperature scaling. θ corresponds to the parameters of the decision function, δ corresponds to the temperature parameter
$\mathcal{A}$	Learning algorithm. Possibly randomized algorithm that assigns to every training set $\mathbf{z}$ a probability distribution on the parameter space Θ
ℓ	Loss function. Judges the quality of a decision function. Maps $\mathcal{Y} \times \mathcal{Y}$ to $\mathbb{R}$

Inference attacks

T	Target property or concept
S	Side-information
φ	Inference attack strategy. Maps the trained model's parameters $\widehat{\theta}(\mathbf{z})$ and side-information S into a prediction of the target concept $\widehat{T}$
γ	Threshold. $\gamma \in \mathbb{R}$
ϕ	Score criteria for a Membership Inference Attack. Maps the trained model's parameters $\widehat{\theta}(\mathbf{z})$ and side-information S into a score which is compared to a threshold γ to predict the target concept $\widehat{T}$

Table 1.2: Quick reference for notation (learning and inference).

Chapter 2

Theoretical Framework

The two main goals of this work are to provide formal guarantees on the privacy of ML models and to translate those guarantees to practical scenarios. The first section of this chapter derives formal results on the success rate of inference attacks. Bounds on the success rate of arbitrary attackers provide formal guarantees for the privacy of individuals present in the training set. If an attacker cannot identify that a particular sample is present in the training set, even when having full access to this sample and to the target model, then it is impossible for the attacker to extract any additional information. The results we derive are difficult to apply in practice, due to the impossibility to estimate the distributions of the parameters of the target model in most practical scenarios. Thus, the later sections of this chapter are dedicated to describing a wide array of strategies for membership and attribute inference attacks in a formal way. This list comprehends strategies for membership inference present in the literature, plus several novel strategies. Finally, some basic strategies for attribute inference are described to illustrate how these can fit into our framework.

Section 2.1 describes theoretical bounds on the success rate of arbitrary inference attacks. Upper bounds are important to assess the privacy of the trained model, while lower bounds show a connection to its generalization gap. Yet, generalization is shown to be a necessary, but not sufficient condition to guarantee privacy. The last part of this section provides a list of results linking the success rate of the attacker to the mutual information between the learned model parameters and the target concept.

Section 2.2 describes a list of strategies for membership inference attacks. Section 2.3 studies the connection between the problem of OOD detection and the problem of membership inference and describes how OOD detection techniques can be adapted for membership inference. Section 2.4 lists a few strategies for attribute inference that are inspired on basic membership inference strategies.

2.1 Bounds on Information Leakage

The goal of our framework is to derive theoretical results on the performance of inference attacks in the most general way possible. This gives us an overview of the problem and an understanding of the capabilities and limitations of the attacker. This section is dedicated to deriving those theoretical results and to explain their consequences.

Section 2.1.1 presents the Bayesian attacker, an attacker having access to the conditional distribution of the target concept given the trained model parameters and side information. The performance of the Bayesian attacker provides upper bounds on the success rate of arbitrary attackers having access to the same or a smaller amount of information. The results of Section 2.1.2 are specific to membership inference, and lower bound the success rate of MIAs by an expression that depends on the generalization gap of the target model. These results mean that, if the target model has a large generalization gap, there exists a MIA strategy that will succeed. Furthermore, these results depend on the characteristics of the loss function considering three different cases: bound, sub-Gaussian and exponentially tail-bounded loss functions. Section 2.1.3 shows by a suitable example that good generalization is not enough to prevent successful membership inference attacks. Section 2.1.4 provides three results. The first result links the gain of the attacker over prior knowledge to the mutual information between the target concept and the model parameters. The second result links the generalization gap to the mutual information between training set and trained model parameters. The third result connects the first two results by a relation between the two mutual information quantities.

2.1.1 Performance of the Bayesian Attacker

In this section, we establish two theorems that provide upper bounds on the success probability of an arbitrary attacker. First, consider the general case in which the target attribute T is not necessarily binary, but finite. This case includes both membership and feature inference attacks. In this case the Bayes classifier is the best possible attacker, which arises naturally from a maximum a posteriori optimization of the target attribute.

Theorem 1 (Success of the Bayesian attacker)**.** *Assume that $\mathcal{T}$ is a finite set and φ is an arbitrary attack strategy.*[1] *The Bayes success probability is upper bounded by,*

$$\mathcal{P}_{\mathrm{Suc}}(\varphi) \leq \mathbb{E}\left[\max_{t\in\mathcal{T}} p_{T|\widehat{\theta}(\mathbf{Z})S}(t|\widehat{\theta}(\mathbf{Z}), S)\right], \tag{2.1}$$

where the upper bound is achieved by the attack strategy,

$$\varphi^{\star}(\theta, s) = \arg\max_{t\in\mathcal{T}} p_{T|\widehat{\theta}(\mathbf{Z})S}(t|\theta, s)\,. \tag{2.2}$$

If the arg max *in (2.2) is not unique, any $t \in \mathcal{T}$ achieving the maximum can be chosen.*

Proof. Let $\widehat{T}$ denote the random variable defined by $\widehat{T} \triangleq \varphi(\widehat{\theta}(\mathbf{Z}), S)$. Note that $\widehat{T}$ is independent from T given $(\widehat{\theta}(\mathbf{Z}), S)$. First, the upper bound in (2.1) is shown, then

[1] As this result provides an upper bound on the success probability, no restrictions are placed on the capabilities of the attacker.

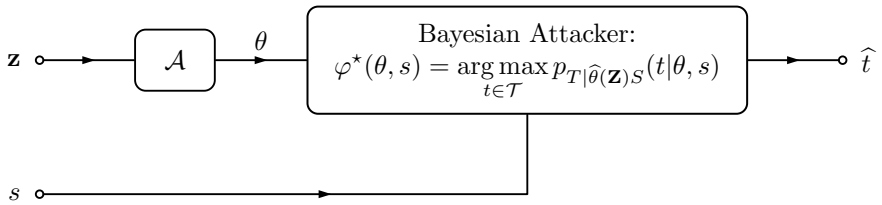


Figure 2.1: Schematic of the Bayesian attacker. The Bayesian attacker achieves the upper bound shown in Theorem 1, but needs to be able to evaluate the conditional distribution $p_{T|\widehat{\theta}(\mathbf{Z})S}$. The observations required for the attack are the side-information s and model parameters θ.

it is shown that this upper bound is achieved by (2.2). Let φ be an arbitrary attack strategy defining pdf $p_{\widehat{T}|\widehat{\theta}(\mathbf{Z})S}(\widehat{t}|\theta, s)$ for each $(\theta, s) \in \Theta \times \mathcal{S}$,

$$\begin{aligned}\mathcal{P}_{\text{Suc}}(\varphi) &= \mathbb{E}\left[\sum_{\widehat{t} \in \mathcal{T}} p_{\widehat{T}|\widehat{\theta}(\mathbf{Z})S}(\widehat{t}|\widehat{\theta}(\mathbf{Z}), S) p_{T|\widehat{\theta}(\mathbf{Z})S}(\widehat{t}|\widehat{\theta}(\mathbf{Z}), S)\right] \\ &\leq \mathbb{E}\left[\max_{t' \in \mathcal{T}} p_{T|\widehat{\theta}(\mathbf{Z})S}(t'|\widehat{\theta}(\mathbf{Z}), S)\right] . \end{aligned} \tag{2.3}$$

Now, consider an attack strategy $\varphi^\star$, such that $\varphi^\star(\theta, s)$ is in

$$\left\{t \in \mathcal{T} : p_{T|\widehat{\theta}(\mathbf{Z})S}(t|\theta, s) = \max_{t' \in \mathcal{T}} p_{T|\widehat{\theta}(\mathbf{Z})S}(t'|\theta, s)\right\} , \tag{2.4}$$

for given $\theta \in \Theta$ and $s \in \mathcal{S}$. Hence,

$$\mathcal{P}_{\text{Suc}}(\varphi^\star) = \mathbb{E}\left[\max_{t' \in \mathcal{T}} p_{T|\widehat{\theta}(\mathbf{Z})S}(t'|\widehat{\theta}(\mathbf{Z}), S)\right] . \tag{2.5}$$

Note that the bound is achieved as long as (2.4) is satisfied. □

A schema of the Bayesian attack is shown in Fig. 2.1. Given white-box access to the model and its parameters, as well as side information, the attacker (2.2) has the highest probability of successfully identifying a record in the training set. Thus, resilience against strategy (2.2) provides a strong privacy guarantee. Note that, even though S plays a very specific role in a MIA, it may contain additional samples, or any other kind of information, making Theorem 1 applicable to other setups.

Theorem 1 can also be applied to the black-box case. A black-box attack is not granted access to the parameters $\theta \in \Theta$, but only to the input-output relation $\{(x, f_\theta(x)) \mid x \in \mathcal{X}\}$ where $f_\theta \in \mathcal{F}$ is the model associated to the parameters θ. Thus, any black-box attack strategy $\varphi' : \mathcal{F} \times \mathcal{S} \to \mathcal{T}$ can be seen as a particular case of a white-box strategy defined as $\varphi(\theta, s) = \varphi'(f_\theta, s)$, and therefore the upper bound expressed by Theorem 1 still applies, since it is an upper bound for *all* strategies.

Similarly, when the attacker has access to only a subset of the parameters, it can be seen as a particular case of the attacker considered in Theorem 1, and therefore the result still applies. Section 3.2.1 illustrates how the upper bound can be computed in an artificial scenario.

The following proposition provides similar results for the membership inference problem.

Proposition 1 (Decision tradeoff)**.** *In a membership inference setup (Definition 4), let $\widehat{\mathcal{T}} \subseteq \Theta \times \mathcal{S}$ be any decision set, and define*

$$\epsilon_1(\widehat{\mathcal{T}}) \triangleq \int_{\widehat{\mathcal{T}}} p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|0)\, d\theta\, ds\ ,\ \epsilon_0(\widehat{\mathcal{T}}^c) \triangleq \int_{\widehat{\mathcal{T}}^c} p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|1)\, d\theta\, ds, \tag{2.6}$$

the average Type-I (false positive) and Type-II (false negative) error probabilities, respectively. Then,

$$\epsilon_0(\widehat{\mathcal{T}}) + \epsilon_1(\widehat{\mathcal{T}}^c) \geq 1 - \Delta, \tag{2.7}$$

where $\Delta \triangleq \left\| p_{\widehat{\theta}(\mathbf{Z})S|T=1} - p_{\widehat{\theta}(\mathbf{Z})S|T=0} \right\|_{\mathrm{TV}}$ and $\|\cdot\|_{\mathrm{TV}}$ is the total variation distance [37]. Equality is achieved by choosing $\widehat{\mathcal{T}}^\star \equiv \widehat{\mathcal{T}}(1)$ according to Definition 6. If the hypotheses are equality distributed, then the minimum average Bayesian error satisfies

$$\inf_{\varphi} \mathrm{P}\left\{ \varphi(\widehat{\theta}(\mathbf{Z}), S) \neq T \right\} = \frac{1}{2}\left(1 - \Delta\right)\ . \tag{2.8}$$

The proof of this proposition is rather lengthy and so is relegated to B. Equation (2.7), similar to (2.1), provides a lower bound for the total error of an arbitrary attacker. Equation (2.8) provides the error of the Bayesian attacker from Theorem 1 in the case where the hypotheses are equally distributed.

2.1.2 Generalization Gap and Success of the Attacker

In this section, we explore the connection between the generalization gap and the success probability of MIAs. Large generalization gap implies poor privacy guarantees against MIAs. Moreover, depending on characteristics of the loss function, the probability of success of the attacker is lower bounded by the generalization gap:

Theorem 2 (Bounded loss function)**.** *If the loss is bounded by $|\ell| \leq \ell_{\max}$, then there is an attack strategy φ for a MIA (Definition 4) such that,*

$$\mathcal{P}_{\mathrm{Suc}}(\varphi) \geq \max\left\{ P_{\mathrm{m}}, P_{\mathrm{m}}\left(\frac{|\mathcal{E}_{\mathrm{G}}(\mathcal{A})|}{2\ell_{\max}} - 1 \right) + 1 \right\}\ , \tag{2.9}$$

where $P_{\mathrm{m}} \triangleq \max_{t\in\{0,1\}} \mathrm{P}\{T = t\}$.

Proof. Recalling Definitions 2 and 4, and in particular $\varrho(\theta, (x, y)) = \mathbb{E}[\ell(\hat{Y}_\theta(x), y)]$, as well as the random variable $R = \varrho(\widehat{\theta}(\mathbf{Z}), S)$, we obtain

$$\begin{aligned} |\mathcal{E}_{\mathrm{G}}(\mathcal{A})| &= \left| \int r(p_{R|T}(r|0) - p_{R|T}(r|1))\, dr \right| \\ &\leq \int |r||p_{R|T}(r|0) - p_{R|T}(r|1)|\, dr \\ &\leq \ell_{\max} \|p_{R|T}(\cdot|0) - p_{R|T}(\cdot|1)\|_1. \end{aligned} \tag{2.10}$$

Assume w.l.o.g. that the attacker φ satisfies the condition,

$$p_{RT}(\varrho(\theta,(x,y)),\varphi(\theta,x,y)) \geq p_{RT}(\varrho(\theta,(x,y)),1-\varphi(\theta,x,y)). \tag{2.11}$$

Thus, we obtain,

$$\begin{aligned}\mathcal{P}_{\mathrm{Suc}}(\varphi) &= \frac{1}{2}\left(1+\int |p_{RT}(r,0)-p_{RT}(r,1)|\,dr\right)\\ &\geq \frac{1}{2}P_{\mathrm{m}}\|p_{R|T}(\cdot|0)-p_{R|T}(\cdot|1)\|_1+1-P_{\mathrm{m}}\\ &\geq P_{\mathrm{m}}\left(\frac{|\mathcal{E}_{\mathrm{G}}(\mathcal{A})|}{2\ell_{\max}}-1\right)+1.\end{aligned} \tag{2.12}$$

Note that the lower bound, $P_{\mathrm{m}}\left(\frac{|\mathcal{E}_{\mathrm{G}}(\mathcal{A})|}{2\ell_{\max}}-1\right)+1$, varies from $1-P_{\mathrm{m}}$ to 1, as the generalization gap increases. However, an attacker with knowledge of the prior on T can always have a success probability of at least P_{m} by guessing $\widehat{t}=\arg\max_{t\in\mathcal{T}}\mathrm{P}\{T=t\}$; therefore,

$$\mathcal{P}_{\mathrm{Suc}}(\varphi) \geq \max\left\{P_{\mathrm{m}}, P_{\mathrm{m}}\left(\frac{|\mathcal{E}_{\mathrm{G}}(\mathcal{A})|}{2\ell_{\max}}-1\right)+1\right\} \qquad \square$$

Theorem 2 indicates that strong privacy guarantees (i.e., small success probability for any attacker), imply that the generalization gap is also small. Viewed in a different way, if the generalization gap is large, there exists a membership inference strategy that will succeed with a certain probability given by (2.9). We remark that, on the other hand, ensuring that the generalization gap is small does not make a model robust against MIAs. We shall return to this important point in Section 2.1.3.

In the following, we extend the result of Theorem 2 to sub-Gaussian and exponentially tail-bounded loss functions.

Theorem 3 (Sub-Gaussian loss)**.** *In a membership inference problem (Definition 4), assume that $R=\varrho(\widehat{\theta}(\mathbf{Z}),S)$ is a sub-Gaussian random variable with variance proxy σ_R^2. For all $R_{\max}\geq r_0\triangleq\sqrt{2\sigma_R^2\log 2}$, there exists an attack strategy φ, such that,*

$$\mathcal{P}_{\mathrm{Suc}}(\varphi) \geq \max\left\{P_{\mathrm{m}}, P_{\mathrm{m}}\left(\frac{|\mathcal{E}_{\mathrm{G}}(\mathcal{A})|}{2R_{\max}}-\frac{C(R_{\max},\sigma_R)}{1-P_{\mathrm{m}}}-1\right)+1\right\}. \tag{2.13}$$

where $C(R_{\max},\sigma_R)\triangleq\exp\left(-\frac{R_{\max}^2}{2\sigma_R^2}\right)\left(1+\frac{\sigma_R^2}{R_{\max}^2}\right)$.

Proof. Given that R is a sub-Gaussian random variable with variance proxy σ_R^2, we have $\mathrm{P}\{|R|\geq r\}\leq 2e^{-\frac{r^2}{2\sigma_R^2}}$ for all $r\geq 0$ [15]. Define the random variable R_0 to have the distribution function $Q_0(r)\triangleq\mathrm{P}\{R_0\leq r\}\triangleq 1-2e^{-\frac{r^2}{2\sigma_R^2}}$ on its support $[r_0,\infty)$, where $r_0=\sqrt{2\sigma_R^2\log 2}$, i.e., the pdf of R_0 is $p_{R_0}(r)=\frac{2r}{\sigma_R^2}e^{-\frac{r^2}{2\sigma_R^2}}$. Let Q be the distribution function of $|R|$. Then, using the construction in the proof of [57, Theorem 1.104], we can write $|R|=Q^{-1}\circ Q_0(R_0)$, where Q^{-1} is the left continuous inverse of Q, noting that Q_0 is continuous. The sub-Gaussian property then implies $Q(r)=1-\mathrm{P}\{|R|\geq r\}\geq Q_0(r)$, which immediately yields $Q^{-1}\circ Q_0(r)\leq r$.

We thus have, for $R_{\max} \geq r_0$,

$$\begin{aligned}\int_{|r|\geq R_{\max}} |r| p_R(r) dr &= \int_{Q_0(r)\geq Q(R_{\max})} Q^{-1}(Q_0(r)) p_{R_0}(r) dr \\ &\leq \int_{Q_0(r)\geq Q(R_{\max})} r p_{R_0}(r) dr \\ &\leq \int_{r\geq R_{\max}} r p_{R_0}(r) dr \\ &\leq 2R_{\max} C(R_{\max}, \sigma_R) \end{aligned} \tag{2.14}$$

Following steps similar to those in (2.10),

$$\begin{aligned}|\mathcal{E}_{\mathrm{G}}(\mathcal{A})| &\leq \int_{|r|\leq R_{\max}} |r| |p_{R|T}(r|0) - p_{R|T}(r|1)|\, dr \\ &\quad + \int_{|r|> R_{\max}} |r| |p_{R|T}(r|0) - p_{R|T}(r|1)|\, dr \\ &\leq R_{\max} \|p_{R|T}(r|0) - p_{R|T}(r|1)\|_1 + \frac{2R_{\max} C(R_{\max}, \sigma_R)}{1 - P_{\mathrm{m}}}\,, \end{aligned} \tag{2.15}$$

where the last inequality follows from (2.14). Consequently,

$$\|p_{R|T}(r|0) - p_{R|T}(r|1)\|_1 \geq \frac{|\mathcal{E}_{\mathrm{G}}(\mathcal{A})|}{R_{\max}} - \frac{2C(R_{\max}, \sigma_R)}{1 - P_{\mathrm{m}}}. \tag{2.16}$$

The rest of the proof follows identically to that of Theorem 2. □

Theorem 4 (Tail-bounded loss)**.** *In a membership inference problem (Definition 4), assume that $R = \varrho(\widehat{\theta}(\mathbf{Z}), S)$ is such that $\mathrm{P}\{|R| \geq r\} \leq 2\exp(-r/2\sigma_R^2)$ for all $r \geq 0$ with some variance proxy $\sigma_R^2 > 0$. Then, for all $R_{\max} \geq r_0 \triangleq 2\sigma_R^2 \log 2$, there is an attack strategy φ such that, (2.13) holds with*

$$C(R_{\max}, \sigma_R) \triangleq \exp\left(-\frac{R_{\max}}{2\sigma_R^2}\right)\left(1 + \frac{2\sigma_R^2}{R_{\max}}\right). \tag{2.17}$$

The proof of this theorem is analogous to that of Theorem 3 and will be omitted here.

Note that in principle both Theorem 3 and Theorem 4 are applicable when the loss is bounded, since all bounded random variables are sub-Gaussian and exponentially tail-bounded; nonetheless, we expect Theorem 2 to provide a tighter bound in this case, as it certainly does for $\ell_{\max} = R_{\max}$.

In practice the distribution of the loss for a particular model is often unknown; however, it can be estimated and fitted to one of the cases presented in this section. Then, these results can be applied to measure the potential impact of generalization on the privacy leakage of the model.

2.1.3 Good Generalization is not Enough to Prevent Successful Attacks

Generalization does not imply privacy. The purpose of this section is to prove that in general the success rate of the attacker may not be directly proportional to the

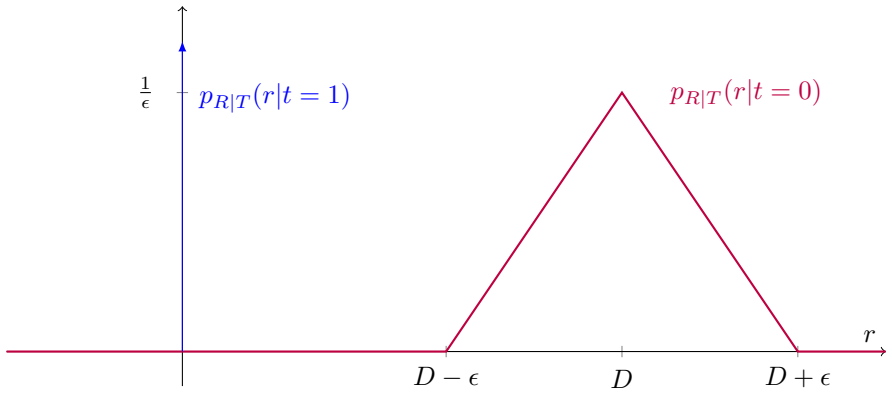


Figure 2.2: Illustration of the conditional distributions of the error incurred by the target model.

generalization gap. We show this by constructing a synthetic example of a membership inference problem, where the generalization gap can be made arbitrarily small, while T can be determined with certainty by an attacker. To construct the counterexample we need to define the random variables X, Y and a loss function ℓ for fixed parameters $0 < \epsilon < D$. Let p_X be an arbitrary continuous pdf on $\mathbb{R}$, e.g., $X \sim \mathcal{N}(0, \sigma^2)$, and define $Y = X + U$, where U is independent of X and uniformly distributed on $[-\frac{\epsilon}{2}, \frac{\epsilon}{2}]$. Given the training set $\mathbf{z}$ and an input x, the learned decision function $f(\,\cdot\,; x)$ either outputs the correct label y, if $(x, y) \in \mathbf{z}$, and otherwise $f(\,\cdot\,; x) = x + D + U'$, where U' is an i.i.d. copy of U, i.e., uniformly distributed on $[-\frac{\epsilon}{2}, \frac{\epsilon}{2}]$. With Euclidean distance loss $\ell(y, y') = |y - y'|$, these definitions immediately yield $\mathrm{P}\{R = 0|T = 1\} = 1$ and the conditional pdf

$$p_{R|T}(r|0) = \frac{1}{\epsilon}\Lambda((r - D)/\epsilon). \tag{2.18}$$

where $\Lambda(r) \triangleq \max(1 - |r|, 0)$ is the triangle distribution. The parameters $0 < \epsilon < D$ can be chosen arbitrarily. Clearly then an attacker can simply check whether $R = 0$ to determine T with probability one. On the other hand, from (2.18), it is easily verified that,

$$|\mathcal{E}_{\mathrm{G}}(\mathcal{A})| = |\mathbb{E}[R|T = 0] - \mathbb{E}[R|T = 1]| = D. \tag{2.19}$$

Thus, by varying the parameter D, we can make the generalization gap arbitrarily small, while the attacker maintains perfect success. Therefore, good generalization does not prevent the attacker from easily determining which samples were part of the training set. Remark that as NNs are universal approximators, any (reasonable) function, including the decision rule in this example, can be approximated to arbitrary degree by a NN; therefore, this behavior could be seen in practice.

2.1.4 On the Amount of Missing Information in Inference Attacks and Generalization

We aim at investigating the following simple but fundamental questions, from the perspective of information theory:

- *How much information do the model parameters $\widehat{\theta}(\mathbf{z})$ store about the training set $\mathbf{z}$? How is this information related to the generalization gap?*
- *How much information about the unknown (sensitive) attribute T is contained in the model parameters $\widehat{\theta}(\mathbf{z})$ and the side information S? And how much information is needed for the inference of T?*
- *How do the above information quantities relate or bound to each other?*

From the point of view of information theory these questions make sense only if we consider $\widehat{\theta}(\mathbf{z})$ and T as random variables, that is, attribute probabilities to the target attribute and model parameters, which is perfectly consistent with the investigated framework in this paper.

To state the following theorem, we need the *Fenchel-Legendre dual function* [12] $g^\star\colon \mathbb{R} \to \mathbb{R}$ of a function $g\colon \mathbb{R} \to \mathbb{R}$, which is defined as $g^\star(t) \triangleq \sup\{\lambda \cdot t - g(\lambda) : \lambda \in \mathbb{R}\}$. We will also use the log-moment-generating function $\psi_W\colon \mathbb{R} \to \mathbb{R}$ of a random variable W, defined as $\psi_W(\lambda) \triangleq \log \mathbb{E}[e^{\lambda W}]$. More information on these quantities and their properties are given in the discussion of the Cramér-Chernoff Method in D.1.1.

Theorem 5 (Mutual information)**.** *Let $\widehat{T} \triangleq \varphi(\widehat{\theta}(\mathbf{Z}), S)$ be the (random) prediction of any attacker φ (Definition 3). Then,*

$$I\big(T; \widehat{\theta}(\mathbf{Z}) \big| S\big) \geq d_{KL}\Big(\mathcal{P}_{\mathrm{Suc}}(\varphi) \,\Big\|\, \max_{t \in \mathcal{T}} p_T(t)\Big), \tag{2.20}$$

where $d_{KL}(p\|q)$ denotes the KL divergence between Bernoulli random variables with probabilities (p, q). Moreover, for $\epsilon \geq 0$, the generalization gap $\mathcal{E}_{\mathrm{G}}$ at $\mathbf{Z}$ satisfies

$$\mathrm{P}\big(\mathcal{G}_{\mathrm{G}}(\mathcal{A}) \geq \epsilon\big) \leq \frac{I(\mathbf{Z}; \widehat{\theta}(\mathbf{Z})) + 1}{nK(\epsilon)}, \tag{2.21}$$

where

$$K(\epsilon) \triangleq \operatorname*{ess\,inf}_{\theta \sim P_{\widehat{\theta}(\mathbf{Z})}} \psi^*_{\mathbb{E}[\varrho(\widehat{\theta},(X,Y))] - \varrho(\theta,(X,Y))}(\epsilon) \tag{2.22}$$

is an essential infimum w.r.t. $\theta \sim P_{\widehat{\theta}(\mathbf{Z})}$ of the Fenchel-Legendre dual function $\psi^\star$ of the log-moment-generating function of $\mathbb{E}[\varrho(\theta, (X, Y))] - \varrho(\theta, (X, Y))$. Furthermore,

$$I(T; \widehat{\theta}(\mathbf{Z})|S) = I(S; \widehat{\theta}(\mathbf{Z})|T) - I(S; \widehat{\theta}(\mathbf{Z})) \leq I(\mathbf{Z}; \widehat{\theta}(\mathbf{Z})) - I(S; \widehat{\theta}(\mathbf{Z})). \tag{2.23}$$

Theorem 5 is proved in D.

The mutual information expressions in (2.20) and (2.21) are related by the inequality (2.23), where $I(\mathbf{Z}; \widehat{\theta}(\mathbf{Z}))$ represents the average amount of information about the random training set $\mathbf{Z}$ retained in the model parameters $\widehat{\theta}(\mathbf{Z})$; and $I(S; \widehat{\theta}(\mathbf{Z}))$ indicates the amount of information already contained in the side information S before observing the parameters $\widehat{\theta}(\mathbf{Z})$.

From (2.23) it is clear that by controlling the average number of bits of information about the training set $\mathbf{Z}$ that the model parameters $\widehat{\theta}(\mathbf{z})$ store, i.e., $I(\mathbf{Z}; \widehat{\theta}(\mathbf{Z})) \leq r$, it is possible to control both the generalization gap in (2.21) and the accuracy of any possible attacker in (2.20). Nevertheless, a more effective defense strategy may aim directly at reducing the mutual information $I(T; \widehat{\theta}(\mathbf{Z})|S)$, which is expected to have

less severe impact on the performance of the trained model, i.e., the expected risk $\mathbb{E}\big[\ell(\hat{Y}_{\widehat{\theta}(\mathbf{Z})}(X), Y)\big]$. As (2.20) indicates, the performance of any attacker must be close to a random guess if the mutual information $I(T; \widehat{\theta}(\mathbf{Z})|S)$ is suitably small. This equation can be numerically computed to obtain an upper bound on $\mathcal{P}_{\mathrm{Suc}}(\varphi)$.

The generalization gap bound in (2.21) is subtly different from most PAC-Bayes scenarios of learning. In the present case, we are bounding the joint probability over both the training data $\mathbf{Z}$ and the randomness involved in the learning algorithm, which is within the spirit of the work by [8]. But due to the term $K(\epsilon)$, the bound presented in (2.21) is tighter.

Assuming that the loss is sub-Gaussian or bounded, it is not difficult to provide a lower bound for $K(\epsilon)$ that is independent of the underlying data distribution.

2.2 Membership Inference Attacks

Membership inference attacks (MIAs) can be used to measure the privacy leakage of ML models [90]. The more effective a MIA strategy is, the more reliably we can assess the privacy of a model. This section provides a comprehensive list of MIA strategies and these are later compared and evaluated in Section 3.3.

Throughout this section let $s = (x, y)$, the target sample, be a realisation of Z. Let $\mathbf{z}$, the training set, be a realisation of $\mathbf{Z}$. Let $\widehat{\theta}(\mathbf{z})$ be the parameters of the target model. Following Definition 4, the goal of a MIA is to determine whether a sample s belongs to the training set of target model ($T = 1$) or not ($T = 0$). The problem can be stated as a binary decision test in which the attacker predicts T according to some criteria. In the following, let ϕ be a scoring criteria, which can be a function of the target model, the target sample and any other additional information, such as additional samples from the training set that the attacker might possess. Given these inputs, ϕ outputs a real number referred to as the prediction score. This prediction score can be compared to a threshold $\gamma \in \mathbb{R}$ to determine if the test sample belongs to the training set of the target model. Formally,

$$\widehat{T} = \varphi(s, \widehat{\theta}(\mathbf{z}), *) = \begin{cases} 1 & \text{if} \quad \phi(s, \widehat{\theta}(\mathbf{z}), *) \geq \gamma \\ 0 & \text{otherwise} \end{cases} , \tag{2.24}$$

where the symbol $*$ denotes any additional inputs that the attacker might use. The hyper parameter γ selects the operating point in ROC curve. In practice, we make our analysis independent of γ by comparing performance for the whole range of possible γ values.

2.2.1 Scoring criteria

This section is dedicated to explaining the different scoring criteria existing in the literature, and those we propose.

Baseline. As a baseline on the performance of MIAs we consider a very simple strategy that predicts all misclassified samples to be outside the training set and all correctly classified samples to be in the training set. The baseline strategy can be defined as follows,

$$\phi(x, y, \widehat{\theta}(\mathbf{z})) = \begin{cases} 1 & \text{if} \quad \arg\max_{y' \in \mathcal{Y}} f_{\widehat{\theta}(\mathbf{z})}(y'; x) = y \\ 0 & \text{otherwise} \end{cases} . \tag{2.25}$$

This strategy will be favorable against models that fail to generalize, that is, classify correctly only samples in the training set. However, as models improve, more advance strategies become necessary in order to determine membership.

Softmax Response. The main claim of this strategy is that models tend to give more confident predictions over samples that belong to their training set. This strategy aims to exploit the confidence of the predictions to identify members of the training set of the target model,

$$\phi(x, \widehat{\theta}(\mathbf{z})) = \max_{y' \in \mathcal{Y}} f_{\widehat{\theta}(\mathbf{z})}(y'; x) , \tag{2.26}$$

where we make use of Definition 1. This scoring criterion has previously been used to build MIAs in [90, 74], by training an attack model using the soft probabilities output by the model, while in [94], the authors directly compare the score to a threshold. Remark that this strategy does not take into account whether the model predicts the correct class, in other words, the ground truth for the target sample is not used. Moreover, this strategy only uses the output of the target model; thus, simply querying the target model is enough to obtain the information necessary to mount the attack.

Modified Entropy. An alternative idea is to look at the uncertainty of the model. Intuitively, this should be lower for samples that were present in the training set. [93] proposes a metric called modified entropy, which decreases with the prediction probability of the correct class and increases with the prediction probability of any other class:

$$\begin{aligned}\phi(x, y, \widehat{\theta}(\mathbf{z})) = &- \left(1 - f_{\widehat{\theta}(\mathbf{z})}(y; x)\right) \log\left(f_{\widehat{\theta}(\mathbf{z})}(y; x)\right) \\ &- \sum_{y' \neq y} f_{\widehat{\theta}(\mathbf{z})}(y'; x) \log\left(1 - f_{\widehat{\theta}(\mathbf{z})}(y'; x)\right). \end{aligned} \tag{2.27}$$

Unlike previously considered metrics, modified entropy (2.27) uses the ground truth y for the target sample, taking into account whether the target model is predicting the correct class or not, and how confident it is on its prediction.

Loss. The learning objective of ML models is to minimize a loss function,

$$\phi(x, y, \widehat{\theta}(\mathbf{z})) = -\ell\left(y, \hat{y}_{\widehat{\theta}(\mathbf{z})}(x)\right), \tag{2.28}$$

over samples from the training set. Hence, we expect the value of the loss to be lower for samples in the training set. The minus sign in front of the loss is added to make this definition consistent with (2.24). An attack proposed in [109] compares the loss on the target sample to the average loss on the training set. This intuition was also exploited in [90].

Gradient Norm. The loss function is minimized via Stochastic Gradient Descent, or similar iterative optimization algorithms. Around the optimal points, the gradient of the loss function with respect to its model parameters should approach 0. This attack strategy measures the l_2 norm of the gradient of the loss function w.r.t. to the model parameters over different samples and expects this norm to be smaller for members of the training set,

$$\phi(x, \widehat{\theta}(\mathbf{z})) = -\left\|\nabla_\theta \ell\left(y, \hat{y}_{\widehat{\theta}(\mathbf{z})}(x)\right)\right\|_2^2. \tag{2.29}$$

Since we expect the norm of the gradient to be smaller for members of the training set, the minus sign is added to make this definition consistent with (2.24). This observation was first used in [90] as part of their MIA.

Although these ideas are not novel, most of them have not been used to make a binary decision test. Instead, they have been used as labeled observations to train an attack model. This requires knowledge of the member/not-member label for each sample. Our aim is to assess and compare in a systematic way the power of these observations and whether or not it is possible to perform MIAs with them without requiring a training set for the attacker.

2.2.2 Attack strategies that train an attack model

Most of the MIA strategies proposed in the literature combine sets of features of the target sample and target model. Combining these features into a single score that can be used for a binary decision test is a challenging task. A common strategy is to train a machine learning model that learns to combine these features and predict whether the target sample belongs to the training set or not. Naturally, the attack model requires a set of samples that are labeled as either part of the training set of the target model or outside of the training set of the target model. In this section we present a short review of attack models previously proposed in the literature:

Grad x and Grad w Attack Models: In [85], they propose an attack model that uses an array of statistics from the gradient of the loss of target model. The statistics considered are the l_1 norm, l_2 norm, maximum value, mean, skewness, kurtosis and absolute minimum of the gradient. These statistics are combined by a logistic regression model trained on labeled data, where the labels indicate whether the sample belongs to the training set of the target model or not. We implement this attack model and reproduce the results in our setting. When the gradient is taken with respect to the model parameters, we refer to the attack model as 'Grad w'. On the other hand, when the gradient is taken with respect to the input sample, we refer to the attack model as 'Grad x'.

Intermediate Outputs: The authors of [85] also consider an attack model that uses the intermediate outputs of the target model. For the models considered in their work, the attacker uses the outputs of the last two layers of the target model. This attack model is also implemented as described in the original paper and evaluated in our setting. This model is later abbreviated as 'Int. Outs'.

White-Box [74]: The attack model proposed by [74] utilizes the gradients of the loss function with respect to model parameters at the target sample, the value of the loss at the target sample, intermediate outputs of the target model and the one hot encoded labels of the target sample. To our knowledge this was the first work to propose using the gradient of the loss w.r.t. model parameters as a criteria to infer membership. We implement this attack strategy and reproduce the results presented in their paper. This attack strategy is referred to as WB [74].

Ensemble Attacker We propose an ensemble attacker that takes as input the Softmax response of the target model, the value of its loss function, the norm of the gradient of the loss with respect to the model parameters, the norm of the gradient of the loss with respect to the input sample, and the modified entropy. This model outperforms the state-of-the-art against AlexNet, and achieves similar performance against ResNext.

This attacker requires not only white-box access to the model, as it needs to compute gradients with respect to input and to model parameters, but it also requires a training set of its own (similarly to [74, 90, 85]). Essentially, what the attacker learns is how to map different observations to a membership label.

The attack model is a DNN with 5 fully connected layers with output sizes $40, 40, 20, 10$ and 1, respectively. The input to the network is a vector of length 6, containing the softmax response, modified entropy, loss value, gradient norm w.r.t. parameters, gradient norm w.r.t. input, and adversarial distance. These quantities are re-scaled to $[0, 1]$, which significantly improves the performance of the model. The rescaling is done according to the maximum and minimum values from the training set. The model is trained with Adam optimizer [56] for up to 300 epochs. The performance of

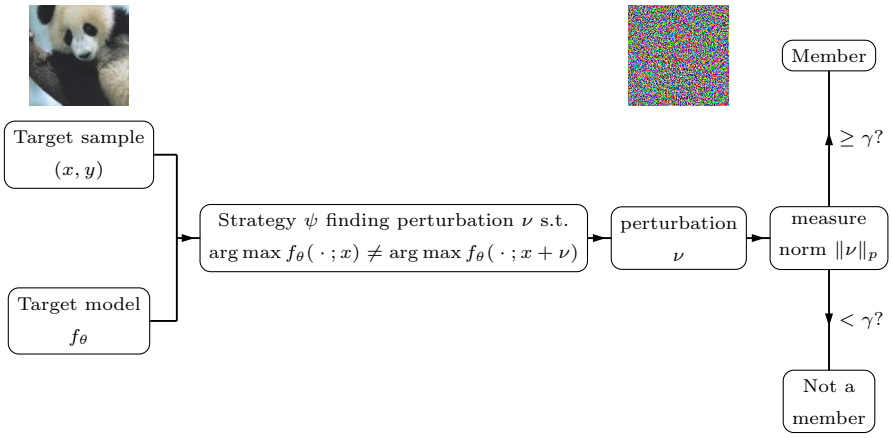


Figure 2.3: Schematic of the Adversarial Distance method for membership inference. A target sample and target model are taken as input. A perturbation is computed using an adversarial attack strategy, such that the model changes its original prediction. The magnitude of the perturbation is computed and compared to a threshold. Based on this comparison, the attacker predicts if the sample belongs to the training set of the target model or not.

the ensemble attacker is evaluated and compared to the performance of other strategies.

2.2.3 Membership inference from adversarial examples

In this section, we show how the adversarial distance bridges the gap between MIAs and adversarial examples. We introduce the adversarial distance strategy and describe the resulting algorithm in detail. Figure 2.3 illustrates the scheme for the attack. The images for the pipeline's input and adversarial noise are provided by [39]. The noise showed in the figure is obtained with the fast gradient sign method against GoogLeNet's classification algorithm. The added noise changes the classifier's output from class "*panda*" to class "*gibbon*".

During training, the target model minimizes the loss over samples from the training set. The objective of Projected Gradient Descent [71, 61] and other algorithms derived from it (e.g., Auto-PGD [22]) is to maximize the very same loss. Hence, we expect this process to require larger perturbations for members of the training set, compared to samples that were not observed during training. We exploit this feature to perform MIAs against machine learning models.

Our membership inference strategy measures the distance between an adversarial example and its original counterpart, i.e., the size of the perturbation, and uses this as a criteria to distinguish members of the training set,

$$\phi(x, \widehat{\theta}(\mathbf{z})) = \|\psi_{p,\epsilon}(x, f_{\widehat{\theta}(\mathbf{z})})\|_p \,, \tag{2.30}$$

where $\|\cdot\|_p$ measures the magnitude of the perturbation. In our experiments, we use either l_1, l_2 or l_∞ norm to measure the distance between samples (i.e., $p \in \{1, 2, \infty\}$) and the same norm is used to constraint the size of the perturbation produced by

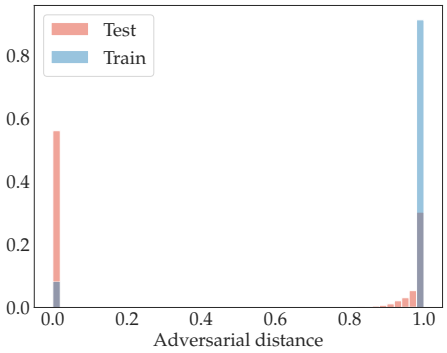


Figure 2.4: Histogram of adversarial distances over $50k$ samples from the training set (blue) superposed to the same histogram over $10k$ samples from the test set. The adversarial examples are computed for AlexNet, trained on CIFAR-100, based on the $\|\cdot\|_\infty$ norm and $\epsilon = 1$.

the adversarial strategy, guaranteeing that $\phi(x, f_{\widehat{\theta}(\mathbf{z})}) \leq \epsilon$ (see Algorithm 1). As a step previous to the computation of the adversarial example, it is verified whether the target sample is correctly classified by the target model or not. If the target sample is misclassified by the target model, then the attacker predicts that the sample is not part of the training set; otherwise, the adversarial example is computed. This assumes that the target model classifies correctly all samples in the training set and serves as a baseline for the attack.

Algorithm 1

Require: Target sample (x, y), target model $f_{\widehat{\theta}(\mathbf{z})}$, adversarial strategy $\psi_{p,\epsilon}$, $p \in \{1, 2, \infty\}$, $\epsilon > 0$ and, $\gamma \in \mathbb{R}$.

1. **if** $\hat{y}_{\widehat{\theta}(\mathbf{z})}(x) \neq y$ **then**
2. **return** 0
   \\The sample is predicted to be outside of the training set.
3. **end if**
4. $\nu \leftarrow \psi_{p,\epsilon}(x, f_{\widehat{\theta}(\mathbf{z})})$
   \\Adversarial perturbation ν.
5. **return** $\mathbb{1}\{\|\nu\|_p \geq \gamma\}$
   \\Is the distance between the adv. ex. and the original input x greater than γ?

When computing adversarial examples, we rescale the images so that their dynamic range lies within $[0, 1]$. This is necessary in order for the adversarial attacks to compute distance and perform clipping properly. However, since the pre-trained models were trained on the natural images (previous to rescaling), we include an additional layer at the input of each target model that reverts the scaling, preserving the performance of the target model.

Since we are not interested in producing subtle perturbations that preserve the perspective of a human, we let the adversarial attacker generate arbitrarily large

perturbations (constrained only by the dynamic range of the image). However, as shown in Fig. 2.4 and as demonstrated in the experimental section, there is a significant shift in the distribution of the size of perturbations, depending on whether (or not) the samples are part of the training set.

2.2.4 Diversity Measures

A diversity coefficient is map from a space of probability distributions into the real line, which reflects differences between individual members of the same population [82]. In this section, we propose the use of diversity coefficients for membership inference. The output of the target model in a MIA can be seen as a distribution over the set of labels $\mathcal{Y}$ given an input to the model. Following this remark, a diversity coefficient for the categorical distribution defined by the target sample is computed. This diversity coefficient is used as a score criteria for membership inference. The way to compute the diversity coefficient is determined by a diversity measure. In the present work, we take a metric learning approach and learn a diversity measure that suits our problem. The aim of this strategy is to learn a diversity measure that minimizes the diversity coefficient for samples in the training set and maximizes the same quantity for samples outside the training set. Therefore, members of the training set of the target model will define a categorical distribution with a lower diversity coefficient.

Let $f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x)$, the output of the target model, define a probability distribution over $\mathcal{Y}$ given $x \in \mathcal{X}$. Let $\hat{Y}_{\widehat{\theta}(\mathbf{z})}(x)$ denote the random variable in $\mathcal{Y}$ distributed according to $f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x)$. In this setup, the set $\mathcal{Y}$ is finite, thus $f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x)$ defines a categorical distribution, meaning that $\hat{Y}_{\widehat{\theta}(\mathbf{z})}(x)$ can be in one of $|\mathcal{Y}|$ possible categories, with the probabilities for each category separately specified. Since $\mathcal{Y}$ is finite, we will abuse notation and take $\mathcal{Y}$ to refer to the set of integers from 1 to $|\mathcal{Y}|$ when needed, i.e., $\mathcal{Y} = \{1, \ldots, |\mathcal{Y}|\}$. In the following, we define the diversity coefficient for a categorical distribution $\mathbf{p}$ and the diversity measure used to compute it.

Given a categorical distribution $\mathbf{p}$, its diversity coefficient is given by,

$$\mathrm{DIVC}\left(\mathbf{p}; M\right) = \sum_{i,j \in \mathcal{Y}} p_i M_{ij} p_j \ , \tag{2.31}$$

where M_{ij} corresponds to the elements of a $|\mathcal{Y}| \times |\mathcal{Y}|$ matrix M which defines a diversity measure. Since, $f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x)$ is a categorical distribution, we can use (2.31) to compute its diversity coefficient. Intuitively, M should provide a notion of distance between the possible values of the random variable $\hat{Y}_{\widehat{\theta}(\mathbf{z})}(x)$. Thus, the diversity coefficient provides the average difference between two randomly drawn instances of $\hat{Y}_{\widehat{\theta}(\mathbf{z})}(x)$.

As mentioned above, the idea of this strategy is to learn a diversity measure M such that the diversity coefficient is minimal for members of the training set and maximal for samples outside of the training set. This problem can be posed as a Lagrangian optimization problem and solved in closed form. Consider a subset of the training set $\{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbf{z}$ and a subset of the test set $\{(x'_1, y'_1), \ldots, (x'_N, y'_N)\} \cap \mathbf{z} = \emptyset$, where N is the number of samples in each set. The optimization problem can be written as,

$$\min_{M \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}} \left(\sum_{k=1}^{N} \text{DIVC}\left(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,; x_k); M \right) - \sum_{k=1}^{N} \text{DIVC}\left(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,; x'_k); M \right) \right) \tag{2.32}$$

$$\sum_{i,j \in \mathcal{Y}} M_{ij}^2 = 1 \tag{2.33}$$

$$M_{ij} > 1 \quad \forall \quad i, j \in \mathcal{Y} \quad , \tag{2.34}$$

where (2.32) is the function to minimize, (2.33) imposes a constraint on the Frobenius norm of M and (2.34) imposes that the elements of M be positive. The Lagrangian function to be optimized is,

$$\begin{aligned} \mathcal{L}(M) = & \sum_{k=1}^{N} \text{DIVC}\left(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,; x_k); M \right) - \sum_{k=1}^{N} \text{DIVC}\left(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,; x'_k); M \right) \\ & - \lambda \left(\sum_{i,j \in \mathcal{Y}} M_{ij}^2 - 1 \right) - \sum_{i,j \in \mathcal{Y}} \mu_{ij} M_{ij} \ , \end{aligned} \tag{2.35}$$

where λ and μ_{ij} for all $i, j \in \mathcal{Y}$ are Lagrange multipliers. The first term corresponds to the minimization of the diversity coefficient over the subset of samples from the training set, while the second term corresponds to the maximization of the diversity coefficient over samples outside the training set. The third term imposes a constraint over the Frobenius norm of the matrix M, and the last term of the equation imposes that the elements of M be positive. The former constraint is necessary to arrive to a unique solution, while the latter constraint is imposed so that the diversity coefficient can be interpreted as a loss function.

The optimization problem described by (2.35) can be solved in closed form with the usual methods, obtaining the following solution,

$$\mu_{ij} = \begin{cases} 0 \quad \text{if} \quad \sum_{k=1}^{N} f_{\widehat{\theta}(\mathbf{z})}(i; x_k) f_{\widehat{\theta}(\mathbf{z})}(j; x_k) - \sum_{k=1}^{N} f_{\widehat{\theta}(\mathbf{z})}(i; x'_k) f_{\widehat{\theta}(\mathbf{z})}(j; x'_k) \geq 0 \\ \sum_{k=1}^{N} f_{\widehat{\theta}(\mathbf{z})}(i; x_k) f_{\widehat{\theta}(\mathbf{z})}(j; x_k) - \sum_{k=1}^{N} f_{\widehat{\theta}(\mathbf{z})}(i; x'_k) f_{\widehat{\theta}(\mathbf{z})}(j; x'_k) \quad \text{otherwise} \end{cases} \tag{2.36}$$

$$\lambda = \frac{1}{2} \sqrt{\sum_{i,j \in \mathcal{Y}} \left(\sum_{k=1}^{N} f_{\widehat{\theta}(\mathbf{z})}(i; x_k) f_{\widehat{\theta}(\mathbf{z})}(j; x_k) - \sum_{k=1}^{N} f_{\widehat{\theta}(\mathbf{z})}(i; x'_k) f_{\widehat{\theta}(\mathbf{z})}(j; x'_k) - \mu_{ij} \right)^2} \tag{2.37}$$

$$M_{ij} = \frac{1}{2\lambda} \left(\sum_{k=1}^{N} f_{\widehat{\theta}(\mathbf{z})}(i; x_k) f_{\widehat{\theta}(\mathbf{z})}(j; x_k) - \sum_{k=1}^{N} f_{\widehat{\theta}(\mathbf{z})}(i; x'_k) f_{\widehat{\theta}(\mathbf{z})}(j; x'_k) - \mu_{ij} \right) \tag{2.38}$$

The above definition of the Lagrangian multipliers μ_{ij} guarantees that the elements of the matrix M be positive. The definition of λ imposes that M has Frobenius norm equal to one. Note that this method requires additional samples from the training set as well as additional samples from outside the training set. However, we have not yet considered the use of the ground truth for each sample. To exploit this information, we consider a solution per class, that is, we learn a different matrix M_y for each class

$y \in \mathcal{Y}$. To compute this matrix, change the sums in the above equations from $\sum_{k=1}^{N}$ to $\sum_{k:y_k=y}$, meaning the sum over the samples is only over those samples that belong to class y.

Finally, the MIA strategy proposed has the following score criteria,

$$\phi(x, y, \widehat{\theta}(\mathbf{z})) = \text{DIVC}\left(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,; x); M_y\right) \ . \tag{2.39}$$

As we can observe in Fig. 2.5, this Lagrangian leads to learning a diversity measure that has every element close to zero, except for the element along the diagonal corresponding to class y, which is close to one. In this case, we observe that the computed diversity coefficient is proportional to the cross-entropy loss. Note, that the diversity coefficient in this case will be approximately $\left(f_{\widehat{\theta}(\mathbf{z})}(y; x)\right)^2$. Therefore, both the diversity coefficient and the cross-entropy loss are monotonously increasing functions of $f_{\widehat{\theta}(\mathbf{z})}(y; x)$, the component of the categorical distribution output by the model corresponding to the ground truth y. This means that from the perspective of a binary hypothesis test, both the cross-entropy loss and the diversity coefficient are equivalent score criteria. As we will see in the experimental section, the performance of the diversity coefficient attack and the performance of the loss attack are very close, as is to be expected. Surprisingly, having additional information in this case does not provide the attacker with an advantage over the case where no additional samples are available, since the attacker learns essentially the same information that would be provided by a simple loss computation. The fact that additional information and resources do not provide an advantage to the attacker is a trend that we saw across all the different strategies we tested in our experiments.

Since using the Lagrangian given by (2.35) does not provide any advantage over the loss computation, we propose to learn a different diversity measure using a different objective. Instead of trying to minimize the diversity coefficient given by (2.31) over samples in the training set and maximizing the same quantity over samples outside the training set, we directly minimize the error incurred by using (2.31) as a score criteria to perform MIAs. Empirically, we found that the diversity measure learnt with this method produces diversity coefficients for samples inside the training set larger than for samples outside the training set. Based on this observation, we change our previous assumptions and predict that samples ins the training set of the target model will have a larger diversity coefficient compared to samples outside of the training set. Additionally, we relax the condition that the elements of the diversity measure matrix M should be positive. The new optimization problem can be written as,

$$\begin{aligned} \min_{M \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}\,,\,\gamma\in\mathbb{R}} \Bigg(& \sum_{k=1}^{N} \max\left\{0\ ,\ \gamma - \text{DIVC}\left(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,; x_k); M\right)\right\} \\ & + \sum_{k=1}^{N} \max\left\{0\ ,\ \text{DIVC}\left(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,; x'_k); M\right) - \gamma\right\} \Bigg) \end{aligned} \tag{2.40}$$

$$\sum_{i,j\in\mathcal{Y}} M_{ij}^2 = 1 \quad , \tag{2.41}$$

where the goal in (2.40) is to minimize the error made by using the diversity coefficient as a score criteria for membership inference and (2.41) imposes that the Frobenius norm

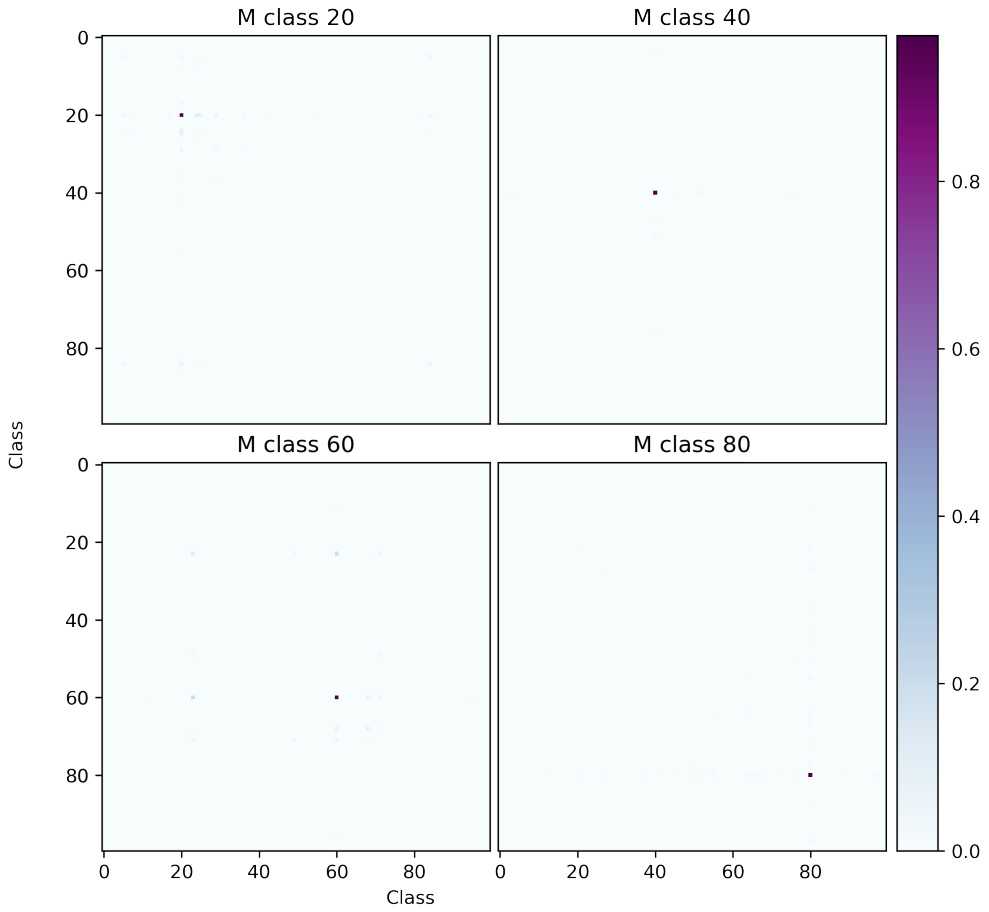


Figure 2.5: Diversity measure matrices determined for different classes. The matrices are computed from the outputs of an AlexNet model trained on the CIFAR-100 dataset. The attacker has access to 80 samples from each class to compute the matrices for the attack. Half of these samples are from the training set and half of them are from outside the training set. From left to right, top to bottom, the corresponding classes are 20, 40, 60 and, 80.

of M be positive. The Lagrangian function to be optimized is,

$$\begin{aligned}\mathcal{L}(M,\gamma) = &\sum_{k=1}^{N} \max\left\{0\ ,\ \gamma - \mathrm{DIVC}\left(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x_k);M\right)\right\}\\ &+\sum_{k=1}^{N} \max\left\{0\ ,\ \mathrm{DIVC}\left(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x'_k);M\right) - \gamma\right\}\\ &-\lambda\left(\sum_{i,j\in\mathcal{Y}} M_{ij}^2 - 1\right)\ ,\end{aligned} \tag{2.42}$$

where the optimization is now also over the threshold γ. The first term in this equation represents the error incurred by classifying a member of the training set as a non-member. Note that if the diversity coefficient is larger than the threshold, the sample is classified as a member of the training set. Thus in the first term, where the sum is over samples from the training set, the samples that are correctly classified will add null to the sum of the error. Similarly, the second term corresponds to the error incurred by classifying samples outside the training set as members. When the diversity coefficient is smaller than the threshold the samples are classified to be outside the training set. Thus in the second term, where the sum is over samples outside of the training set, the samples that are correctly classified will add null to the sum of the error. The last term puts a constraint on the Frobenius norm of M. Notably, we remove the positivity constraint on the elements of M.

In contrast to (2.35), (2.42) does not have a close form solution. This is notably due to the presence of the max operation inside the sums in the first two terms. In consequence, we need to look for an alternative optimization method. We propose to solve the problem iteratively through Stochastic Gradient Descent. Algorithm 2 explains how the learning of M is performed. In practice, the max operation in (2.42) is replaced by a ReLU activation function. Additionally, we will require the membership labels of samples t in order to write the loss function to be minimizing during training. Following the established convention, $t = 1$ corresponds to the sample being part of the training set, while $t = 0$ corresponds to the sample being outside to the training set. Thus, the loss used for learning the diversity measure that minimizes the error is given by,

$$\begin{aligned}\ell_{\mathrm{DIVC}}(x,t;M,\gamma) = &\, t\,\mathrm{ReLU}\left\{\gamma - \mathrm{DIVC}\left(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x);M\right)\right\}\\ &+(1-t)\,\mathrm{ReLU}\left\{\mathrm{DIVC}\left(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x);M\right) - \gamma\right\}\ .\end{aligned} \tag{2.43}$$

The constraint on the Frobenius norm of M is imposed iteratively during training by dividing M by its norm.

In practice, we fix the threshold at 0 and optimize only the diversity measure. The diversity measure M is randomly initialized by drawing independently each component from a normal distribution. This initialization is compatible with the choice of the threshold used for training, allowing the loss to be optimized for both samples in and outside of the training set. The diversity measure M is trained for N_{iter} epochs, and saved after each epoch. At the end of training, the best M value is chosen across different epochs based on its performance over a validation set. At each epoch the

diversity measure is updated iteratively for each minibatch by setting its Frobenius norm to one and then applying the Adam [56] optimizer. In our experiments N_{iter} is set to 1, the batch size for minibatches is set to 128 and the learning rate parameter for Adam is 0.01. The training set for the attacker is balanced in terms of the number of members and non-members of the training set of the original model. It is also balanced in terms of the classes for the original classification task.

Algorithm 2 Learning of the divergence measure

Require: Training set samples $\{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbf{z}$, test set samples $\{(x'_1, y'_1), \ldots, (x'_N, y'_N)\} \cap \mathbf{z} = \emptyset$, target model $f_{\widehat{\theta}(\mathbf{z})}$, threshold γ, learning rate η, number of iterations N_{iter}.

1. Initialize $M^{(0)}$ with each component drawn randomly independently from the same normal distribution $\mathcal{N}(\cdot; 0, 1)$
2. **for** k in $[N_{\text{iter}}]$ **do**
3. Split $\{(x_1, y_1), \ldots, (x_N, y_N)\} \cup \{(x'_1, y'_1), \ldots, (x'_N, y'_N)\}$ in Minibatches
4. $M \leftarrow M^{(k-1)}$
5. **for** Minibatch in Minibatches **do**
6. $M \leftarrow M - \eta \nabla_M \left(\sum_{(x,t) \in \text{Minibatch}} \ell_{\text{DIVC}}(x, t; M/\|M\|_2, \gamma) \right)$
7. $M \leftarrow M/\|M\|_2$ \\Set the norm to 1.
8. **end for**
9. $M^{(k)} \leftarrow M$
10. **end for**
11. $M \leftarrow$ Select the best $M^{(k)}$, with $k \in [N_{\text{iter}}]$
12. **return** M

Finally, this solution can be improved by considering a separate diversity measure for each class $y \in \mathcal{Y}$. In our experiments, the training procedure described above is performed separately for each class to produce a diversity measure M_y for each $y \in \mathcal{Y}$. In contrast to the closed form solution obtained by minimizing the average diversity coefficient over samples of the training set and maximizing the average diversity coefficient over samples outside of the training set (2.35), minimizing the error provides a very distinct metric with respect to the cross-entropy loss. In Fig. 2.6 we can observe that the matrices learned are very distinct from those obtained by computing the closed form solution. In the benchmark section, we will see that this solution achieves a similar performance to the state-of-the-art; however, it does not provide an advantage to the attacker despite the attacker having access to additional samples in this setup.

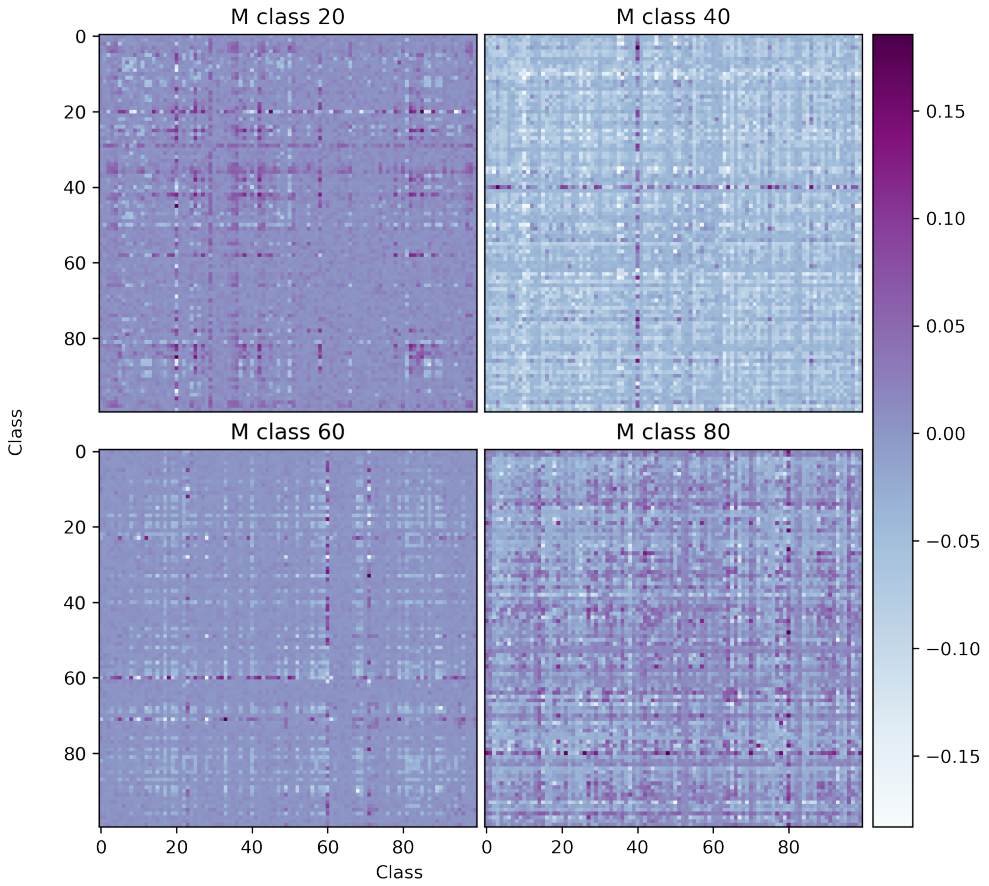


Figure 2.6: Diversity measure matrices determined for different classes. The matrices are learned by minimizing the error loss using as a training set for the attacker the outputs of an AlexNet model trained on the CIFAR-100 dataset. The attacker has access to 80 samples from each class to compute the matrices for the attack. Half of these samples are from the training set and half of them are from outside the training set. From left to right, top to bottom, the corresponding classes are 20, 40, 60 and, 80.

2.2.5 Renyi Divergence

The Renyi alpha-divergence generalizes the Kullback-Leibler divergence and other notions of divergence. It allows to measure the divergence of a given probability distribution from another. We propose the use of this divergence for membership inference. The intuition is that samples from the training set belonging to the same class will produce similar outputs when fed to the target model. Since these outputs can be interpreted as categorical probability distributions, we can measure the divergence between the output at the target sample and the output at a reference sample belonging to the training set and use this value as a score criteria to perform MIAs. In other words, the smaller the value of the Renyi divergence of the target sample from the reference, the more likely the target sample is to be part of the training set. In the following we define the Renyi alpha-divergence for two generic categorical distributions. Let $\mathbf{p}$ and $\mathbf{q}$ be two categorical distributions. The Renyi alpha-divergence of $\mathbf{p}$ from $\mathbf{q}$ is given by,

$$d_\alpha\Big(\mathbf{p}\Big\|\mathbf{q}\Big) = \frac{1}{\alpha-1}\sum_{i\in\mathcal{Y}}\frac{p_i^\alpha}{q_i^{\alpha-1}}\ , \tag{2.44}$$

where $0<\alpha<\infty$ and $\alpha\neq 1$. Remark that this quantity is not symmetric on the two probability distributions. This quantity is equal to zero if and only if $\mathbf{p}$ is equal to $\mathbf{q}$ and otherwise positive. The special cases where $\alpha = 0, 1$ and $\alpha\to\infty$ are given by computing the respective limits. In particular the case where $\alpha = 1$ leads to the Kullback-Leibler divergence.

Let $f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x)$ be the probability distribution defined by the output of the target model $f_{\widehat{\theta}(\mathbf{z})}$ at input x. Given two different inputs x and x', we can compute the Renyi alpha-divergence of $f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x)$ from $f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x')$,

$$d_\alpha\Big(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x)\Big\|f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x')\Big) = \frac{1}{\alpha-1}\sum_{i\in\mathcal{Y}}\frac{f^\alpha_{\widehat{\theta}(\mathbf{z})}(i;x)}{f^{\alpha-1}_{\widehat{\theta}(\mathbf{z})}(i;x')}\ , \tag{2.45}$$

In this case x refers to the target sample and x' to the reference sample. Note that the target model could produce equal or very similar outputs even if x is different than x'. Thus $d_\alpha\Big(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x)\Big\|f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x')\Big) = 0$, does not imply $x = x'$.

In order to use this quantity as a score criteria for membership inference, we need to choose a reference sample from the training set. This implies that the attacker needs to have access to a subset of the training set of the target model. The attacker takes only reference samples from the same class as the target sample. Then the attacker measures the Renyi alpha-divergence of the target sample from every possible reference sample and takes the minimum of those values as the score criteria for the attack. I.e. the attacker uses as score criteria the smallest divergence value of the target sample from the reference samples in the subset of the training set. Formally, the score criteria for the attack is,

$$\phi(x,y,\widehat{\theta}(\mathbf{z})) = \min_{(x',y')\in\mathbf{z'}_{\text{train}}:y'=y} d_\alpha\Big(f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x)\Big\|f_{\widehat{\theta}(\mathbf{z})}(\,\cdot\,;x')\Big)\ , \tag{2.46}$$

where $\mathbf{z'}_{\text{train}}\subset\mathbf{z}$ denotes the subset of the training set of the target model that the attacker possesses. Empirically, we determined $\alpha = 0.0005$ to be the value that

maximizes the AUROC score of the attack. To determine this value, we performed a grid search and computed the AUROC score for each value of α. Note that this attack falls into the category of attacks that require additional samples. It requires extra samples from the training set to be used as reference samples and extra samples from both inside and outside the training set of the target model in order to determine the best value for α.

In our experiments, we observed this strategy to be effective, achieving comparable performance, or even improving over the state-of-the-art. However, the improvement is marginal and might not be justified considering there is other strategies that do not require the use of additional samples.

2.2.6 Merlin

Merlin, which stands for *Measuring Relative Loss in Neighbourhood*, is a strategy for membership inference developed in [50]. The strategy consists in perturbing the input sample to observe how this changes the values of the loss over the sample. This process is repeated a certain number of iterations, and the number of times that the loss increases is counted. If the loss increases more frequently than it decreases, then the sample is predicted to be part of the training set. The intuition is that, since the loss function should be minimized over samples that belong to the training set, perturbing these samples should increase the value of the loss. Algorithm 3 describes the attack strategy. The number of iterations N_{iter} and the value of σ are taken as 100 and 0.01, respectively, which are the original values used in [50].

Algorithm 3 Merlin

Require: Number of iterations N_{iter}, target model $\hat{y}_{\widehat{\theta}(\mathbf{z})}$, standard deviation σ, target sample (x, y), threshold γ

1. $count \leftarrow 0$
2. **for** N_{iter} **do**
3. $\quad w \leftarrow \mathcal{N}(\,\cdot\,; 0, \sigma^2)$ \\Draw random noise from a Gaussian distribution.
4. $\quad$ **if** $\ell(\hat{y}_{\widehat{\theta}(\mathbf{z})}(x+w), y) > \ell(\hat{y}_{\widehat{\theta}(\mathbf{z})}(x), y)$ **then**
5. $\quad\quad count \leftarrow count + 1$
6. $\quad$ **end if**
7. **end for**
8. **return** $\mathbb{1}\{\, count/N_{\text{iter}} \geq \gamma\}$ \\Threshold check

Algorithm 3 outputs 1 if the attack is successful and 0 if the attack fails. In our experiments, we found this strategy to be ineffective, achieving a performance similar to a random guess. [50] proposes an improvement to this attack, *Morgan*, which involves adjusting two different thresholds. We do not consider this strategy, as our analysis is threshold independent and adjusting two different thresholds would lead to an unfair comparison to other methods.

2.3 MIAs from Out-of-Distribution detection techniques

2.3.1 Why use OOD detection techniques?

In this section, we introduce the use of OOD detection techniques for membership inference. We discuss the similarities between the two problems, and why it makes sense to pose membership inference as a OOD detection problem.

Consider the output of the target model $\hat{Y}_{\widehat{\theta}(\mathbf{z})}(X)$, with input X a random variable given by $(X, Y) = S$ and the random variable S given by Definition 4 as $S = TZ_J + (1-T)Z$. Remark that T determines whether S is Z_J, a member of the training set, or $Z \sim p_Z$, independently drawn. In this setup, the randomness in the observation $\hat{Y}_{\widehat{\theta}(\mathbf{z})}(X)$ comes from the fact that the input is random X, from the learning algorithm and from the fact that the training set $\mathbf{Z}$ is random.

In terms of the underlying probability distribution of the target model's output, there is a fundamental difference between the case where $T = 1$, which corresponds to the target sample belonging to the training set, and the case where $T = 0$, making and the training set and the target sample independent. Therefore, it might be possible to identify a shift in terms of the distributions of $\hat{Y}_{\widehat{\theta}(\mathbf{z})}(X)$ when $T = 1$, i.e. X is part of the training set and when $T = 0$, i.e. X is not in the training set. Interestingly enough, this observation is not only true for the output of the model $\hat{Y}_{\widehat{\theta}(\mathbf{z})}(X)$, but for any other function $\phi(S, \widehat{\theta}(\mathbf{z}))$ of the input sample, its label and the target model. For example, if we consider ϕ to be the output of an intermediate layer of the model, we could observe a similar shift in the distribution of this random variable. This is precisely what we exploit in the following section to perform membership inference attacks.

Note that both MIAs and OOD detection can be posed as binary decision problems. In membership inference we try to determine if the target sample belongs to the training set of the target model or not; while in OOD detection we try to determine whether the target sample is 'in-distribution' or 'out-of-distribution'. This similarity permits to directly apply many of the existing methods for OOD detection to the problem of membership inference. However, as we observed in our experiments, applying these techniques directly does not often lead to good results for the attacker. Significant work is required to adapt these techniques, showing that there is a fundamental difference between these problems.

The next subsections provide detailed descriptions of the state of the art methods for OOD detection and how they can be adapted to launch MIAs.

2.3.2 ODIN Membership Score

ODIN [65] represents an adaptation of the *softmax response* framework for selective classification [36] to the problem of OOD sample detection. Consider a model $f_{\widehat{\theta}(\mathbf{z})}$ and a sample $x \in \mathcal{X}$. In this context consider the output of the model to be the logits. The core idea of ODIN is to leverage the information contained in the maximum of the model's *softmax probabilities*, on which the decision is based, by comparing it to a suitable threshold in order to tell in-distribution and OOD samples apart. Using

Definition 1, tThe scoring criterion for a MIAs is,

$$\phi(x, \widehat{\theta}(\mathbf{z})) = \max_{y' \in \mathcal{Y}} f_{\widehat{\theta}(\mathbf{z}),\delta}(y'; x) \ , \tag{2.47}$$

where the temperature δ is a parameter of the attack. The temperature parameter can have the effect to smooth out the distribution of the outputs of the target model when $\delta \geq 1$, making it closer to a uniform distribution. On the other hand, when $\delta < 1$, the distribution becomes more peaked.

The intuition is that a model will be more confident on a sample it has already seen at training time. Note that, save for the temperature scaling, this strategy is identical to the *softmax response* strategy defined by (2.26). Just like *softmax response*, ODIN uses the confidence in the prediction of the model without taking into account the ground truth.

By carefully choosing a threshold $\gamma \in [0, 1]$, and interpreting $\phi(x, \widehat{\theta}(\mathbf{z}))$ as the confidence of the target model when predicting the class of the target sample, ODIN proposes to label x as in-distribution if $\phi(x, \widehat{\theta}(\mathbf{z})) > \gamma$, or as out-of-distribution otherwise. It is straightforward to draw a parallelism between in-distribution and member samples on the one side, and out-of-distribution and non member samples on the other, hence proposing the application of such a technique to the MIA problem.

2.3.3 DOCTOR Membership Score

Another method that can be used to identify members of the training set is the DOCTOR membership score. Although it has neither been designed nor optimized for OOD detection, it proves to be very effective in the misclassification detection problem [40], i.e. telling correctly and incorrectly classified samples apart. Though similar to (2.47), the DOCTOR Membership score, formalized as,

$$\phi(x, \widehat{\theta}(\mathbf{z})) = \sum_{y' \in \mathcal{Y}} \left(f_{\widehat{\theta}(\mathbf{z}),\delta}(y'; x) \right)^2 \ , \tag{2.48}$$

uses all the softmax probability components. With δ the parameter for the temperature scalling. Crucially, the score above is closely related to the Rényi divergence [101] between the model's output distribution and the uniform distribution over the classes, indeed gauging the self-uncertainty of the model. Much like the previous technique, it is trivial to draw a parallellism between the OOD detection and the membership inference problem by assigning the in-distribution samples to member samples on the one side, and the out-of-distribution samples to non member samples on the other.

2.3.4 Mahalanobis Membership Score

First proposed in [63] for the problem of OOD detection, we adapt this technique for membership inference. This method measures the distance from the candidate sample to two empirical distributions, the first corresponding to training samples and the second corresponding to samples outside the training set. The assumption made by this method is that the intermediate outputs of the target model follow a multivariate class-conditional Gaussian distribution.

Let $\hat{y}_{\widehat{\theta}(\mathbf{z})\,l}(x)$ denote an intermediate output of the model, indexed by $l \in \{1, \ldots, L\}$, with L the number of layers of the target model. In this setup, the attacker is given two set of samples, the first a subset of the training set and the second a subset of the test set of the target model. For ease of notation, $\{(x_i, y_i) : i \in \{1, \ldots, N\}\}$ will refer to either of these sets in the following. For the l-th layer and class $y \in \mathcal{Y}$, the parameters of the empirical distribution of the target model's output are computed as,

$$\widehat{\mu}_{y,l} = \frac{1}{N_y} \sum_{i:y_i=y} \hat{y}_{\widehat{\theta}(\mathbf{z})\,l}(x_i) \ , \tag{2.49}$$

$$\widehat{\Sigma}_l = \frac{1}{N} \sum_{y \in \mathcal{Y}} \sum_{i:y_i=y} (\hat{y}_{\widehat{\theta}(\mathbf{z})\,l}(x_i) - \widehat{\mu}_{y,l})(\hat{y}_{\widehat{\theta}(\mathbf{z})\,l}(x_i) - \widehat{\mu}_{y,l})^\top \ , \tag{2.50}$$

where N_y is the number of samples from class y used to estimate the parameters and $N = \sum_y N_y$. The mean $\widehat{\mu}_{y,l}$ is estimated per class, while the covariance matrix $\widehat{\Sigma}_l$ is averaged over all classes. In our setup, the attacker is given a small number of samples N to estimate $\widehat{\Sigma}_l$; thus the error incurred by trying to estimate a covariance matrix per class would be too high. Note that in contrast to other methods, this one requires additional samples from the training set and from outside the training set to estimate the distributions of training and outside the training set samples, respectively.

The *Mahalanobis distance-based confidence score* [63], provides a notion of distance between a single sample and the class-conditional Gaussian distribution defined above,

$$M_l(x) = \max_{y \in \mathcal{Y}} \left[-(\hat{y}_{\widehat{\theta}(\mathbf{z})\,l}(x) - \widehat{\mu}_{y,l})^\top \widehat{\Sigma}_l^{-1} (\hat{y}_{\widehat{\theta}(\mathbf{z})\,l}(x) - \widehat{\mu}_{y,l}) \right] \ . \tag{2.51}$$

The main difference with the OOD detection framework posed in [63] is that they measure the distance to a single distribution, and the goal is to determine if the target sample follows or not that distribution. In the MIA framework, we measure the distance to two distributions, the training set distribution and the outside-of-the-training-set distribution, and determine membership by verifying to which distribution the target sample is closest.

In fact, our MIA strategy uses the ratio between the distance to the outside-of-the-training-set distribution and the training set distribution. This requires to estimate two sets of parameters, one using samples from the training set, and the other using samples from outside the training set. For this we use subsets of the training set and test set, respectively. We denote the Mahalanobis distance score to the training set distribution as $M_{\text{in}\,l}(x)$ and the equivalent with respect to samples outside of the training set as $M_{\text{out}\,l}(x)$. With this notation we can write the *Mahalanobis membership score*,

$$\phi_l(x, \widehat{\theta}(\mathbf{z})) = \frac{M_{\text{out}\,l}(x)}{M_{\text{in}\,l}(x)} \ . \tag{2.52}$$

As for all the other MIAs strategies, the score is compared to a threshold in order to predict the membership of the target sample. Indeed, if the ratio is large, the candidate sample is closer in distribution to the training set, and therefore is predicted to be in the training set. Otherwise, the candidate sample is closer in distribution to samples outside the training set and is predicted to be outside the training set. The whole process of computing the Mahalanobis membership score is illustrated by Fig. 2.7.

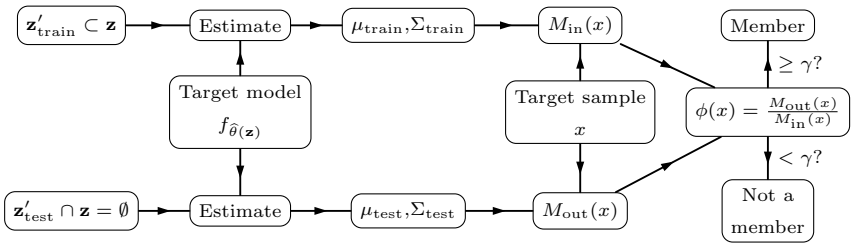


Figure 2.7: Schematic of Mahalnobis membership score computation. A subset $\mathbf{z}'_{\text{train}}$ of the training set of the target model is used to estimate the parameters of the training set distribution. Similarly, a subset $\mathbf{z}'_{\text{test}}$ of the test set of the target model is used to estimate the parameters of the outside-of-the-training-set distribution. With the estimated parameters, the Mahalanobis distance with respect to the training set M_{in} and the Mahalanobis distance with respect to the test set M_{out} are computed. The ratio of this quantities is taken as a score criteria to perform membership inference.

Algorithm 4 Mahalanobis Membership Inference Attack

Require: Target sample x, target model $f_{\hat{\theta}(\mathbf{z})}$, parameters of the training set distribution $\widehat{\mu}_{y,l}$, $\widehat{\Sigma}_l$, parameters of the test set distribution $\widehat{\mu}'_{y,l}$, $\widehat{\Sigma}'_l$, and threshold $\gamma \in \mathbb{R}$.

1. **for** $l \in \{1, \ldots, L\}$ **do**
2. $\quad M_{\text{in}\ l}(x) \leftarrow \max_y -(\hat{y}_{\hat{\theta}(\mathbf{z})\ l}(x) - \widehat{\mu}_{y,l})^\top \widehat{\Sigma}_l^{-1} (\hat{y}_{\hat{\theta}(\mathbf{z})\ l}(x) - \widehat{\mu}_{y,l})$
3. $\quad M_{\text{out}\ l}(x) \leftarrow \max_y -(\hat{y}_{\hat{\theta}(\mathbf{z})\ l}(x) - \widehat{\mu}'_{y,l})^\top \widehat{\Sigma}_l'^{-1} (\hat{y}_{\hat{\theta}(\mathbf{z})\ l}(x) - \widehat{\mu}'_{y,l})$
4. $\quad \phi_l(x, f_{\hat{\theta}(\mathbf{z})}) \leftarrow \frac{M_{\text{out}\ l}(x)}{M_{\text{in}\ l}(x)}$
5. **end for**
6. $\phi(x, \widehat{\theta}(\mathbf{z})) \leftarrow \sum_l w_l \phi_l(x, \widehat{\theta}(\mathbf{z}))$ \\Mahalanobis membership score
7. **return** $\mathbb{1}\{\phi(x, \widehat{\theta}(\mathbf{z})) \geq \gamma\}$ \\Threshold check

In order to combine the Mahalanobis membership scores for different layers, we compute their weighted sum,

$$\phi(x, \widehat{\theta}(\mathbf{z})) = \sum_{l=1}^{L} w_l \, \phi_l(x, f_{\hat{\theta}(\mathbf{z})}) \, . \tag{2.53}$$

The weights $\{w_l\}$ are learned by logistic regression with the goal of maximizing the prediction accuracy of the MIA. The logistic regression uses the same subsets of the training and test set that are used to estimate the parameters of the distribution in (2.49) and (2.50). Algorithm 4 summarizes this MIA strategy.

It is important to mention that the outputs of intermediate layers are reduced by averaging across the spatial dimensions. The computational resources necessary to store and to compute algebraic operations on the output of intermediate layers can grow exponentially for certain models and datasets. Therefore, to make these computations feasible for the intermediate layers we need to reduce their dimension. Inspired by [63], the idea is to reduce the intermediate outputs from $F \times H \times W$ to $F \times 1$, where F is the number of channels, and $H \times W$ is the spatial dimension.

In our experiments, we tried to reduce the spatial dimensions of the outputs of intermediate layers with different methods. Namely, by taking the pixel with the maximum value, by taking the average of all pixels along each channel, and by taking the pixel with the maximum variation across a subset of samples. We found that taking the average per channel was the most efficient in terms of the performance of the attacker, thus we chose to apply this method.

2.3.5 Information Geometry Approach to OOD Detection

The following method is similar to the Mahalanobis membership score in the sense that it measures how much the target sample follows a given distribution. The attacker will estimate the empirical distribution of the outputs of the target model over samples from training set and over samples from outside the training set. The attacker will measure the distance to both of these distributions and compute their ratio. The ratio is used as a score to predict membership. Figure 2.7 could also be used to describe this technique, with the only difference being the form of the distributions and the parameters that have to be learned from the additional samples given to the attacker. The following describes the assumptions of the model and defines the distributions and their parameters to be estimated.

Let us consider a statistical manifold, i.e. a parameterized family of probability distributions that is obtained by fixing the parameters of a neural network model and changing its input features. Through the Fisher-Rao distance (see [5, 80] and references therein), it is possible to measure the dissimilarity between two probability models within this family by calculating the geodesic distance between two points on the learned manifold. This measure has been successfully applied to the OOD detection problem in [38] and to the adversarial robustness probelm in [79].

We apply two different formulations of the Fisher-Rao (FR) distance measure. For the logits layer, that is $l = L$, we use expression (1.1), and let [79]:

$$d_{\mathrm{FR-Logits}}(x, x'; \widehat{\theta}(\mathbf{z}), \delta) = 2 \arccos \left(\sum_{y' \in \mathcal{Y}} \sqrt{f_{\widehat{\theta}(\mathbf{z}),\delta}(y'; x) f_{\widehat{\theta}(\mathbf{z}),\delta}(y'; x')} \right), \tag{2.54}$$

for two inputs $x, x' \in \mathcal{X}$. Using this expression, we compute the Fisher-Rao score,

$$\mathrm{FR}_L(x) = \sum_{y \in \mathcal{Y}} d_{\mathrm{FR-Logits}}(x, \mu_y; \widehat{\theta}(\mathbf{z}), \delta), \tag{2.55}$$

where μ_y is the empirical centroid for the logits of each class $y \in \mathcal{Y}$ according to the Fisher-Rao distance (2.54). The empirical centroids were estimated according to the following expression,

$$\mu_y = \underset{\mu \in \mathbb{R}^{|\mathcal{Y}|}}{\arg\min} \frac{1}{N_y} \sum_{i: y_i = y} d_{\mathrm{FR-Logits}}(x_i, \mu; \widehat{\theta}(\mathbf{z}), \delta) \ , \tag{2.56}$$

where N_y is the amount of considered samples with label y. We optimize this expression where the parameter to be tuned is μ in the logits space. Before the optimization procedure for class $y \in \mathcal{Y}$ begins, μ is initialized as the y-th standard basis vector of $\mathbb{R}^{|\mathcal{Y}|}$. For each class, we minimize the expression in equation (2.56), selecting ADAM as

gradient descent optimizer, using 1000 epochs with a fixed learning rate equal to 10^{-2} for each model and dataset involved in our empirical evaluation. The parameters μ_y are estimated separately over a set of samples belonging to the training set and over a set of samples that do not belong to the training set so that we can compute the distance in distribution to samples inside and outside the training set, respectively.

For the intermediate layers (latent code), i.e. $l \in \{1, \ldots, L-1\}$, since their outputs are not directly interpretable as distributions, it is possible to define a set of class-conditional Gaussian distributions with diagonal covariance matrix Σ_l and class-conditional mean $\widehat{\mu}_{y,l}$. The mean is computed as in (2.49), while the diagonal elements of the covariance matrix Σ_l are,

$$(\Sigma_l)_{j,j} = \frac{1}{N} \sum_{y \in \mathcal{Y}} \sum_{i:y_i=y} \left(\left(\hat{y}_{\widehat{\theta}(\mathbf{z})\, l}(x_i) \right)_j - (\widehat{\mu}_{y,l})_j \right)^2 , \tag{2.57}$$

where $j \in \{1, \ldots, J\}$, J denotes the size of the l-th intermediate layer and, with N_y being the amount of samples in class y, $N = \sum_y N_y$. The Fisher-Rao distance ρ_{FR} between two arbitrary univariate Gaussian pdfs $\mathcal{N}_1$ and $\mathcal{N}_2$ with parameters (μ_1, σ_1) and (μ_2, σ_2) respectively, is given by,

$$\rho_{\text{FR}}\left((\mu_1, \sigma_1), (\mu_2, \sigma_2)\right) = \sqrt{2} \log \frac{\left\| \left(\frac{\mu_1}{\sqrt{2}}, \sigma_1\right) - \left(\frac{\mu_2}{\sqrt{2}}, -\sigma_2\right) \right\| + \left\| \left(\frac{\mu_1}{\sqrt{2}}, \sigma_1\right) - \left(\frac{\mu_2}{\sqrt{2}}, \sigma_2\right) \right\|}{\left\| \left(\frac{\mu_1}{\sqrt{2}}, \sigma_1\right) - \left(\frac{\mu_2}{\sqrt{2}}, -\sigma_2\right) \right\| - \left\| \left(\frac{\mu_1}{\sqrt{2}}, \sigma_1\right) - \left(\frac{\mu_2}{\sqrt{2}}, \sigma_2\right) \right\|} , \tag{2.58}$$

where $\| \cdot \|$ denotes the 2-norm in $\mathbb{R}^2$. The Fisher-Rao distance $d_{\text{FR}-\text{Gauss}}$ between two multivariate Gaussian pdfs with diagonal standard deviation matrix is derived from the univariate case and is given by,

$$d_{\text{FR}-\text{Gauss}}\left((\mu, \Sigma), (\mu', \Sigma')\right) = \sqrt{\sum_{j=1}^{J} \rho_{\text{FR}}\left((\mu_j, \Sigma_{j,j}), (\mu'_j, \Sigma'_{j,j})\right)^2} , \tag{2.59}$$

where $\mu, \mu' \in \mathbb{R}^J$ and $\Sigma, \Sigma' \in \mathbb{R}^{J \times J}$ are diagonal, positive definite matrices.

Finally, a score based on $d_{\text{FR}-\text{Gauss}}$ can be derived by applying the notion of Fisher-Rao distance between the candidate sample x and the closest class-conditional diagonal Gaussian distribution. We obtain,

$$\text{FR}_l(x) = \min_{y \in \mathcal{Y}} d_{\text{FR}-\text{Gauss}}\left((x, \Sigma_l), (\widehat{\mu}_{y,l}, \Sigma_l)\right) , \tag{2.60}$$

where $\widehat{\mu}_{y,l}$ and Σ_l are given in (2.49) and (2.57), respectively. Similarly to the concept reported in Section 2.3.4 (cf. (2.52)), the quantities Σ_l and $\widehat{\mu}_{y,l}$ can be empirically estimated using training data (resulting in $\text{FR}_{\text{in}\, l}(x)$), or test data (resulting in $\text{FR}_{\text{out}\, l}(x)$). The same reasoning can be applied to (2.55), allowing us to reduce these scores to the ratio $\frac{\text{FR}^l_{\text{out}}(x)}{\text{FR}^l_{\text{in}}(x)}$. Once again, taking a cue from Section 2.3.4, these ratio scores are aggregated,

$$\phi(x, \widehat{\theta}(\mathbf{z})) = \sum_{l=1}^{L} w_l \, \frac{\text{FR}_{\text{out}\, l}(x)}{\text{FR}_{\text{in}\, l}(x)} , \tag{2.61}$$

using logistic regression to compute the aggregation weights as in (2.53).

2.4 Attribute Inference Attacks

The idea of this section is to illustrate how AIAs can be formalized within our framework and how membership inference strategies can translate to the attribute inference problem. Following Definition 5, the attacker has access to side information s, which includes the *non-sensitive* attributes v of the target sample, and their goal is to determine the *sensitive* attribute t. We will take the same approach as we did for MIAs and determine the target (sensitive) attribute by choosing the most likely value according to some criteria. Thus, each criteria will determine a AIAs strategy. The *Gradient* and *Loss* strategies are inspired by similar strategies from the membership inference literature [74, 90].

Softmax Response: The intuition behind this attack is that a model is more confident on samples that were part of its training. Therefore, by choosing the correct value t, the model will maximize its output for the predicted label. Note that this criteria does not care about the model making the right prediction. The side information given to the attacker are the non-sensitive attributes, $s = v$. This strategy chooses the sensitive attribute that outputs the highest score, i.e.,

$$\varphi(v, \widehat{\theta}(\mathbf{z})) = \arg\max_{t \in \mathcal{T}} \left[\max_{y' \in \mathcal{Y}} f_{\widehat{\theta}(\mathbf{z})}\left(y'; (v, t)\right) \right] . \tag{2.62}$$

Accuracy: In contrast to the previous one, this strategy chooses the sensitive attribute that produces the *right* prediction with the highest score. This is the closest to the strategy proposed by [32]. The side information given to the attacker are the non-sensitive attributes and the label, $s = (v, y)$. Define set $\widehat{X}_{y\widehat{\theta}(\mathbf{z})} \triangleq \left\{ x \in \mathcal{X} : \arg\max_{y' \in \mathcal{Y}} \left(f_{\widehat{\theta}(\mathbf{z})}(y'; x) \right) = y \right\}$, then,

$$\varphi(v, y, \widehat{\theta}(\mathbf{z})) = \arg\max_{t \in \mathcal{T} :\, x \in \widehat{X}_{y\widehat{\theta}(\mathbf{z})}} \left[\max_{y' \in |\mathcal{Y}|} f_{\widehat{\theta}(\mathbf{z})}\left(y'; (v, t)\right) \right] . \tag{2.63}$$

Loss: This attack strategy assumes that the target sample was part of the training set of the target model. During training, the target model minimizes the loss over samples in its training set. Thus for a 'real' sample, i.e. a sample with the correct t value, the loss should be at a minimum. The side information given to the attacker is the non-sensitive attributes and the label: $s = (v, y)$. This strategy chooses the sensitive attribute that minimizes the loss, i.e.,

$$\varphi(v, y, \widehat{\theta}(\mathbf{z})) = \arg\min_{t \in \mathcal{T}} \ell \left(\hat{y}_{\widehat{\theta}(\mathbf{z})}\left((v, t)\right), y \right) . \tag{2.64}$$

Gradient: Similar to the previous strategy, the assumption here is that the target sample was part of the training set of the target model. Near a minimum of the loss function, the norm of its gradient with respect to its model parameters should approach 0; the attacker exploits this knowledge for the present attack strategy. While the previous attacks only make use of the output of the model or the value of its loss, the present attack makes explicit use of its parameters, thus being considered a white-box attack. The side information given to the attacker are the non-sensitive attributes and the label, $s = (v, y)$. This strategy chooses the sensitive attribute that minimizes the

gradient norm, i.e.,

$$\varphi(v, y, \widehat{\theta}(\mathbf{z})) = \arg\min_{t \in \mathcal{T}} \left\| \nabla_{\widehat{\theta}(\mathbf{z})} \ell \left(\hat{y}_{\widehat{\theta}(\mathbf{z})}((v, t)), y \right) \right\|_2^2 . \tag{2.65}$$

In Section 3.4 we test these strategies against a model trained on the PenDigits dataset. In this experiment, the model is trained to identity digits using, among other attributes, the identity of the writer. After training, the identities of the writers are removed from the dataset and the attacker is asked to recover those identities from the incomplete samples plus the target model. As shown in our experiments, attribute inference strategies inspired from membership inference can be effective at recovering sensitive information from a dataset where this information has been removed.

Chapter 3

Experiments

The aim of this chapter is to illustrate and provide empirical evidence for the ideas and results presented in the previous chapter. The first part of the chapter is dedicated to illustrating our theoretical results. We start by considering a simple scenario, linear regression with Gaussian data, which allows to estimate the performance of the Bayesian attacker and its lower bound connecting to the generalization gap of the target model. Following these experiments, we apply our theoretical results in a more complex scenario; namely, DNNs for image classification. Although this scenario does not allow us to compute the success rate of the Bayesian attacker, we can still compute its lower bound which connects to the generalization gap of the target model.

The second part of this chapter focuses on finding the most effective MIA strategy in practice. For this purpose, we evaluate a wide array of MIA strategies by launching attacks against pre-trained state-of-the-art models for image classification and comparing their performance.

Starting off, Section 3.1 explains the datasets and target models used in the experiments of the following sections. The theoretical results presented in section Section 2.1 are illustrated in Section 3.2, while the membership inference strategies presented throughout Sections 2.2 and 2.3 are evaluated and compared in Section 3.3. Finally, the attribute inference strategies presented in Section 2.4 are evaluated in Section 3.4.

3.1 Datasets and Target Models

In this section we describe the datasets and target models used in our experiments.

3.1.1 Datasets

MNIST. Published in [62]. The dataset contains 70k 28×28 pixels, grayscale images of hand-written digits split amongst 10 different classes. The classes correspond to the digits from 0 to 9. In standard libraries, such as PyTorch [76], this dataset is divided into a training set containing 60k images and a test set containing 10k images. The standard training set provided by PyTorch is used to train the target models we consider, the rest is used as outside-the-training-set data.
Fashion MNIST. This dataset was first published in [105]. Similar to MNIST, it contains 70k 28×28 pixels, grayscale images split amongst 10 different classes. In this case the classes correspond to different types of clothing items, such as 'shoe' or 'bag'. The splitting of the dataset in standard libraries is the same as for MNIST, with a training set containing 60k images and a test set containing 10k images. The standard training set provided by PyTorch is used to train the target models we consider, the rest is used as outside-the-training-set data.
CIFAR10 and CIFAR100. These datasets are a standard benchmark for image recognition tasks [59]. They contain 60k 32×32 pixels, color (RGB) images split amongst 10, 100 distinct classes, respectively. The classes correspond to different animals or objects, e.g. 'bird' or 'truck'. In standard libraries these datasets are usually divided into a training set containing $50k$ images and a test set containing the remaining $10k$ images. The standard training set provided by PyTorch is used to train the target models we consider, the rest is used as outside-the-training-set data.
PenDigits The dataset [27] was taken by asking participants to write digits from 0 to 9 on a tablet. The original data contains variable-length time series that correspond to the position of the pen on the tablet over time. We pre-process the data to make the length of the time series uniform (length 32). Since the capture rate of the tablet is uniform, we can infer the time that it took to write a digit by the length of the original series. We keep this information, along with the number of strokes that were used to write the digit and the identity of the writer. Thus, each sample from this dataset contains a multivariate time series with the coordinates of the pen, a float indicating the total time of recording, an integer number indicating the number of strokes and a one-hot-encoding of the identity of the writer. For attribute inference experiments, the whole dataset of 11990 samples is used as a pool of training samples, from which training sets can be selected. For membership inference, the dataset is split into a pool of 8k training samples and a pool of 3990 test samples. In this case, the training pool is used to select a training set for the target model and the test pool is used as outside-the-training-set data.

3.1.2 Target Models

Custom DNNs. The target model for the experiments of Section 3.2.2 is a Deep Neural Network with 4 convolutional layers and 3 fully connected layers. For CIFAR10 the model has a total of 439722 parameters, while for MNIST and Fashion MNIST it has only 376714. The loss used for training is the mean squared error (MSE) between

the soft probabilities and the one-hot-encoded labels. The model is trained for up to 150 epochs using the Adam optimizer [56] with learning rate $5 \cdot 10^{-3}$. The batch size used for training the models is 200 (this represents the whole training set when the total number of samples in the training set is equal to 200). An early stop criteria compares the current loss over the training set to the total loss after the previous epoch, and stops training if the difference is below 10^{-3}. The number of epochs of training can change drastically depending on the size of the training set.

State-of-the-art models for image recognition. We consider popular models for image recognition, pre-trained and publicly available[1]. Namely, the models considered are AlexNet [60], ResNet [42], ResNext [106] and DenseNet [46], trained for image classification on CIFAR100. These are the same pretrained models considered in previous works that evaluate the performance of MIA strategies [74, 85].

DNN for PenDigits classification. The model is a DNN trained to classify handwritten digits. The input to the network consist of two time series (one for each coordinate) indicating the position of the pen over time, an integer indicating the number of strokes, a float between 0 and 1 indicating the length of the original sequences and a one-hot-encoding of the identity of the writer. The latter is considered as the sensitive attribute, while the other inputs are considered non-sensitive.

The target model possesses 4 fully-connected layers and a total of 4650 parameters. The loss for training is the MSE between the soft probabilities and the one-hot-encoded labels; this is a bounded loss function, allowing us to use Theorem 2 to lower bound the success rate of the Bayesian attacker. The model is trained with Adam optimizer (learning rate $5 \cdot 10^{-3}$) for up to 2500 epochs. An early stop criteria compares the current loss over the training set to the total loss after the previous epoch, and stops training if the difference is below 10^{-4}.

[1] Model implementations, pre-trained weights and code to train the models available at `https://github.com/bearpaw/pytorch-classification`

3.2 Empirical Assessment of the Bounds

The aim of this section is to illustrate the theoretical results from Section 2.1 and how they can be used to assess the privacy of a ML model in practice. We propose a setup where the target model is a linear regression algorithm trained on synthetic Gaussian data. The simplicity of this setup allows to compute several important quantities, such as the generalization gap and the error incurred by the Bayesian attacker, in closed form. This in turn allows to apply Theorem 1 directly to assess the privacy of the model. It is important to remember that privacy means nothing without taking into account the utility of the trained model. A model that is not trained will present no privacy risks, but it will also provide no utility. In view of this remark, we measure the accuracy and generalization gap of the trained model along with its privacy, to have a complete picture of the trained model.

To see how these results translate to a more realistic scenario, we perform similar experiments on a DNN trained for image classification. The complexity of the target model forces us to measure the generalization gap in an empirical way, rather than being able to compute it in closed form. It also results impossible to compute the error incurred by the Bayesian attacker in closed form in this setup. To circumvent this difficulty, we implement state-of-the-art membership inference strategies and use the success rate achieved by these strategies as a surrogate for the success rate of the Bayesian attacker. Since the loss function for this model is bounded, we are able to compute the lower bound on the success rate of the Bayesian attacker given by Theorem 2 and compare it to the success rate of the implemented MIA strategies.

3.2.1 Linear Regression on (Synthetic) Gaussian Data

The following example allows us to illustrate how the theoretical results from Section 2.1 might be used to assess the privacy guarantees of a specific model. We implement the Bayesian attacker from Theorem 1 and estimate its success probability to monitor the privacy leakage of the model. Empirically, we observe that both the success rate of the attacker and the generalization gap of the model are a function of the number of training samples; thus, we study the variation of both of these quantities as we increase the number of training samples. Second, since the loss is tail-bounded exponentially, we use Theorem 4 to derive lower bounds on the success probability of the attacker. Lastly, we utilize (2.20) from Theorem 5 to upper bound the success probability of the Bayesian attacker.

For $i \in [n]$, let x_i be a fixed vector on $\mathbb{R}^d$ and for a fixed vector $\beta \in \mathbb{R}^d$, let $Y_i = \beta^T x_i + W_i$ with $\mathbb{E}[W_i] = 0$ and $\mathbb{E}[W_i^2] = \sigma^2 < \infty$ for $i \in [n]$. The training set is $\mathbf{z} = \{y_1, \ldots, y_n\}$, a realization of Y_i for each $i \in [n]$. The function space $\mathcal{F}$ consists of linear regression functions $f_\theta(x_i) = \theta^T x_i$ for $\theta \in \mathbb{R}^d$ and the deterministic algorithm $\mathcal{A}$ minimizes squared error on the training set and thus yields[2] $\widehat{\theta}(\mathbf{y}) = (\mathbf{x}\mathbf{x}^T)^{-1}\mathbf{x}\mathbf{y}^T$ and the associated decision function $f_{\widehat{\theta}(\mathbf{y})}(x_i) = \mathbf{y}\mathbf{x}^T(\mathbf{x}\mathbf{x}^T)^{-1}x_i$. Using squared error loss, $\ell(y, y') = (y - y')^2$, we obtain the generalization gap,

$$\mathcal{E}_{\mathrm{G}}(\mathcal{A}) = \frac{2d}{n}\sigma^2 \,, \tag{3.1}$$

[2]Let $\mathbf{x}$ be the $[d \times n]$ matrix $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. Similarly, $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ and $\mathbf{W} = (W_1, W_2, \ldots, W_n)$ are $[1 \times n]$ vectors.

A derivation of this formula is presented in E. Assuming the noise W to be Gaussian, the scalar response $\mathbf{Y} = \beta^T \mathbf{x} + \mathbf{W}$ then also follows a Gaussian distribution, with $\mathbf{W}$ a row vector of i.i.d. components. Similarly, the model parameters $\widehat{\theta}(\mathbf{Y})$ are normally distributed. Now choose a test sample $S_J = T(Y_J) + (1-T)(Y'_J)$, where J is an index in $[n]$, Y_J is the $J-$th component of the (random) training set and Y'_J is drawn independently of the training set. Assuming a Bernoulli $1/2$ prior on the hypothesis T, the success probability of the Bayesian attacker $\varphi^\star$ is given by

$$\mathcal{P}_{\mathrm{Suc}}(\varphi^\star) = 1 - \frac{1}{2}\left[\epsilon_0\big(\widehat{\mathcal{T}}(1)\big) + \epsilon_1\big(\widehat{\mathcal{T}}(1)^c\big)\right] , \tag{3.2}$$

with the Type-I and Type-II errors defined by (2.6), and the optimal decision region $\widehat{\mathcal{T}}(1)$ defined by (1.7). With posteriors defined by,

$$p_{S_J J \widehat{\theta}|T}(s, j, \theta|0) = \frac{1}{n} Q(\theta) p_{Y_j}(s) , \tag{3.3}$$

$$p_{S_J J \widehat{\theta}|T}(s, j, \theta|1) = \frac{1}{n} Q_j(\theta|s) p_{Y_j}(s) . \tag{3.4}$$

The index j indicates the feature vector x_j from which the test sample s is generated. $Q(\theta)$ is the distribution of the model parameters conditioned to $T = 0$. It is independent of the test sample s and of the index j. $Q_j(\theta|s)$ is the distribution of the model parameters conditioned to $T = 1$. Since, under this hypothesis, the attacker assumes s is one of the samples in the training set, this conditional distribution depends on the test sample s and its corresponding index j. The distribution of the test sample p_{Y_j} is defined by $p_{Y_j}(\,\cdot\,) \triangleq \mathcal{N}(\,\cdot\,; \beta^T x_j, \sigma^2)$. $Q(\,\cdot\,)$ and $Q_j(\,\cdot\,|s)$ are defined by $Q(\,\cdot\,) \triangleq \mathcal{N}(\,\cdot\,; \beta, \sigma^2 \overline{x}^{-1})$ and $Q_j(\,\cdot\,|s) \triangleq \mathcal{N}\big(\,\cdot\,; \beta + \overline{x}^{-1} x_j (s - x_j^T \beta), \sigma^2 \overline{x}^{-1} (\mathbb{I}^{d\times d} - x_j x_j^T \overline{x}^{-1})\big)$, respectively, where $\overline{x} \triangleq \mathbf{x}\mathbf{x}^T$. These distributions are derived in E.

The success probability of the Bayesian attack strategy in Theorem 1 is given by (3.2). In our experiments we perform a Monte Carlo estimation of the integrals in (2.6), by randomly drawing T, s and θ. The posterior distributions can be computed in closed form with the above definitions. Since the loss is exponentially tail-bounded, we can apply Theorem 4 to obtain the lower bound

$$\mathcal{P}_{\mathrm{Suc}}(\varphi^\star) \geq \frac{1}{2} + \frac{d}{2n} \frac{\sigma^2}{R_{\max}} - C(R_{\max}, \sigma), \tag{3.5}$$

where we used (3.1) and $C(R_{\max}, \sigma)$ is defined in expression (2.17). $R_{\max}$ can be chosen to maximize the upper bound in this expression. In our experiments, we choose the optimal $R_{\max}$ using the golden section search algorithm. Furthermore, from (2.20) we have,

$$I\big(S_J; \widehat{\theta}(\mathbf{Y})\big|T\big) \geq d_{\mathrm{KL}}\Big(\mathcal{P}_{\mathrm{Suc}}(\varphi) \,\Big\|\, \max_{t\in\mathcal{T}} p_T(t)\Big) . \tag{3.6}$$

Note that $I\big(S_J; \widehat{\theta}(\mathbf{Y})\big|T\big) \geq I\big(T; \widehat{\theta}(\mathbf{Y})\big|S_J\big)$. The mutual information between the testing sample and the model parameters given the sensitive attribute, $I\big(S_J; \widehat{\theta}(\mathbf{Y})\big|T\big)$, can be explicitly computed in this setup; the details of this computation are relegated to E. Fixing the prior on the hypothesis T to a Bernoulli $1/2$, we can utilize (3.6) to find an upper bound on the success probability of the Bayesian attacker. This is done by

Algorithm 5 Estimate success rate of the attacker

Input: feature vectors $\mathbf{x}$, training set size n
Draw t uniform in $\{0, 1\}$
Draw j uniform in $[n]$
$\mathbf{y} \longleftarrow \beta^T \mathbf{x} + \mathbf{W}$
if t **then**
    $s \longleftarrow y_j$
else
    $s \longleftarrow \beta^T x_j + W$
end if
$\theta \longleftarrow (\mathbf{x}\mathbf{x}^T)^{-1}\mathbf{x}\mathbf{y}^T$
return $p_{S_J \hat{J\theta}|T}(s, j, \theta|1) > p_{S_J \hat{J\theta}|T}(s, j, \theta|0)$ **XNOR** t

searching for the success rate $\mathcal{P}_{\text{Suc}}(\varphi)$ that makes the l.h.s. of (3.6) equal to its r.h.s. Namely, the golden section search algorithm is used to minimize the square distance between the mutual information and the KL-divergence with respect to $\mathcal{P}_{\text{Suc}}(\varphi)$.

Algorithm 5 details our simulations to estimate the success rate of the Bayesian attacker. It returns '1' when the attacker successfully predicts whether the test sample s was part of the training set or not, and '0' otherwise. In our experiments we vary n to study how the generalization gap and success rate of the attacker evolve as a function of the number of training samples. The dimension of the feature space is fixed to $d = 20$. For each value of n, we fix $\mathbf{x}$ and we repeat (10k times) Algorithm 5 to estimate the success rate of the attacker. The feature vectors $\mathbf{x}$ are generated i.i.d. and then fixed for each value of n. Additionally, for n, we compute the generalization gap (3.1), which is used to compute the lower bound (3.5). We also compute the Mutual Information in the l.h.s. of (3.6), which is used to compute the upper bound on the success probability of the attacker.

Figure 3.1 (**Top**) shows the success rate (SR) of the Bayesian attacker as a function of the number of samples in the training set n. Along with it is the lower bound (LB) provided by Theorem 4 and the upper bound (UP) provided by equation (3.6). The lower bound predicts the behavior of the SR as a function of the generalization gap. For large n (small generalization gap), the success rate and its lower bound approach 0.5, the success rate of an attacker that only uses knowledge on the prior of T. While the lower bound seems loose in this setting, it is worth noting that we compare with the best possible strategy. Nonetheless, this example shows that the bounds are not vacuous and they may serve as a framework for understanding the connection between information leakage and generalization in ML. On the other hand, the upper bound provides a strong privacy guarantee. In cases where the success rate of the Bayesian attacker cannot be explicitly computed, its upper bound is the best privacy guarantee that can be provided. Additionally, Fig. 3.1 (**Bottom**) shows the mutual information (l.h.s. of (3.6)) that is used to compute the upper bound.

3.2.2 Examples on DNNs

We train DNNs on various datasets to study the interplay between generalization gap and the success rate of three different black-box MIA strategies. We compare the success rate of the different attack strategies to the lower bound provided by Theorem 2 to

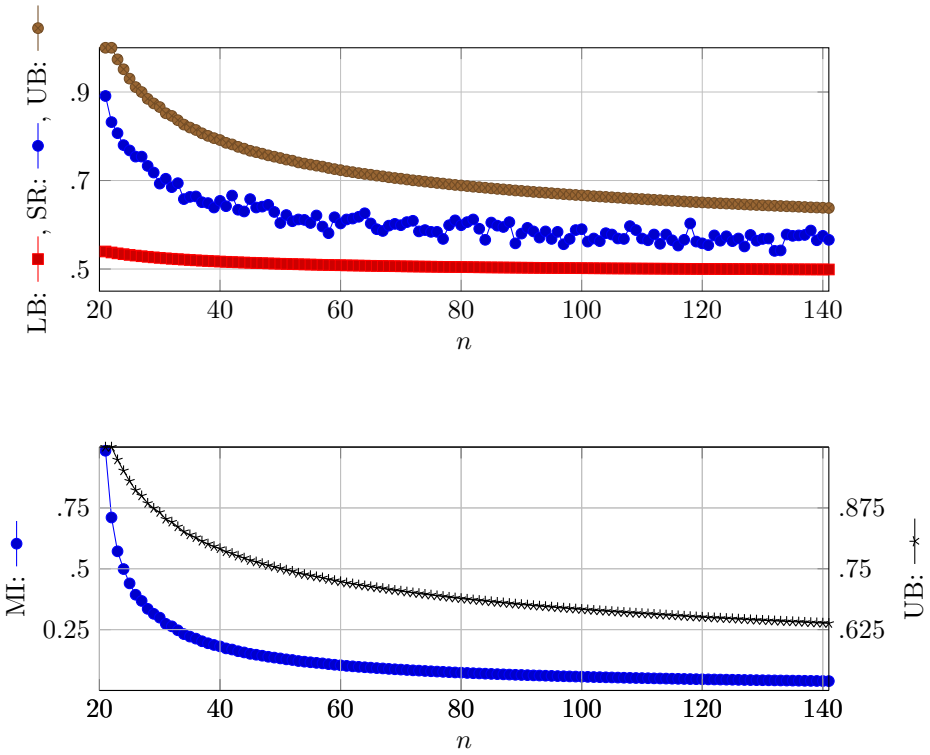


Figure 3.1: Dependence of success rate of the Bayesian attacker, generalization gap, and mutual information on the number of training samples n, using Gaussian data. **Top:** Success Rate (SR), Lower Bound (LB), and Upper Bound (UB). **Bottom:** Mutual Information (MI), Upper Bound (UB; axis labels on the right).

	Softmax Response	Loss	Modified Entropy [93]	[109]	[90]
Attack complexity	One query	One query	One query	One query	Thousands of queries and train shadow models
Required Knowledge	Soft Probabilities	Loss value	Soft Probabilities	Training Loss	Additional Samples
PPV MNIST	0.444 ± 0.000	0.446 ± 0.000	0.444 ± 0.000	0.505	0.517
PPV CIFAR-10	0.446 ± 0.001	0.451 ± 0.001	0.449 ± 0.001	0.694	0.72
PPV Fashion-MNIST	0.445 ± 0.000	0.447 ± 0.001	0.446 ± 0.001	–	–
Recall	> 0.99	> 0.99	> 0.99	> 0.99	> 0.99

Table 3.1: Comparison of the basic MIA strategies to previous black-box MIAs from the literature. Precision (PPV; Positive Predictive Value) and recall are reported for CIFAR10, MNIST and Fashion MNIST.

Algorithm 6 Estimate success rate of the Attacker

1: **Require:** Target model $f_{\widehat{\theta}(\mathbf{z})}$, score criteria ϕ, threshold γ, training set, test set.
2: Draw t uniform in $\{0, 1\}$
3: **if** t **then**
4: Draw s uniform from the training set.
5: **else**
6: Draw s uniform from the test set.
7: **end if**
8: **return** $\phi(s, \widehat{\theta}(\mathbf{z})) > h$ **XNOR** t

assess the quality of the bound. Our datasets for these experiments are MNIST, Fashion MNIST and CIFAR10. Details about datasets and the target model are provided in Section 3.1.

The loss function used for training and for computing the generalization gap is the MSE loss between the one-hot encoded labels and the soft probabilities output by the network. Note that this loss function is bounded by 2. While Cross-Entropy is a more common choice for loss function, it is not bounded. On the other hand MSE has a negligible effect on performance and allows us to apply Theorem 2 to lower bound the success probability of the Bayesian attacker. However, in this setup it results impossible to estimate the success probability of the Bayesian attacker, due to the high number of model parameters. To circumvent this limitation and assess the quality of the bound provided by Theorem 2, we implement the Softmax response, Loss, and Modified Entropy attack strategies for membership inference described in Section 2.2, and compare their success rate to the bound.

To compute the success rate of the given attack strategies we draw the target concept t uniformly in $\{0, 1\}$ and then draw the target sample from the training set or the test set of the target model accordingly. This part of the procedure is detailed in Algorithm 6. Note that Algorithm 6 outputs 1 if the attacker infers membership correctly and 0 otherwise. Then, the success rate of the attacker is computed by simply counting the number of times it succeeds over the total number of trials, which is set to 10k. Experimentally, we found that a threshold of $h = 0.8$ works best across different values of n.

The number of samples in the training set, n, varies in our experiments. For fixed n, that many samples are uniformly randomly picked from a pool of training samples. A test set is also fixed to measure the accuracy of the trained model and to empirically compute the generalization gap. In the case of MNIST and Fashion MNIST, the training

set is picked from a pool of 60k samples. A separate pool of 10k samples is fixed as the test set. For CIFAR10, the pool of training samples is of size 50k, and the pool of test samples is of size 10k.

We vary the size n of the training set and observe how this affects the success rate of attacks, the generalization gap and consequently the lower bound derived from Theorem 2. For a fixed value of n, the number of samples in the training set, the success rate of the Softmax response (SR), Loss (LS), and Modified Entropy (ME) attacks along with the lower bound (LB) provided by Theorem 2 and the accuracy on the test set (Acc) are obtained empirically in 100 runs. The results over different realizations of the target model are averaged to produce a single value for each n.

The results for CIFAR10, MNIST and Fashion MNIST are reported in Figure 3.2. The lower bound predicts the behavior of the success rate of the MIAs as a function of the generalization gap; both approach 0.5 (the success rate of a random guess) as the generalization gap vanishes. Note that it is possible for the success rate of the Softmax Response attack to go below the lower bound of the Bayesian attack. For some large n values of MNIST the average success rate of the attacker goes below 0.5. In this region the attacker cannot do better than a random guess and sometimes its success rate goes below 0.5, which implies the model can be more confident in samples outside the training set. This is an artifact of the random sampling of the training set and the training of the model.

Table 3.1 compares the strategies considered in our experiments to other previous MIA strategies found in the literature. The three strategies here considered do not require access to the model parameters or additional samples, and they only need to query the model once, while the other strategies [109, 90] require extra information or significantly more computing power. The attack is performed against target models with a training set of 8000 samples, to match the setup used in [109, 90]; however, the architectures of the target models, as well as the (random) selection of training samples differ in all three setups.

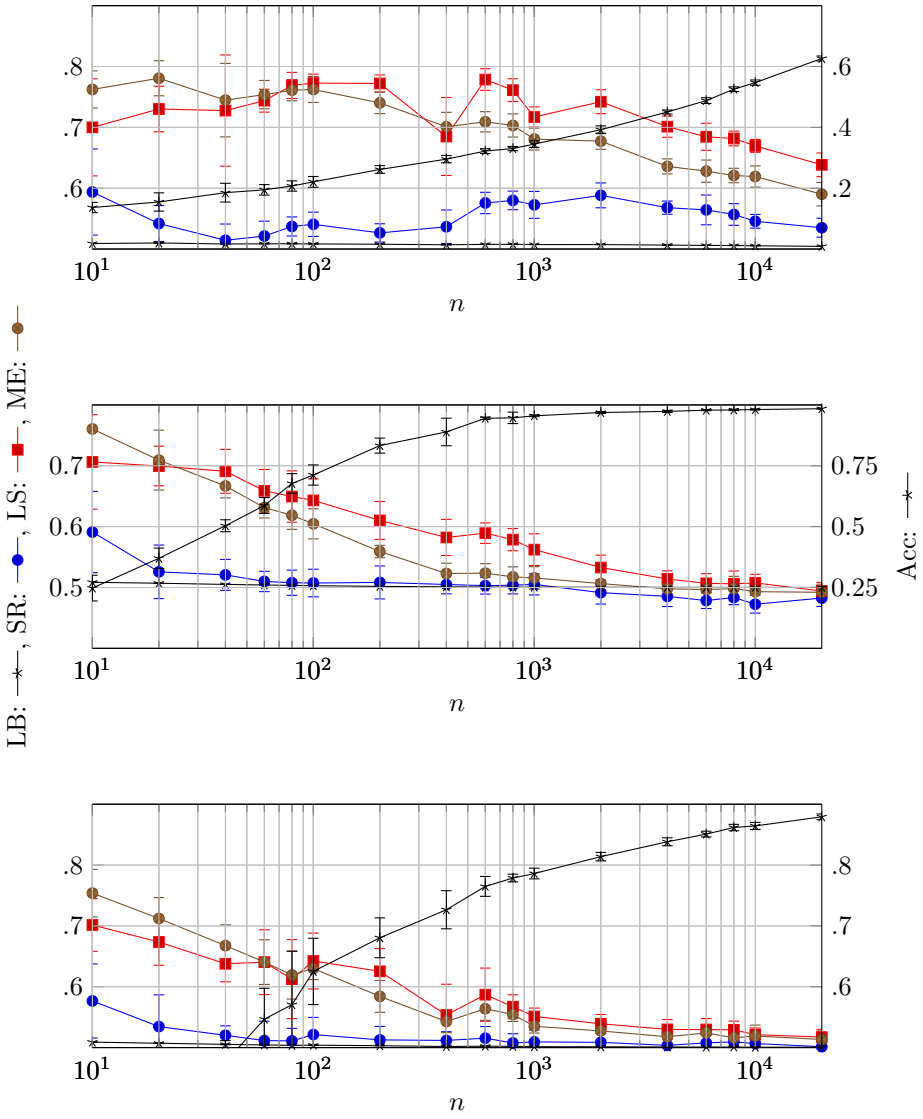


Figure 3.2: Success rate of the Softmax Response (SR), Loss attack (LS), Modified Entropy attack (ME), lower bound (LB) and accuracy (Acc; axis labels on the right) depend on the number of training samples n. **Top:** CIFAR10; **Middle:** MNIST; **Bottom:** Fashion MNIST.

Attack Strategy	AlexNet (10)	DenseNet (10)	ResNet (10)	AlexNet (100)	DenseNet (100)	ResNet (100)	ResNext (100)
Baseline	57.32 ± 0.17	51.64 ± 0.09	52.78 ± 0.09	74.31 ± 0.31	58.60 ± 0.13	60.80 ± 0.26	58.75 ± 0.19
Softmax Resp.	54.94 ± 0.38	56.93 ± 0.40	53.35 ± 0.38	68.31 ± 0.31	74.56 ± 0.47	60.42 ± 0.34	79.08 ± 0.23
Mod. Entropy	57.96 ± 0.35	56.61 ± 0.12	54.78 ± 0.30	77.56 ± 0.21	76.21 ± 0.29	63.03 ± 0.39	79.64 ± 0.20
Loss	57.89 ± 0.35	57.04 ± 0.39	53.63 ± 0.35	77.40 ± 0.19	74.82 ± 0.47	62.85 ± 0.39	79.23 ± 0.23
Grad. Norm x	57.98 ± 0.35	57.05 ± 0.42	53.65 ± 0.35	76.25 ± 0.20	74.82 ± 0.46	62.86 ± 0.37	79.12 ± 0.24
Grad. Norm w	57.99 ± 0.34	57.11 ± 0.39	53.64 ± 0.35	77.51 ± 0.22	74.95 ± 0.46	63.08 ± 0.39	79.32 ± 0.22
Grad. x	58.69 ± 0.39	55.55 ± 0.48	54.88 ± 0.38	76.61 ± 0.29	73.65 ± 0.56	63.78 ± 0.58	76.34 ± 1.25
Grad. w	58.72 ± 0.41	55.52 ± 0.30	56.37 ± 0.43	78.99 ± 0.29	73.43 ± 0.76	63.60 ± 0.38	76.48 ± 0.28
Int. Outputs	50.91 ± 0.50	53.38 ± 0.55	51.08 ± 0.56	50.41 ± 0.40	52.48 ± 1.11	50.71 ± 0.45	78.70 ± 0.42
White-Box	51.64 ± 1.02	50.32 ± 0.16	50.65 ± 0.43	74.47 ± 5.91	53.56 ± 2.49	52.65 ± 0.93	52.94 ± 1.65
Ensemble Attacker	59.57 ± 0.86	56.50 ± 1.88	53.95 ± 0.75	78.94 ± 0.95	74.91 ± 0.46	63.86 ± 1.12	79.26 ± 0.22
Adv. Distance l_2	57.74 ± 0.22	51.61 ± 0.36	53.01 ± 0.37	73.56 ± 0.31	58.13 ± 0.36	60.09 ± 0.34	58.30 ± 0.39
Adv. Distance l_∞	57.51 ± 0.19	51.66 ± 0.46	52.95 ± 0.39	73.76 ± 0.19	58.24 ± 0.33	60.08 ± 0.35	57.93 ± 0.31
Div. Metric	59.81 ± 0.0	57.35 ± 0.83	53.98 ± 0.62	77.67 ± 0.73	75.19 ± 0.83	63.71 ± 0.77	79.25 ± 0.56
Div. Metric (Learned)	53.14 ± 0.0	55.35 ± 1.07	52.06 ± 0.77	65.60 ± 0.80	70.89 ± 0.62	54.48 ± 0.92	76.45 ± 0.74
Renyi Div.	56.51 ± 0.31	56.69 ± 0.38	54.90 ± 0.19	76.55 ± 0.32	76.57 ± 0.17	62.60 ± 0.32	80.32 ± 0.34
Merlin	50.58 ± 0.23	50.63 ± 0.26	50.37 ± 0.27	50.83 ± 0.34	52.35 ± 0.34	51.63 ± 0.38	51.01 ± 0.35
ODIN	54.86 ± 0.38	56.34 ± 0.38	53.31 ± 0.37	68.27 ± 0.32	74.10 ± 0.47	59.99 ± 0.36	78.99 ± 0.24
DOCTOR	54.96 ± 0.38	56.93 ± 0.40	53.35 ± 0.38	68.30 ± 0.30	74.56 ± 0.47	60.47 ± 0.34	79.08 ± 0.23
Mahalanobis	50.72 ± 0.65	51.01 ± 1.14	51.45 ± 0.77	53.09 ± 2.20	52.52 ± 1.81	53.29 ± 0.85	80.63 ± 1.38
Fisher-Rao	51.87 ± 0.66	56.29 ± 0.35	51.74 ± 0.55	58.29 ± 0.61	67.88 ± 0.39	54.50 ± 0.44	78.69 ± 0.16

Table 3.2: Comparison of different MIA Techniques. The AUROC score (%) on a balanced evaluation set is reported. 6k samples are uniformly selected from the training set (members) and 6k samples are uniformly selected from the test set (non-members). All the data selected is used for evaluation. The models marked (10) are trained on Cifar10, while the models marked as (100) are trained on Cifar100.

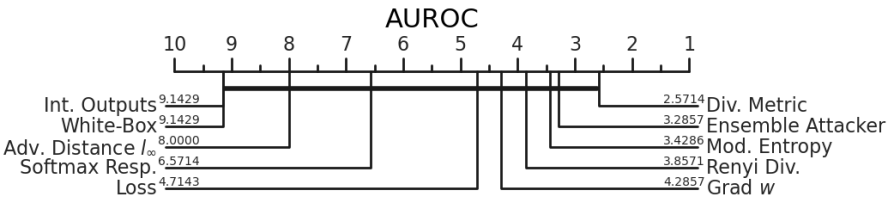


Figure 3.3: Critical Difference Diagram providing the average ranking of different MIA strategies based on their AUROC score. The average ranking is taken over all the different scenarios considered, but for each scenario a single value of the AUROC score is taken (the average over different cross-validation runs).

3.3 Membership Inference Attack Benchmark

Hereafter, we first present the experimental setting for MIAs and then provide numerical results on real world data. The code necessary to reproduce these experiments is available in our repository[3].

To evaluate a membership inference strategy, two groups of samples are needed: samples from the training set and samples outside the training set of the target model. The pre-trained models considered in this work are trained on $50k$ samples from the Cifar10 or Cifar100 datasets. The remaining $10k$ samples constitute the test set, which is used in our experiments as outside-of-the-training-set data.

The accuracy presented in Table 3.3 is computed by choosing a threshold along the ROC curve for each strategy. The threshold is chosen in order to maximize the

[3] https://github.com/ganeshdg95/Leveraging-Adversarial-Examples-to-Quantify-Membership-Information-Leakage

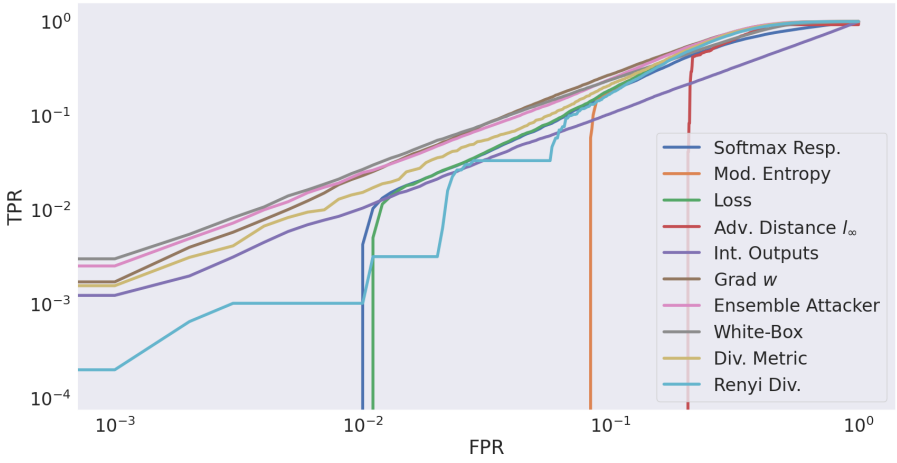


Figure 3.4: ROC curves of top performing strategies for membership inference evaluated against the AlexNet model trained on Cifar100. The curves are plotten of a log-log scale to emphasizes the performance on the low FPR region.

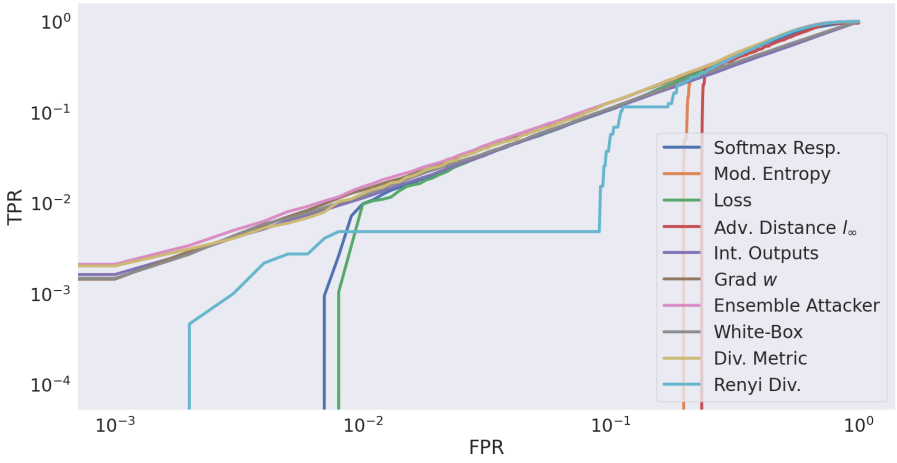


Figure 3.5: ROC curves of top performing strategies for membership inference evaluated against the ResNet model trained on Cifar100. The curves are plotten of a log-log scale to emphasizes the performance on the low FPR region.

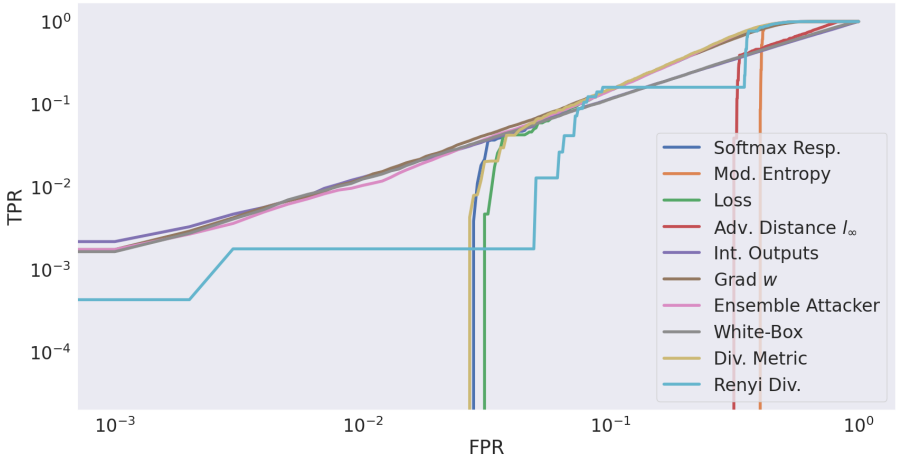


Figure 3.6: ROC curves of top performing strategies for membership inference evaluated against the DenseNet model trained on Cifar100. The curves are plotten of a log-log scale to emphasizes the performance on the low FPR region.

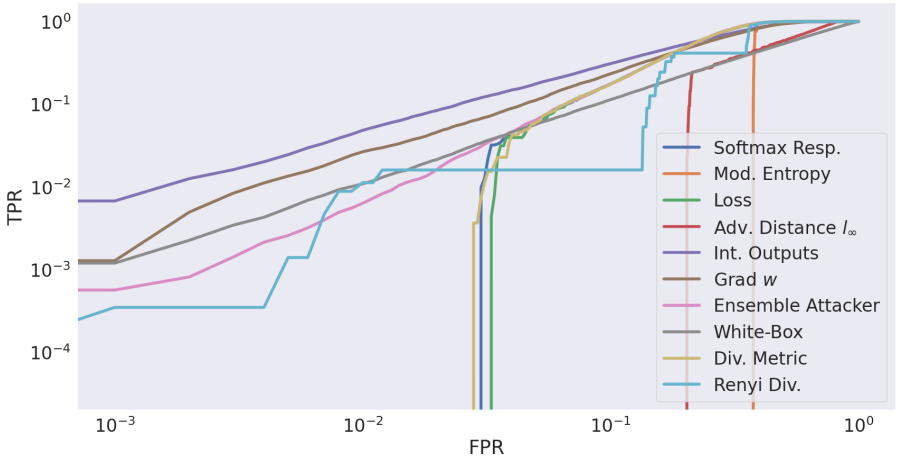


Figure 3.7: ROC curves of top performing strategies for membership inference evaluated against the ResNext model trained on Cifar100. The curves are plotten of a log-log scale to emphasizes the performance on the low FPR region.

Attack Strategy	AlexNet (10)	DenseNet (10)	ResNet (10)	AlexNet (100)	DenseNet (100)	ResNet (100)	ResNext (100)
Baseline	57.32 ± 0.17	51.64 ± 0.09	52.78 ± 0.09	74.31 ± 0.31	58.60 ± 0.13	60.80 ± 0.26	58.75 ± 0.19
Softmax Resp.	54.74 ± 0.27	56.64 ± 0.20	54.33 ± 0.33	65.52 ± 0.33	73.71 ± 0.28	60.11 ± 0.24	77.97 ± 0.21
Mod. Entropy	57.94 ± 0.28	56.44 ± 0.12	54.79 ± 0.30	74.50 ± 0.27	74.07 ± 0.23	62.69 ± 0.32	78.15 ± 0.20
Loss	57.91 ± 0.26	56.79 ± 0.18	54.79 ± 0.29	74.44 ± 0.29	74.08 ± 0.26	62.66 ± 0.31	78.18 ± 0.21
Grad. Norm x	57.91 ± 0.28	57.31 ± 0.12	54.88 ± 0.29	73.64 ± 0.30	74.40 ± 0.24	62.96 ± 0.30	78.33 ± 0.18
Grad. Norm w	58.12 ± 0.29	57.19 ± 0.15	54.86 ± 0.30	75.08 ± 0.31	74.4 ± 0.27	63.01 ± 0.32	78.43 ± 0.20
Grad. x	57.93 ± 0.27	54.30 ± 0.38	54.85 ± 0.31	73.64 ± 0.31	70.16 ± 0.74	62.94 ± 0.29	73.31 ± 1.68
Grad. w	57.99 ± 0.28	55.02 ± 0.52	55.21 ± 0.25	75.03 ± 0.30	72.14 ± 0.82	62.99 ± 0.31	72.49 ± 0.29
Int. Outputs	51.14 ± 0.40	52.64 ± 0.42	51.22 ± 0.44	50.89 ± 0.21	52.24 ± 0.70	51.07 ± 0.37	73.02 ± 0.36
White-Box	51.53 ± 0.70	50.69 ± 0.10	50.83 ± 0.25	70.64 ± 5.16	52.99 ± 2.10	52.22 ± 0.64	52.40 ± 1.13
Ensemble Attacker	58.10 ± 0.28	56.84 ± 0.96	54.93 ± 0.29	75.24 ± 0.30	74.39 ± 0.26	63.04 ± 0.32	78.39 ± 0.20
Adv. Distance l_2	57.37 ± 0.17	51.81 ± 0.10	52.93 ± 0.12	74.33 ± 0.31	58.64 ± 0.11	60.81 ± 0.26	58.77 ± 0.18
Adv. Distance l_∞	57.38 ± 0.19	51.83 ± 0.08	52.84 ± 0.09	74.32 ± 0.31	58.60 ± 0.12	60.81 ± 0.26	58.77 ± 0.17
Div. Metric	59.53 ± 0.0	57.14 ± 0.59	54.80 ± 0.40	74.66 ± 0.72	74.55 ± 0.55	62.85 ± 0.62	78.22 ± 0.52
Div. Metric (Learned)	55.39 ± 0.0	55.78 ± 0.46	53.55 ± 0.40	66.07 ± 0.80	69.25 ± 0.81	57.49 ± 0.63	74.13 ± 0.78
Renyi Div.	56.25 ± 0.19	56.52 ± 0.26	54.56 ± 0.16	73.53 ± 0.27	74.17 ± 0.29	62.41 ± 0.11	78.13 ± 0.16
Merlin	50.72 ± 0.21	51.05 ± 0.22	50.64 ± 0.18	50.90 ± 0.27	52.24 ± 0.22	51.54 ± 0.35	51.44 ± 0.26
ODIN	54.68 ± 0.30	55.62 ± 0.22	54.16 ± 0.33	65.55 ± 0.38	72.58 ± 0.29	59.69 ± 0.28	77.28 ± 0.21
DOCTOR	54.73 ± 0.28	56.64 ± 0.20	54.33 ± 0.32	65.52 ± 0.34	73.71 ± 0.29	60.13 ± 0.24	77.97 ± 0.21
Mahalanobis	51.13 ± 0.73	51.37 ± 0.96	51.58 ± 0.33	52.73 ± 1.61	52.19 ± 1.19	52.97 ± 0.71	75.43 ± 1.28
Fisher-Rao	51.74 ± 0.48	55.40 ± 0.25	51.64 ± 0.42	57.07 ± 0.46	64.72 ± 0.34	53.65 ± 0.42	74.54 ± 0.30

Table 3.3: Comparison of different MIA Techniques. The best accuracy (%) on a balanced evaluation set is reported. 6k samples are uniformly selected from the training set (members) and 6k samples are uniformly selected from the test set (non-members). All the data selected is used for evaluation. All the data selected is used for evaluation. The models marked (10) are trained on Cifar10, while the models marked as (100) are trained on Cifar100.

accuracy. A similar process is done in [85], where 80% of the data is used to determine the threshold that maximizes the accuracy, and then the accuracy is reported for the other 20% of the data.

We perform membership inference attacks using a total of 21 different strategies, which are listed and described throughout Sections 2.2 and 2.3. The Baseline, Softmax Response, Modified Entropy, Loss, ODIN, and DOCTOR strategies are black-box strategies, since the attacker only requires access to the target sample, its label and the output of the model (either the logits or Softmax response of the model). Note that this can be achieved with a single query to the target model, without need of additional samples from inside or outside the training set. On the other hand, the Gradient Norm w, Gradient Norm x and Adversarial Distance strategies are white-box, as the attacker requires access to the model parameters in order to compute gradients of the loss function. The Merlin strategy is black-box but it requires many queries to the target model. The rest of the listed strategies, namely Gradient w, Gradient x Intermediate Outputs, White-Box, Ensemble Attacker, Divergence Metric, Renyi Divergence, Mahalanobis and Fisher-Rao, require additional samples from the training set and additional samples from outside of the training set of the target model, in addition to white-box access. Strategies that require additional samples are provided 4k samples selected uniformly from the training set as *in-training* and 4 samples selected uniformly from the test set of the target model as *outside-of-the-training-set* data.

In our analysis we consider a balanced evaluation set and report the AUROC score (Table 3.2), the maximum accuracy (Table 3.3) and the false positive rate at true positive rate 95% (3.4) obtained for each strategy against each target model considered. In this setting, subsets of 6k samples are selected uniformly from the training set and the test set of the target model as *in-training* and *outside-of-the-training-set* data, respectively. Since the choice of these subsets influences our results, the experiments

Attack Strategy	AlexNet (10)	DenseNet (10)	ResNet (10)	AlexNet (100)	DenseNet (100)	ResNet (100)	ResNext (100)
Baseline	91.75 ± 0.24	0.0 ± 0.0	99.44 ± 0.07	92.50 ± 0.38	99.98 ± 0.01	95.76 ± 0.20	99.99 ± 0.00
Softmax Resp.	97.45 ± 0.25	99.99 ± 0.00	98.58 ± 0.20	98.67 ± 0.17	100.0 ± 0.0	98.95 ± 0.10	100.0 ± 0.0
Mod. Entropy	99.63 ± 0.07	100.0 ± 0.0	99.71 ± 0.04	100.0 ± 0.0	100.0 ± 0.0	99.92 ± 0.02	100.0 ± 0.0
Loss	99.71 ± 0.05	100.0 ± 0.0	99.73 ± 0.04	100.0 ± 0.0	100.0 ± 0.0	99.94 ± 0.03	100.0 ± 0.0
Grad. Norm x	99.03 ± 0.07	100.0 ± 0.0	99.63 ± 0.06	99.74 ± 0.05	100.0 ± 0.0	99.76 ± 0.05	100.0 ± 0.0
Grad. Norm w	99.47 ± 0.08	100.0 ± 0.0	99.68 ± 0.05	99.93 ± 0.02	100.0 ± 0.0	99.87 ± 0.03	100.0 ± 0.0
Grad. x	98.98 ± 0.10	99.96 ± 0.06	99.61 ± 0.06	99.74 ± 0.05	100.0 ± 0.0	99.76 ± 0.05	100.0 ± 0.0
Grad. w	99.49 ± 0.07	100.0 ± 0.0	99.68 ± 0.06	99.94 ± 0.02	100.0 ± 0.0	99.86 ± 0.03	100.0 ± 0.0
Int. Outputs	95.11 ± 0.35	97.03 ± 0.41	95.29 ± 0.31	94.74 ± 0.38	95.65 ± 0.64	94.92 ± 0.26	99.99 ± 0.00
White-Box	95.78 ± 0.67	95.38 ± 0.38	95.57 ± 0.32	99.69 ± 0.83	96.78 ± 1.28	96.30 ± 0.70	96.35 ± 0.93
Ensemble Attacker	99.72 ± 0.05	100.0 ± 0.0	99.69 ± 0.05	99.99 ± 0.00	100.0 ± 0.0	99.92 ± 0.03	100.0 ± 0.0
Adv. Distance l_2	91.75 ± 0.24	98.37 ± 0.24	99.44 ± 0.07	92.50 ± 0.38	99.98 ± 0.01	95.76 ± 0.20	99.99 ± 0.00
Adv. Distance l_∞	91.75 ± 0.24	98.36 ± 0.09	99.44 ± 0.07	92.50 ± 0.38	99.98 ± 0.01	95.76 ± 0.20	99.99 ± 0.00
Div. Metric	99.84 ± 0.0	100.0 ± 0.0	99.77 ± 0.14	99.98 ± 0.04	100.0 ± 0.0	99.75 ± 0.31	100.0 ± 0.0
Div. Metric (Learned)	98.75 ± 0.0	99.98 ± 0.03	99.29 ± 0.26	99.53 ± 0.21	99.98 ± 0.03	98.96 ± 0.26	99.98 ± 0.03
Renyi Div.	99.18 ± 0.11	99.99 ± 0.00	99.57 ± 0.08	99.99 ± 0.01	100.0 ± 0.0	99.94 ± 0.03	100.0 ± 0.0
Merlin	94.77 ± 0.51	94.74 ± 0.31	84.14 ± 7.05	93.77 ± 0.59	94.17 ± 0.30	95.06 ± 0.28	94.40 ± 0.33
ODIN	97.34 ± 0.29	99.95 ± 0.02	98.50 ± 0.16	98.67 ± 0.18	99.99 ± 0.01	98.61 ± 0.11	100.0 ± 0.0
DOCTOR	97.34 ± 0.21	99.99 ± 0.00	98.56 ± 0.18	98.57 ± 0.16	100.0 ± 0.0	99.04 ± 0.09	100.0 ± 0.0
Mahalanobis	95.07 ± 0.31	95.39 ± 0.45	95.6 ± 0.50	95.48 ± 0.79	95.54 ± 0.48	95.45 ± 0.75	99.90 ± 0.19
Fisher-Rao	95.54 ± 0.52	99.37 ± 0.14	95.58 ± 0.33	97.30 ± 0.28	99.98 ± 0.03	96.64 ± 0.33	100.0 ± 0.0

Table 3.4: Comparison of different MIA Techniques. The FPR at 95% TPR (%) on a balanced evaluation set is reported. 6k samples are uniformly selected from the training set (members) and 6k samples are uniformly selected from the test set (non-members). All the data selected is used for evaluation. All the data selected is used for evaluation. The models marked (10) are trained on Cifar10, while the models marked as (100) are trained on Cifar100.

are repeated 10 times, choosing a different subset each time for cross-validation. All the quantities reported are averaged over these 10 runs of the experiment and the error reported is the empirical standard deviation. The results of this analysis are reported in Tables 3.2 to 3.4. Note that the strategies are listed in the tables in the order they appear in previous sections.

To summarize these results we make use of Critical Difference diagrams, originally proposed in [24], which provide an average ranking of different methods across different scenarios based on a performance statistic. In our case the different MIA methods (the rows on Table 3.2) are ranked based on their AUROC score and the different scenarios considered are the different target models (the columns on Table 3.2). Thus Fig. 3.3 summarizes the results from Table 3.2. The lower the number in the diagram the higher the rank and the better the performance is for that method. The Diversity Metric method has an average rank of 2.57; being the lowest number, it is on average the best method across the different scenarios considered. Figure 3.3 is generated using the procedure described in [48], where the average rank comparison is replaced by a Wilcoxon signed-rank test [4]. The thick horizontal line across all methods means that they are not significantly different in terms of AUROC score, according to the Wilcoxon test with Holm's alpha correction (see [48] for more details).

Figures 3.4 to 3.7 show the ROC curves obtained for the top performing methods against the AlexNet, DenseNet, ResNet, and ResNext models trained on Cifar100, respectively. The curves are plotted on a log-log scale to emphasize the performance on the low false positive rate region, after the fashion of [17].

The best performing strategy when no additional samples are available is the

[4] Code necessary to generate Critical Difference diagrams is available at `https://github.com/hfawaz/cd-diagram`

Modified Entropy strategy, with the Diversity Metric strategy providing a marginal improvement at the cost of requiring additional samples to learn a suitable diversity measure. The diversity measure computed in closed form by using additional samples provides a criteria that is, for the purposes of a binary hypothesis test, equivalent to that of the Loss.This suggest that the value of the loss might be the optimal criteria given the output of the model. Statistically, the different in performance between the Loss strategy and the Diversity Metric strategy is not enough to justify the requirement of additional data.

Note that the Modified Entropy and Loss strategies perform consistently across all target models, regardless of how well these generalize, and even surpass the more resource hungry strategies in most cases. When additional samples are available, the Diversity Metric and Ensemble strategies are most effective. It is worth to mention that it might be infeasible for the attacker to obtain enough samples from the training set of the target model to launch these attacks, in which case Modified Entropy might be a better alternative.

3.4 Attribute Inference on PenDigits

To demonstrate the risk of information leakage from ML models, we consider attribute inference attacks against a model that classifies hand-written digits. We consider the PenDigits dataset [27], as it contains identity information about the writers, which we use as the sensitive attribute. The target model is a fully-connected network trained to classify hand-written digits. Details about the model and its training are provided, along with information about the dataset and its pre-processing, in Section 3.1. When performing MIAs, we utilize MSE, which is bounded, as the loss for training. This allows us to apply Theorem 2.

In our experiments we perform attribute inference attacks using each of the strategies described in Section 2.4 as we vary n. For each value of n, we randomly uniformly select 100 different training sets drawn from a pool containing a total of 11990 samples. For each training set we train a model. Subsequently, we apply each attack criteria to 100 training samples of each trained model. The success rate of the attacker is computed by counting the amount of times the attack is successful. The reported success rate is an average over different target models. Since there are 44 different writers in the data set, a random guess would amount to a success rate of approximately 2.3%.

The success rates for each strategy are computed and reported in Figure 3.8 (**Top**). For a small training set (100 samples), the attacker has a gain of 25% over a random guess. This decreases significantly with the size of the training set; however, even for a large training set, the attacker still has twice as much accuracy as a random guess.

Additionally, we perform MIAs against the same models. The attack strategy utilized is the Softmax response attack, given in Section 2.2. The procedure used for these experiments identical to that described in Section 3.2.2, with the exception that the size of the pool of training samples is of 8k and the size of the pool of test samples is of 3990. The success rate of the attacker, lower bound on the Bayesian attacker and accuracy of the model are presented in Fig. 3.8 (**Bottom**). We can observe that there is a significant leakage of membership information for low values of n, while this drops almost to the value of a random guess for large values ($n = 8000$).

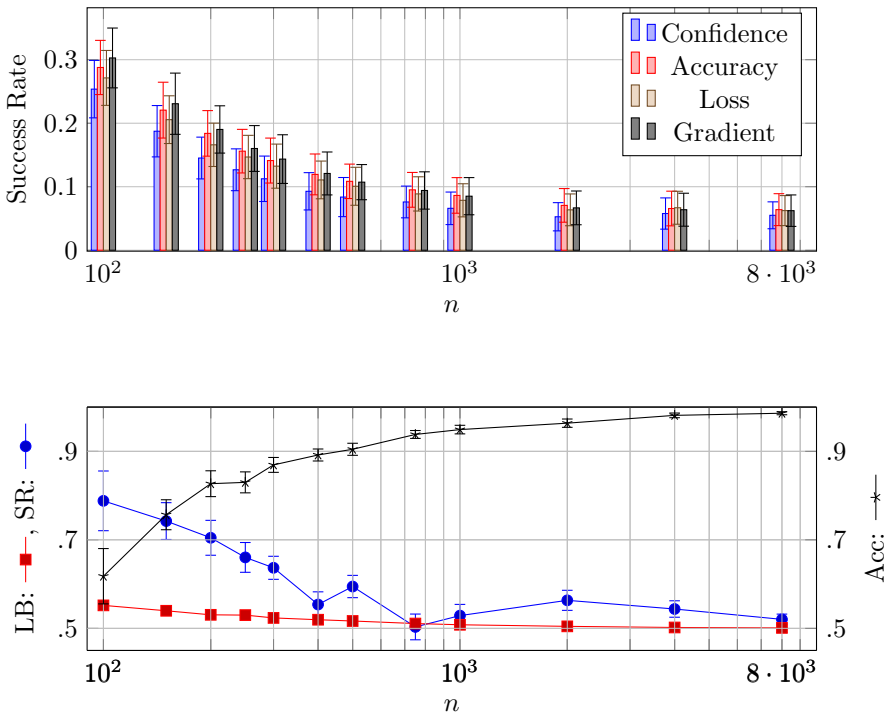


Figure 3.8: Attribute and Membership Inference Attacks on PenDigits for different sizes n of training sets. **Top:** Success Rate of different attribute inference attack strategies; **Bottom:** Success Rate of the Softmax Response attack (SR), Lower Bound (LB) and Accuracy (Acc; axis labels on the right).

Chapter 4

Conclusion

In Section 2.1 we presented the Bayesian attacker, which servers as an upper bound for the success rate of inference attacks. Such an upper bound guarantees the privacy of ML models, as no other strategy with the same resources and information can hope to attain a better success rate. Thus, if the Bayesian attacker presents no gain over a random guess, it is impossible for any other attacker to extract additional information from the model.

Complementary to this result, we presented a lower bound on the success rate of the Bayesian attacker, which connects the success rate of the attacker to the generalization gap of the target model. We found that a model that does not generalize well will be vulnerable to MIAs. Namely, if the target model suffers from a large generalization gap, the success rate of the Bayesian attacker is bound to be high. This implies that good generalization is a necessary condition for privacy. Nonetheless, good generalization does not guarantee privacy, as we proved by a theoretical example where the generalization gap can be made arbitrarily close to zero, while the success rate of a membership attack can be made arbitrarily close to one. Therefore, generalization is a necessary but not a sufficient condition for privacy.

Additionally, we analyzed the problem from the perspective of Information Theory. By considering the mutual information between the model parameters and their training set, we provided a concentration bound on the generalization gap of the target model. This bound is tighter than similar bounds previously presented in the PAC-Bayes learning literature, e.g. [8]. We found that the performance of the Bayesian attacker is upper bounded by the conditional mutual information between the target concept and the target model's parameters given the attacker's side-information. The mutual information appearing in the latter result is upper bounded by the mutual information appearing in the former result. From these results we conclude that both the generalization gap of the target model and the success rate of the Bayesian attacker can be controlled through the amount of information stored by the model's parameters about the training set.

The framework we proposed allowed us to have an overview of the problem and of the important concepts at play. For instance, we now have a deeper understanding of the role played by generalization on the vulnerability of the target model. The amount of information about the target model required to compute the success rate of the Bayesian attacker makes it unrealistic to implement it as an attack strategy in most

practical scenarios. Nonetheless, our results can be more easily applied during training in order to assess the privacy guarantees of ML models. Thus, these theoretical results can find utility in auditing and defending models against membership inference, rather than launching MIAs.

In Sections 2.2 and 2.3 we presented an array of strategies for membership inference. Section 2.2 provides a comprehensive list of MIA strategies present in the literature. Remark that many of the MIA strategies present in the literature take certain quantities (e.g. the value of the loss) and use them for training attack models (e.g. [74]). In this work, we took these quantities and used them directly as criteria to perform membership inference. This represents an advantage from the point of view of the attacker, as the attack requires less resources and information. The question that follows is whether the attack's performance deteriorates as a result of the diminished resources and information. As we were able to verify in our experiments, simpler attacks, i.e. those that require no extra samples or resources for training an attack model, tend to perform better.

We proposed the use of adversarial examples to launch MIAs against ML models. Empirically, we found the success rate of this strategy to be comparable to the state-of-the-art in some cases. However, the performance of the strategy in comparison to strategies such as the *Modified Entropy* strategy does not justify the requirement to have white-box access to the model. We proposed the use of diversity measures for membership inference, which achieves state-of-the-art performance. In a critical difference diagram, the diversity metric method achieved the highest average ranking over different target models and datasets. Yet again, the gain in performance over previous methods does not justify the use of additional samples to compute the desired diversity measure. In fact, after comparing a wide array of strategies (some from the literature, some proposed by us), we found that having white-box access to the target model, having access to additional resources, or having access to samples from the training set does not translate into a significant gain for the attacker, compared to less resource hungry methods.

In Section 2.3 we studied the use of OOD detection techniques for membership inference. We found that, although it is straight forward to apply most OOD detection techniques in the context of membership inference, the results given by these techniques are sub-optimal in comparison to other well established MIA methods. In practice, the difference in distribution between training data and test data is too subtle to be captured by OOD detection techniques. Our efforts to modify and re-adapt these methods were insufficient to produce a powerful membership inference strategy. As with previous methods, we saw that additional information did not result in an advantage for the attacker.

In Section 3.2, we illustrated how to implement the Bayesian attacker against a linear regression model trained to classify Gaussian data. Through this experiments we were able to validate our theoretical results. The success rate of the Bayesian attacker is lower bounded by a function of the generalization gap of the target model. The specific form of this bound hinges on the fact that the loss function used to train the target model in this setup is exponentially tail-bounded. To showcase the possible application of our framework to more realistic scenarios, we also computed the lower bound of the Bayesian attacker in the setup of DNNs for image classification. In this case, the specific form of the bound comes from the fact that the loss function used to train the models is itself bounded. In both experiments we observed how increasing the amount of samples in the training set of the target model led to a decrease in the generalization

gap and in the success rate of membership inference attacks.

In Section 3.3 we implemented and tested all the MIA strategies listed in Sections 2.2 and 2.3. We empirically determined the Diversity Measure strategy to be the most effective on average across all scenarios, with the Modified Entropy and Loss strategies coming close (with little statistical difference) in terms of performance. We hope that this benchmark serves as a starting point for future works that aim to develop novel and more powerful MIAs, providing inspiration and a reference point to compare new strategies. Through our analysis, we determined that even well generalizing models, such as the DenseNet and ResNext models, are susceptible to MIAs. This verifies our previous observation that good generalization is a necessary but not sufficient condition to prevent membership inference.

Empirically, we observed that methods that use additional samples from the training set or that train an attack model to perform membership inference do not significantly improve over methods that simply query the target model one time. Despite our extensive efforts we were not able to come up with strategies that effectively exploit extra resources and information. This suggests that the most relevant information to determine membership is concentrated in the output of the last layers of the target model, which is precisely the information used by methods such as Modified Entropy. As extensive as experiments can be, it is not possible to prove such conjecture empirically. An interesting direction for future work would be to provide a formal prove for this conjecture.

In Section 2.4 we proposed a list of techniques for attribute inference inspired in the state-of-the-art for membership inference. We found these techniques to be effective and easily applied in the context of hand-written digit classification with the PenDigits dataset. Our experiments in Section 3.4 showed that the success rate of AIA can be tightly correlated to the success rate of MIAs. Given the nature of attribute inference attacks, it is non-trivial to determine a setup in which all strategies can be tested. In fact, most of the AIA strategies proposed in the literature are tailored to specific target models. An interesting, and much necessary direction for research in this field is to determine a common setup in which different AIA strategies can be tested and compared.

4.1 Patents

During the course of my PhD, I did an internship at the Ericsson research center in France. During this time, I co-authored two patents. The patents have not yet been published by the time of submission of this manuscript, therefore, I do not have the liberty to disclose any details about their contents. Nonetheless, we list the patents and provide a brief description:

- **Predictive canary testing.** Canary testing is a technique used for testing and gradually releasing a new version of an application. Canary testing allows live testing of the new version of the application by releasing it to a small number of users. Predictive canary testing aims to prevent deterioration in the quality of service of users that are part of the test group. Our novel technique allows to predict the behavior of the new version of the software and to control its release in the context of telecommunications.
- **Decision making based on interdependencies between different levels related to an anomaly in telecommunication networks.** Anomaly detection is vital to ensure quality of service in telecommunications systems. When a failure occurs, it often propagates through several layers of the system, and the level at which it is detected is not necessarily the one in which it was produced. Our solution aims to exploit the interdependencies between different levels in telecommunications systems to detect anomalies and to determine not only their origin, but the extent to which they propagate over different levels.

List of Figures

List of Tables

Glossary

AIA Attribute Inference Attack. 1, 2, 8, 10, 50, 72

DNN Deep Neural Network. 13, 52, 54, 55, 57, 71

DP Differential Privacy. 1, 7, 11–13

MIA Membership Inference Attack. 1, 2, 7–9, 11–13, 15–17, 22–25, 30, 33, 35, 37, 42, 44–47, 50, 52, 54, 55, 57, 59, 60, 62, 65, 66, 68, 70–72, 96, 97

ML Machine Learning. 1, 2, 7, 9–13, 21, 55, 57, 68, 70, 71

MSE mean squared error. 53, 54, 59

OOD Out of Distribution. 1, 9, 21, 44–46, 48, 71

pdf probability density function. 14, 19, 25, 49, 89

pmf probability mass function. 14

Chapter 5

Bibliography

[1] The california consumer privacy act. 2018.

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 308–318, New York, NY, USA, 2016. Association for Computing Machinery.

[3] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018.

[4] Dario Amodei, Christopher Olah, Jacob Steinhardt, Paul Francis Christiano, John Schulman, and Dandelion Mané. Concrete problems in ai safety. *ArXiv*, abs/1606.06565, 2016.

[5] Colin Atkinson and Ann F. S. Mitchell. Rao's distance measure. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 43(3):345–365, 1981.

[6] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning, 2019.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[8] Raef Bassily, Shay Moran, Ido Nachum, Jonathan Shafer, and Amir Yehudayoff. Learners that use little information. In Firdaus Janoos, Mehryar Mohri, and Karthik Sridharan, editors, *Proceedings of Algorithmic Learning Theory*, volume 83 of *Proceedings of Machine Learning Research*, pages 25–55. PMLR, 07–09 Apr 2018.

[9] Raef Bassily, Shay Moran, Ido Nachum, Jonathan Shafer, and Amir Yehudayoff. Learners that use little information. In *Algorithmic Learning Theory*, pages 25–55. PMLR, 2018.

[10] Samyadeep Basu, Rauf Izmailov, and Chris Mesterharm. Membership model inversion attacks for deep networks. *NeurIPS 2019, Workshop on Privacy in Machine Learning*, abs/1910.04257, 2019.

[11] Thomas Baumhauer, P. Schöttle, and M. Zeppelzauer. Machine unlearning: Linear filtration for logit-based classifiers. *ArXiv*, abs/2002.02730, 2020.

[12] Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[13] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155, 2003.

[14] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 1175–1191, New York, NY, USA, 2017. Association for Computing Machinery.

[15] Valerii V Buldygin and Yu V Kozachenko. Sub-gaussian random variables. *Ukrainian Mathematical Journal*, 32(6):483–489, 1980.

[16] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.

[17] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP) (SP)*, pages 1519–1519, Los Alamitos, CA, USA, may 2022. IEEE Computer Society.

[18] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium*, pages 267–284, Santa Clara, CA, Aug. 2019. USENIX Association.

[19] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 343–362, New York, NY, USA, 2020. Association for Computing Machinery.

[20] Yufei Chen, Chao Shen, Yun Shen, Cong Wang, and Yang Zhang. Amplifying membership exposure via data poisoning. In *NeurIPS 2022*, November 2022.

[21] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.

[22] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2206–2216. PMLR, 13–18 Jul 2020.

[23] Ganesh Del Grosso, Hamid Jalalzai, Georg Pichler, Catuscia Palamidessi, and Pablo Piantanida. Leveraging adversarial examples to quantify membership

information leakage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10399–10409, June 2022.

[24] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[26] Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Jialin Pan, Tao Qin, Renjun Xu, and Chongjun Wang. Adarnn: Adaptive learning and forecasting of time series. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021.

[27] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[28] Cynthia Dwork. Differential privacy. In *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II*, ICALP'06, page 1–12, Berlin, Heidelberg, 2006. Springer-Verlag.

[29] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, Aug. 2014.

[30] David Forsyth and Jean Ponce. *Computer vision: A modern approach.* Prentice hall, 2011.

[31] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1322–1333, New York, NY, USA, 2015. Association for Computing Machinery.

[32] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 17–32, San Diego, CA, Aug. 2014. USENIX Association.

[33] Andrea Galassi, Marco Lippi, and Paolo Torroni. Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10):4291–4308, 2021.

[34] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(2):652–662, feb 2021.

[35] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[36] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4878–4887, 2017.

[37] Alison L. Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International statistical review*, 70(3):419–435, 2002.

[38] Eduardo Dadalto Camara Gomes, Florence Alberge, Pierre Duhamel, and Pablo Piantanida. Igeood: An information geometry approach to out-of-distribution detection. In *International Conference on Learning Representations*, 2022.

[39] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[40] Federica Granese, Marco Romanelli, Daniele Gorla, Catuscia Palamidessi, and Pablo Piantanida. DOCTOR: A simple method for detecting misclassification errors. In *Annual Conference on Neural Information Processing Systems 2021 (NeurIPS 2021), December 7-10, 2021*, volume abs/2106.02395, 2021. (Spotlight).

[41] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019(1):133 – 152, 2019.

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[43] A Hern. Royal free breached uk data law in 1.6 m patient deal with google's deepmind', guardian, 3 july 2017, 2017.

[44] S. Hidano, T. Murakami, S. Katsumata, S. Kiyomoto, and G. Hanaoka. Model inversion attacks for prediction systems: Without knowledge of non-sensitive attributes. In *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, pages 115–11509, 2017.

[45] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: Information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 603–618, New York, NY, USA, 2017. Association for Computing Machinery.

[46] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[47] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[48] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33, 07 2019.

[49] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1895–1912, Santa Clara, CA, Aug. 2019. USENIX Association.

[50] Bargav Jayaraman, Lingxiao Wang, David E. Evans, and Quanquan Gu. Revisiting membership inference under realistic assumptions. *Proceedings on Privacy Enhancing Technologies*, 2021:348 – 368, 2021.

[51] Simon Jegou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[52] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 259–274, New York, NY, USA, 2019. Association for Computing Machinery.

[53] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[54] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[55] Jared L. Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1), Feb 2018.

[56] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[57] Achim Klenke. *Probability Theory*. Springer, Sept. 2013.

[58] Jakub Konecny, H. Brendan McMahan, Daniel Ramage, and Peter Richtarik. Federated optimization: Distributed machine learning for on-device intelligence, 2016.

[59] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

[60] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[61] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016.

[62] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[63] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 7167–7177, Red Hook, NY, USA, 2018. Curran Associates Inc.

[64] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In *Proceedings of the 29th USENIX Conference on Security Symposium*, SEC'20, USA, 2020. USENIX Association.

[65] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

[66] Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

[67] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A. Gunter, and Kai Chen. Understanding membership inferences on well-generalized learning models. *CoRR*, abs/1802.04889, 2018.

[68] Fred Lu, Joseph Munoz, Maya Fuchs, Tyler LeBlond, Elliott V. Zaresky-Williams, Edward Raff, Francis Ferraro, and Brian Testa. A general framework for auditing differentially private machine learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[69] Alexander Selvikvåg Lundervold and Arvid Lundervold. An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik*, 29(2):102 – 127, 2019. Special Issue: Deep Learning in Medical Physics.

[70] Michael Lustig, David L Donoho, Juan M Santos, and John M Pauly. Compressed sensing mri. *IEEE signal processing magazine*, 25(2):72–82, 2008.

[71] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[72] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706, 2019.

[73] Viraaji Mothukuri, Reza M. Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

[74] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. *2019 IEEE Symposium on Security and Privacy (SP)*, May 2019.

[75] Jerzy Neyman and Egon Sharpe Pearson. IX. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

[76] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[77] Li Pengcheng, Jinfeng Yi, and Lijun Zhang. Query-efficient black-box attack by active learning. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1200–1205, 2018.

[78] NhatHai Phan, My T. Thai, Han Hu, Ruoming Jin, Tong Sun, and Dejing Dou. Scalable differential privacy with certified robustness in adversarial learning, 2020.

[79] Marine Picot, Francisco Messina, Malik Boudiaf, Fabrice Labeau, Ismail Ben Ayed, and Pablo Piantanida. Adversarial robustness via fisher-rao regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022.

[80] Julianna Pinele, João E. Strapasson, and Sueli I. R. Costa. The fisher-rao distance between multivariate normal distributions: Special cases, boundsand applications. *Entropy*, 22(4):404, 2020.

[81] Yury Polyanskiy and Yihong Wu. Lecture notes on information theory. *Lecture Notes for ECE563 (UIUC)*, 2019.

[82] C.Radhakrishna Rao. Diversity and dissimilarity coefficients: A unified approach. *Theoretical Population Biology*, 21(1):24–43, 1982.

[83] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[84] General Data Protection Regulation. Regulation eu 2016/679 of the european parliament and of the council of 27 april 2016. *Official Journal of the European Union*, 2016.

[85] S. Rezaei and X. Liu. On the difficulty of membership inference attacks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7888–7896, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society.

[86] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Herve Jegou. White-box vs black-box: Bayes optimal strategies for membership inference. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *ICML*, volume 97 of *Proc. of Machine Learning Research*, pages 5558–5567. PMLR, June 2019.

[87] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. Updates-leak: Data set inference and reconstruction attacks in online learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*, SEC'20, USA, 2020. USENIX Association.

[88] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

[89] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 909–910, 2015.

[90] R. Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

[91] Jinhyun So, Basak Guler, and A. Salman Avestimehr. A scalable approach for privacy-preserving collaborative machine learning, 2020.

[92] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 587–601, New York, NY, USA, 2017. Association for Computing Machinery.

[93] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security Symposium*, pages 2615–2632, 2021.

[94] Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, Nov 2019.

[95] Jasper Tan, Blake Mason, Hamid Javadi, and Richard Baraniuk. Parameters or privacy: A provable tradeoff between overparameterization and membership inference. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[96] Colin Tankard. What the gdpr means for businesses. *Network Security*, 2016(6):5–8, 2016.

[97] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, page 601–618, USA, 2016. USENIX Association.

[98] Aleksei Triastcyn and Boi Faltings. Improved accounting for differentially private learning. *CoRR*, abs/1901.09697, 2019.

[99] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, pages 1–1, 2019.

[100] Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[101] Tim van Erven and Peter Harremoës. Rényi divergence and kullback-leibler divergence. *IEEE Trans. Inf. Theory*, 60(7):3797–3820, 2014.

[102] Rui Wang, Yihe Dong, Sercan O Arik, and Rose Yu. Koopman neural operator forecaster for time-series with temporal distributional shifts. In *The Eleventh International Conference on Learning Representations*, 2023.

[103] Derek Wong and Stephen Yip. Machine learning classifies cancer, 2018.

[104] Xi Wu, Matthew Fredrikson, Somesh Jha, and Jeffrey F. Naughton. A methodology for formalizing model-inversion attacks. In *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, pages 355–370, 2016.

[105] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[106] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[107] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[108] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*,

CCS '19, page 225–240, New York, NY, USA, 2019. Association for Computing Machinery.

[109] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282, Los Alamitos, CA, USA, jul 2018. IEEE Computer Society.

[110] B. Zhao, A. Agrawal, C. Coburn, H. Asghar, R. Bhaskar, M. Kaafar, D. Webb, and P. Dickinson. On the (in)feasibility of attribute inference attacks on machine learning models. In *2021 IEEE European Symposium on Security and Privacy (EuroSamp;P)*, pages 232–251, Los Alamitos, CA, USA, sep 2021. IEEE Computer Society.

[111] Benjamin Zi Hao Zhao, Mohamed Ali Kaafar, and Nicolas Kourtellis. Not one but many tradeoffs. *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, Nov 2020.

Appendix A

Experimental Details

Most of the code for the experiments in Sections 3.2.1, 3.2.2 and 3.4 was run on a Latitude-7400 computer, with an Intel Core i7-8665U CPU @ 1.90GHz x 8 Processor. Part of the experiments were run on a server with two NVIDIA Quadro RTX 6000 GPUs and an AMD EPYC 7302 16-Core processor.

The code and instructions necessary to reproduce the experiments in Sections 3.2.1, 3.2.2 and 3.4 can be found at `https://github.com/anonymus369/Formalizing-Attribute-and-Membership-Inference`.

Most of the code for the experiments in Section 3.3 was run on a cluster with multiple nodes, each with NVIDIA Quadro RTX 6000 GPUs and an AMD EPYC 7302 16-Core processor.

The code and instructions necessary to reproduce the experiments in Section 3.3 can be found at `https://github.com/ganeshdg95/Leveraging-Adversarial-Examples-to-Quantify-Membership-Information-Leakage`.

Appendix B

Proof of Proposition 1

We recall the definition of the total variation distance when applied to distributions P, Q on a set $\mathcal{X} \subseteq \mathbb{R}^d$ and Scheffé's identity [100, Lemma 2.1]

$$\|P - Q\|_{\mathrm{TV}} \triangleq \sup_{\mathcal{A} \in \mathcal{B}^d} |P(\mathcal{A}) - Q(\mathcal{A})| = \frac{1}{2} \int |p_X(\mathbf{x}) - q_X(\mathbf{x})| d\mu(\mathbf{x}) \,, \tag{B.1}$$

with respect to a base measure μ, where $\mathcal{B}^d$ denotes the class of all Borel sets on $\mathbb{R}^d$.

Proof. First of all, we prove equality for $\gamma = 1$. Let us denote the optimal decision regions with $\mathcal{T}^\star \equiv \mathcal{T}(1)$ and $\mathcal{T}^{\star c} \equiv \mathcal{T}^c(1)$ (cf. Definition 6). Let $\epsilon_0(\mathcal{T}^{\star c})$ and $\epsilon_1(\mathcal{T}^\star)$ the Type-I and Type-II errors. Then,

$$\begin{aligned}
\epsilon_1(\mathcal{T}^\star) + \epsilon_0(\mathcal{T}^{\star c}) &= \int_{\mathcal{T}^\star} p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|0)\, d\theta ds + \int_{\mathcal{T}^{\star c}} p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|1)\, d\theta ds \\
&= \int_{\mathcal{T}^\star} \min_{t \in \{0,1\}} p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|t)\, d\theta ds + \int_{\mathcal{T}^{\star c}} \min_{t \in \{0,1\}} p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|t)\, d\theta ds \\
&= \int_{\Theta \times \mathcal{S}} \min_{t \in \{0,1\}} p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|t)\, d\theta ds \\
&= 1 - \left\| p_{\widehat{\theta}(\mathbf{Z})S|T}(\cdot|1) - p_{\widehat{\theta}(\mathbf{Z})S|T}(\cdot|0) \right\|_{\mathrm{TV}} = 1 - \Delta,
\end{aligned} \tag{B.2}$$

where the last identity follows by applying Scheffé's identity (B.1). From (B.2), we have for any decision region $\widehat{\mathcal{T}} \subseteq \Theta \times \mathcal{S}$,

$$\begin{aligned}
1 - \Delta &= \int_{\Theta \times \mathcal{S}} \min_{t \in \{0,1\}} p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|t)\, d\theta ds \\
&= \int_{\widehat{\mathcal{T}}} \min_{t \in \{0,1\}} p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|t)\, d\theta ds + \int_{\widehat{\mathcal{T}}^c} \min_{t \in \{0,1\}} p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|t)\, d\theta ds \\
&\leq \int_{\widehat{\mathcal{T}}} p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|0)\, d\theta ds + \int_{\widehat{\mathcal{T}}^c} p_{\widehat{\theta}(\mathbf{Z})S|T}(\theta, s|1)\, d\theta ds \\
&= \epsilon_1(\widehat{\mathcal{T}}) + \epsilon_0(\widehat{\mathcal{T}}^c).
\end{aligned} \tag{B.3}$$

It remains to show (2.8), assuming that $\mathrm{P}\{T=1\} = \mathrm{P}\{T=0\} = 1/2$. Using (B.2), we have

$$\begin{aligned} \frac{1}{2}\big[1-\Delta\big] &= \frac{1}{2}\big[\epsilon_1(\mathcal{T}^\star) + \epsilon_0(\mathcal{T}^{\star c})\big] \\ &= \inf_{\varphi} \mathrm{P}\left\{\varphi(\widehat{\theta}(\mathbf{Z}), S) \neq T\right\}, \end{aligned} \tag{B.4}$$

where the last identity follows by the definition of the decision regions. □

Appendix C

Proof of Theorem 4

Proof. Using the definitions from the previous proofs, let $R \triangleq \varrho(\widehat{\theta}(\mathbf{Z}), \overline{X}, \overline{Y})$ be the square of a sub-Gaussian random variable $R_{\mathrm{SG}} \triangleq \sqrt{|R|}$ with variance proxy σ_R^2. Then, we have $\mathrm{P}\{R \geq r^2\} = \mathrm{P}\{|R_{\mathrm{SG}}| \geq r\} \leq 2e^{-\frac{r^2}{2\sigma_R^2}}$ for all $r \geq 0$, which in turn yields $\mathrm{P}\{R \geq r\} \leq 2e^{-\frac{r}{2\sigma_R^2}}$ for all $r \geq 0$. Define the random variable R_0 to have the distribution function $Q_0(r) \triangleq P\{R_0 \leq r\} \triangleq 1 - 2e^{-\frac{r}{2\sigma_R^2}}$ on its support $[r_0, \infty)$, where $r_0 = 2\sigma_R^2 \log 2$, i.e., the pdf of R_0 is $p_{R_0}(r) = \frac{1}{\sigma_R^2} e^{-\frac{r}{2\sigma_R^2}}$.

Let Q be the distribution function of R. Then, using the construction in the proof of in [57, Theorem 1.104], we can write $R = Q^{-1} \circ Q_0(R_0)$, where Q^{-1} is the left continuous inverse of Q, noting that Q_0 is continuous. The tail bound on R then implies $Q(r) = 1 - P\{|R| \geq r\} \geq Q_0(r)$, which immediately yields $Q^{-1} \circ Q_0(r) \leq r$.

Following similar steps as for Theorem 3, for $R_{\max} \geq r_0$, we get,

$$
\begin{aligned}
\int_{|r| \geq R_{\max}} |r| p_R(r) dr &= \int_{Q^{-1} \circ Q_0(r) \geq R_{\max}} Q^{-1} \circ Q_0(r) p_{R_0}(r) dr \\
&= \int_{Q_0(r) \geq Q(R_{\max})} Q^{-1} \circ Q_0(r) p_{R_0}(r) dr \\
&\leq \int_{Q_0(r) \geq Q(R_{\max})} r p_{R_0}(r) dr \\
&\leq \int_{r \geq R_{\max}} r p_{R_0}(r) dr \\
&= \int_{R_{\max}}^{\infty} \frac{r}{\sigma_R^2} e^{-\frac{r}{2\sigma_R^2}} dr \\
&= -2 \int_{R_{\max}}^{\infty} \frac{\partial\, r e^{-\frac{r}{2\sigma_R^2}}}{\partial r} dr + 2 \int_{R_{\max}}^{\infty} e^{-\frac{r}{2\sigma_R^2}} dr \\
&= 2R_{\max} \exp\left(-\frac{R_{\max}}{2\sigma_R^2}\right) \left(1 + \frac{2\sigma_R^2}{R_{\max}}\right). \qquad \text{(C.1)}
\end{aligned}
$$

The rest of the proof follows identically to that of Theorem 3 and yields,

$$\mathcal{P}_{\mathrm{Suc}}(\varphi) \geq \max\left\{ P_{\mathrm{m}}, P_{\mathrm{m}}\left(\frac{|\mathbb{E}[\mathcal{E}_{\mathrm{G}}(\mathcal{A}, \mathbf{Z})]|}{2R_{\max}} \right.\right. \\ \left.\left. - \frac{1}{1-P_{\mathrm{m}}} \exp\left(-\frac{R_{\max}}{2\sigma_R^2} \right)\left(1 + \frac{2\sigma_R^2}{R_{\max}} \right) - 1 \right) + 1 \right\}. \tag{C.2}$$

□

Appendix D

Proof of Theorem 5

Before we proceed with the proof of Theorem 5, we provide a series of definitions and preliminary results.

D.1 Basic Definitions and Change of Measure

Let us consider two probability measures P and Q on a common measurable space $(\Omega, \mathcal{F})$. Let X denote a random variable $X : \Omega \to \mathcal{X}$ and P_X, Q_X correspond to the induced distributions. Assuming absolute continuity $P_X \ll Q_X$, the KL-divergence of Q_X with respect to P_X is defined by

$$D_{\mathrm{KL}}(P_X \| Q_X) \triangleq \mathbb{E}_{Q_X}\left[-\log\left(\frac{dP_X}{dQ_X}\right)\right]. \tag{D.1}$$

Consider a kernel (or channel) according to the law $P_{Y|X}$ that produces the random variable Y given X. Let P_Y be the induced distribution of Y when X is generated according to P_X while Q_Y is the distribution of Y when X is generated according to Q_X. Then, by the data-processing inequality for KL-divergence [81, Theorem 2.2 6.], we have

$$D_{\mathrm{KL}}(P_X \| Q_X) \geq D_{\mathrm{KL}}(P_Y \| Q_Y). \tag{D.2}$$

Equality holds if and only if $P_{X|Y} = Q_{X|Y}$, where $P_{X|Y}P_Y = P_{Y|X}P_X$ and $Q_{X|Y}Q_Y = P_{Y|X}P_X$. A simple application of this inequality leads to the following result.

Lemma 1 (Data-processing reduces KL-divergence [21])**.** *For any measurable set $\mathcal{B} \in \mathcal{F}(\mathcal{X})$, inequality (D.2) applied to the degenerate channel based on the indicator function $Y = \mathbb{1}\{X \in \mathcal{B}\}$ implies:*

$$D_{KL}(P_X \| Q_X) \geq d_{KL}(p_{\mathcal{B}} \| q_{\mathcal{B}}) = d_{KL}(1 - p_{\mathcal{B}} \| 1 - q_{\mathcal{B}}), \tag{D.3}$$

where $d_{KL}(\cdot\|\cdot)$ denotes the binary KL-divergence with parameters $p_{\mathcal{B}} = P_X(\mathcal{B})$ and $q_{\mathcal{B}} = Q_X(\mathcal{B})$. Note that if $t \in [0, 1]$ and $M > 1$, then

$$\log_2(M) - d_{KL}(t\|1 - 1/M) = t \log_2(M - 1) + H_2(t)\ , \tag{D.4}$$

where $H_2(t) \triangleq -t \log_2 t - (1 - t) \log_2(1 - t)$ is the binary entropy function. Equality holds in (D.3) if and only if $P_{X|X\in\mathcal{B}} = Q_{X|X\in\mathcal{B}}$ and $P_{X|X\notin\mathcal{B}} = Q_{X|X\notin\mathcal{B}}$.

The proof of this lemma is rather straightforward from basic properties and will be omitted. We next revisit a well-known result to obtain bounds for the probability of an arbitrary event $\mathcal{B} \in \mathcal{F}(\mathcal{X})$.

Lemma 2 (Change of measure [9])**.** *Let the distributions P_X and Q_X be induced by the random variable X as described. Then,*

$$\sup_{\mathcal{B}\in\mathcal{F}(\mathcal{X})} P_X(\mathcal{B}) \log_2(1/Q_X(\mathcal{B})) \leq D_{KL}(P_X\|Q_X) + 1, \tag{D.5}$$

where the supremum is taken over all measurable sets $\mathcal{B} \in \mathcal{F}(\mathcal{X})$.

Proof. For any set $\mathcal{B} \in \mathcal{F}(\mathcal{X})$, we have by Lemma 1, that

$$\begin{aligned} D_{\mathrm{KL}}(P_X\|Q_X) &\geq d_{\mathrm{KL}}(p_{\mathcal{B}}\|q_{\mathcal{B}}) \\ &= P_X(\mathcal{B}) \log \frac{P_X(\mathcal{B})}{Q_X(\mathcal{B})} + P_X(\mathcal{B}^c) \log \frac{P_X(\mathcal{B}^c)}{Q_X(\mathcal{B}^c)} \\ &= P_X(\mathcal{B}) \log_2 \frac{1}{Q_X(\mathcal{B})} + P_X(\mathcal{B}^c) \log_2 \frac{1}{Q_X(\mathcal{B}^c)} - H_2(p_{\mathcal{B}}) \\ &\geq P_X(\mathcal{B}) \log_2 \left(\frac{1}{Q_X(\mathcal{B})}\right) - 1. \end{aligned} \tag{D.6}$$

The final inequality (D.5) follows by taking the supremum over all measurable sets $\mathcal{B} \in \mathcal{F}(\mathcal{X})$ in (D.6). □

D.1.1 Cramér-Chernoff Method

We recall a distribution-dependent deviation bound based on the optimization of the Markov inequality which is known as Cramér-Chernoff method.

Let Z be a real-valued random variable and define its log-moment-generating function as

$$\psi_Z(\lambda) = \log \mathbb{E}\left[\exp \lambda Z\right], \quad \lambda \geq 0. \tag{D.7}$$

For $\lambda \geq 0$, the Markov inequality implies:

$$\begin{aligned} \mathrm{P}(Z \geq t) &\leq \mathrm{P}(e^{\lambda Z} \geq e^{\lambda t}) \\ &\leq e^{-\lambda t}\mathbb{E}[e^{\lambda Z}] \\ &= \exp\left[-\lambda t + \psi_Z(\lambda)\right]. \end{aligned} \tag{D.8}$$

As (D.8) holds for any $\lambda \geq 0$, we immediately obtain $\mathrm{P}(Z \geq t) \leq \exp\left[-\psi_Z^*(t)\right]$ for $t \geq \mathbb{E}[Z]$, where

$$\psi_Z^*(t) = \sup_{\lambda\in\mathbb{R}} \left\{\lambda t - \log \mathbb{E}\left[e^{\lambda Z}\right]\right\}. \tag{D.9}$$

This expression is known as the Fenchel-Legendre dual function of $\psi_Z(\lambda)$ and it equals $\psi_Z^*(t) = \sup\{\lambda t - \psi_Z(\lambda) : \lambda \geq 0\}$ whenever $t \geq \mathbb{E}[Z]$.

And therefore, for $t \geq \mathbb{E}[Z]$,

$$\mathrm{P}(Z \geq t) \leq \exp\left[-\psi_Z^*(t)\right]. \tag{D.10}$$

We will need the following properties:

- If $Z = X_1 + \cdots + X_n$ with $\{X_i\}_{i=1}^n$ being i.i.d. copies of X, then

$$\psi_Z^*(t) = n\psi_X^*\left(\frac{t}{n}\right); \tag{D.11}$$

- For any random variable Z,

$$\psi_{Z/n}^*(t) = \psi_Z^*(nt). \tag{D.12}$$

An immediate consequence of these properties is that the random variable

$$Z = \mathbb{E}[X] - \frac{1}{n}\sum_{i=1}^n X_i,$$

with $\{X_i\}_{i=1}^n$ i.i.d. copies of X, satisfies

$$\mathrm{P}(Z \geq t) \leq \exp\left[-n\psi_{\mathbb{E}[X]-X}^*(t)\right], \ \forall t \geq 0. \tag{D.13}$$

D.2 Proof

Using these preliminary results, we are now ready to prove Theorem 5. The proof requires three steps which are described below.

Information loss. First, we observe that $(T, S) \leftrightarrow \mathbf{Z} \leftrightarrow \widehat{\theta}(\mathbf{Z})$ and thus $T \leftrightarrow (\mathbf{Z}, S) \leftrightarrow \widehat{\theta}(\mathbf{Z})$ form Markov chains since $\widehat{\theta}$ is a stochastic function of $\mathbf{Z}$. As a consequence of the data-processing inequality [21, Theorem 2.8.1], we obtain (2.23) from

$$I(T; \widehat{\theta}(\mathbf{Z})|S) \leq I(\mathbf{Z}; \widehat{\theta}(\mathbf{Z})|S) = I(\mathbf{Z}; \widehat{\theta}(\mathbf{Z})) - I(S; \widehat{\theta}(\mathbf{Z})). \tag{D.14}$$

Interestingly, we will show that $I(T; \widehat{\theta}(\mathbf{Z})|S)$ bounds the accuracy of the membership (sensitive attribute) inference while $I(\mathbf{Z}; \widehat{\theta}(\mathbf{Z}))$ bounds the generalization gap of the hypothesis associated with $\widehat{\theta}$.

Generalization gap. The proof of the bound on the generalization gap (2.21) easily follows from application of well-known results. For $\epsilon \geq 0$ let us define the region

$$\mathcal{B} \triangleq \left\{(\theta, \mathbf{z}) \in \Theta \times \mathcal{Z}^n : \mathbb{E}[\varrho(\theta, Z)] - \frac{1}{n}\sum_{i=1}^n \varrho(\theta, z_i) \geq \epsilon\right\}. \tag{D.15}$$

By the definition of the generalization gap (Definition 2), we have $\mathcal{G}_{\mathrm{G}}(\mathcal{A}, \mathbf{Z}) \geq \epsilon$ if and only if $\left(\widehat{\theta}(\mathbf{Z}), \mathbf{Z}\right) \in \mathcal{B}$. We define the associated fibers $\mathcal{B}_{(\theta)} \triangleq \{\mathbf{z} \in \mathcal{Z}^n : (\theta, \mathbf{z}) \in \mathcal{B}\}$ for $\theta \in \Theta$. First, we apply the Cramér-Chernoff method (D.1.1) to the random variable

$$R_\theta \triangleq \mathbb{E}[\varrho(\theta, (X, Y))] - \varrho(\theta, (X, Y)) \tag{D.16}$$

$\mathcal{B}_{(\theta)}$ with respect to the data probability measure $P_{\mathbf{Z}}$, where (D.10)–(D.12) then yield

$$\mathrm{P}\left(\mathbf{Z} \in \mathcal{B}_{(\theta)}\right) \leq \exp\left[-n\psi_{R_\theta}^*(\epsilon)\right], \tag{D.17}$$

where $\psi^*_{R_\theta}$ is the Fenchel-Legendre dual function of the real-valued random variable: . Then, it follows that,

$$\operatorname*{ess\,sup}_{\theta\sim P_{\widehat{\theta}(\mathbf{Z})}} \mathrm{P}\left(\mathbf{Z}\in\mathcal{B}_{(\theta)}\right) \leq \exp\Big[-n \operatorname*{ess\,inf}_{\theta\sim P_{\widehat{\theta}(\mathbf{Z})}} \psi^*_{R_\theta}(\epsilon)\Big]. \tag{D.18}$$

We can now use Lemma 2, rearranging terms and taking the expectation w.r.t. $P_{\widehat{\theta}(\mathbf{Z})}$, we have that

$$\begin{aligned}
\mathrm{P}\left(\mathcal{G}_{\mathrm{G}}(\mathcal{A},\mathbf{Z})\geq\epsilon\right) &= \mathrm{P}\left((\widehat{\theta}(\mathbf{Z}),\mathbf{Z})\in\mathcal{B}\right) \\
&\leq \frac{I(\mathbf{Z};\widehat{\theta}(\mathbf{Z}))+1}{-\log(P_{\widehat{\theta}(\mathbf{Z})}\times P_{\mathbf{Z}}(\mathcal{B}))} \qquad \text{(D.19)}\\
&\leq \frac{I(\mathbf{Z};\widehat{\theta}(\mathbf{Z}))+1}{-\log\left(\int P_{\mathbf{Z}}(\mathcal{B}_{(\theta)})\,dP_{\widehat{\theta}(\mathbf{Z})}(\theta)\right)} \\
&\leq \frac{I(\mathbf{Z};\widehat{\theta}(\mathbf{Z}))+1}{-\log\left[\operatorname*{ess\,sup}_{\theta\sim P_{\widehat{\theta}(\mathbf{Z})}}\mathbf{P}\left(\mathbf{Z}\in\mathcal{B}_{(\theta)}\right)\right]} \\
&\leq \frac{I(\mathbf{Z};\widehat{\theta}(\mathbf{Z}))+1}{n\left[\operatorname*{ess\,inf}_{\theta\sim P_{\widehat{\theta}(\mathbf{Z})}}\psi^*_{R_\theta}(\epsilon)\right]}, \qquad \text{(D.20)}
\end{aligned}$$

where inequality (D.19) follows from (D.5) and (D.20) follows from (D.18).

Attribute inference. Let $\varphi^\star$ be the attack strategy given in (2.2) with $\mathcal{P}_{\mathrm{Suc}}(\varphi^\star) = \mathrm{P}\{\widehat{T}=T\}\geq 1/2$, where $\widehat{T}$ denotes the random variable $\widehat{T}\triangleq\varphi^\star(\widehat{\theta}(\mathbf{Z}),S)$. Note that $\widehat{T}$ is independent of T given $(\widehat{\theta}(\mathbf{Z}),S)$. We will show that,

$$I(T;\widehat{\theta}(\mathbf{Z})|S)\geq d_{\mathrm{KL}}\left(\mathcal{P}_{\mathrm{Suc}}(\varphi^\star)\,\middle\|\,\mathbb{E}\Big[\min_{t\in\mathcal{T}}P_{T|S}(t|S)\Big]\right), \tag{D.21}$$

where,

$$d_{\mathrm{KL}}(p\|q)\triangleq p\log_2\frac{p}{q}+(1-p)\log_2\frac{(1-p)}{(1-q)}. \tag{D.22}$$

To this end, denote by $D_{\mathrm{KL}}(\cdot\|\cdot)$ the KL-divergence between two distributions and observe that, by Lemma 1,

$$D_{\mathrm{KL}}\left(P_{T|\widehat{\theta}(\mathbf{Z})S}(\cdot|\theta,s)\|P_{T|S}(\cdot|s)\right)\geq d_{\mathrm{KL}}\left(P_{T|\widehat{\theta}(\mathbf{Z})S}(t|\theta,s)\,\|\,P_{T|S}(t|s)\right), \tag{D.23}$$

where $t=t(s,\theta)$ may be any function of $(s,\theta)\in\mathcal{S}\times\Theta$. By taking the expectation over $\theta,s\sim p_{S,\widehat{\theta}(\mathbf{Z})}$, we obtain

$$I(T;\widehat{\theta}(\mathbf{Z})|S)\geq\mathbb{E}\left[d_{\mathrm{KL}}\left(P_{T|\widehat{\theta}(\mathbf{Z})S}(t|\widehat{\theta}(\mathbf{Z}),S)\,\|\,P_{T|S}(t|S)\right)\right]. \tag{D.24}$$

We choose a mapping $t^\star_{(s,\theta)}$ that satisfies,

$$\mathbb{E}\Big[P_{T|\widehat{\theta}(\mathbf{Z})S}(t^\star|\widehat{\theta}(\mathbf{Z}),S)\Big]=\mathbb{E}\Big[\max_{t\in\mathcal{T}}P_{T|\widehat{\theta}(\mathbf{Z})S}(t|\widehat{\theta}(\mathbf{Z}),S)\Big]=\mathcal{P}_{\mathrm{Suc}}(\varphi^\star). \tag{D.25}$$

It is straightforward to verify that,

$$\mathcal{P}_{\mathrm{Suc}}(\varphi^\star) \geq \mathbb{E}\Big[\max_{t\in\mathcal{T}} P_{T|S}(t|S)\Big]. \tag{D.26}$$

Then, by convexity of the function $(p,q) \mapsto d_{\mathrm{KL}}(p\|q)$, we can continue from (D.24) to show,

$$\begin{aligned} I(T;\widehat{\theta}(\mathbf{Z})|S) &\geq \mathbb{E}\left[d_{\mathrm{KL}}\Big(P_{T|\widehat{\theta}(\mathbf{Z})S}(t^\star|\widehat{\theta}(\mathbf{Z}),S)\Big\|\, P_{T|S}(t^\star|S)\Big)\right] \\ &\geq d_{\mathrm{KL}}\left(\mathcal{P}_{\mathrm{Suc}}(\varphi^\star)\,\middle\|\,\mathbb{E}\big[P_{T|S}\big(t^\star|S\big)\big]\right) \\ &\geq d_{\mathrm{KL}}\left(\mathcal{P}_{\mathrm{Suc}}(\varphi^\star)\,\middle\|\,\mathbb{E}\big[\max_{t\in\mathcal{T}} P_{T|S}(t|S)\big]\right), \end{aligned} \tag{D.27}$$

where the last inequality (D.27) follows by using (D.26) and noticing that the function $q \mapsto d_{\mathrm{KL}}(p\|q)$ is non-increasing for $q \in [0,p]$.

Finally, notice that we can apply the bound,

$$d_{\mathrm{KL}}(p\|q) \geq \max\left\{2(p-q)^2, -p\log_2(q) - 1\right\}, \tag{D.28}$$

with $p \geq q$. □

Appendix E

Gaussian Data and Linear Regression

Recall the following notation: $\mathbf{x}$ is the $[d \times n]$ matrix given by $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, while $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ and $\mathbf{W} = (W_1, W_2, \ldots, W_n)$ are $[1 \times n]$ vectors. Let each copy of noise W be normal i.i.d.; $W \sim \mathcal{N}(\,\cdot\,; 0, \sigma^2)$. Since Y_i is linear in W_i, Y_i is also normal distributed, $Y_i \sim \mathcal{N}(\,\cdot\,; \beta^T x_i, \sigma^2)$. Since model parameters are linear in the training set $\mathbf{Y}$, their pdf is a multivariate Gaussian, $\widehat{\theta}(\mathbf{Y}) \sim Q(\,\cdot\,) \triangleq \mathcal{N}(\,\cdot\,; \beta, \sigma^2 \overline{x}^{-1})$, where $\overline{x} \triangleq \mathbf{x}\mathbf{x}^T$. Furthermore, fixing the $j-$th sample in the training set to s, we have $\widehat{\theta}(\mathbf{Y})$ distributed as $Q_j(\,\cdot\,|s) \triangleq \mathcal{N}\big(\,\cdot\,; \beta + \overline{x}^{-1} x_j (s - x_j^T \beta), \sigma^2 \overline{x}^{-1} (\mathbb{I}^{d\times d} - x_j x_j^T \overline{x}^{-1})\big)$.

Consider a MIA against this model. The attacker possesses side information (S_J, J), that is, a test sample and its corresponding index. Recall our definition $S_J = T(Y_J) + (1-T)(Y'_J)$, where J is a random index in $[n]$. When $T = 0$, $S = Y'_J$, independent of the training set; hence,

$$\begin{aligned} p_{S_J J \widehat{\theta}|T}(s, j, \theta|0) &= \frac{1}{n} p_{S_J \widehat{\theta}(\mathbf{Y})|T,J}(s, \theta|0, j) \\ &= \frac{1}{n} p_{\widehat{\theta}(\mathbf{Y})|T}(\theta|0) p_{S_J|T,J}(s|0, j) \\ &= \frac{1}{n} Q(\theta) p_{Y_j}(s)\ , \end{aligned} \tag{E.1}$$

On the other hand, when $T = 1$, $S = Y_J$ is the $J-$th component of the training set $\mathbf{Y}$; therefore,

$$\begin{aligned} p_{S_J J \widehat{\theta}|T}(s, j, \theta|1) &= \frac{1}{n} p_{S_J \widehat{\theta}(\mathbf{Y})|T,J}(s, \theta|1, j) \\ &= \frac{1}{n} p_{\widehat{\theta}(\mathbf{Y})|T J S_J}(\theta|1, j, s) p_{S_J|T,J}(s|1, j) \\ &= \frac{1}{n} Q_j(\theta|s) p_{Y_j}(s)\ . \end{aligned} \tag{E.2}$$

Note that $Q(\,\cdot\,)$ and $Q_i(\,\cdot\,|s)$ differ only by their mean and variance. The second pdf has shifted mean and reduced variance. The reduced variance is to be expected, since fixing one of the samples in the training set should reduce randomness. Note that

if the dimension of the space of features is equal to the amount of samples (i.e., $d \geq n$) an attacker having access to the feature vectors in the training set $\mathbf{x}$ can solve a system of equations to obtain $\mathbf{y}$.

In the following, we derive a theoretical lower bound for (3.2). Define $R \triangleq x_J^T(\mathbf{x}\mathbf{x}^T)^{-1}\mathbf{x}\mathbf{Y}^T - S_J$. Fixing J and T, R is a linear combination of Gaussian r.v.s, and thus R is a Gaussian random variable. Regardless of T and J, $\mathbb{E}[R] = 0$. If $T = 0$; then $S_J = Y'_J$, independent of $\mathbf{Y}$,

$$\mathrm{Var}[R|T=0] = \sigma^2 + \frac{d\sigma^2}{n} \ . \tag{E.3}$$

If $T = 1$, then $S_J = Y_J$ is the $J-$th component of $\mathbf{Y}$; consequently,

$$\mathrm{Var}[R|T=1] = \sigma^2 - \frac{d\sigma^2}{n} \ . \tag{E.4}$$

In total,

$$\sigma_R^2 \triangleq \mathrm{Var}[R] = \sigma^2 \ . \tag{E.5}$$

Since R is a Gaussian random variable, the squared error, defined by $R^2 \triangleq \left(x^T(\mathbf{x}\mathbf{x}^T)^{-1}\mathbf{x}\mathbf{Y} - S_J\right)^2$, is exponentially tail-bounded[1]; hence, we can apply Theorem 4 to get a theoretical lower bound on the success probability of the Bayesian MIA. Assume that T is Bernoulli $1/2$ distributed; thus,

$$\begin{aligned}
\mathcal{P}_{\mathrm{Suc}}(\varphi^\star) &\geq \frac{1}{2} + \frac{|\mathbb{E}[\mathcal{E}_{\mathrm{G}}(\mathcal{A}, \mathbf{Z})]|}{4R_{\max}} \\
&\quad - \exp\left(-\frac{R_{\max}}{2\sigma_R^2}\right)\left(1 + \frac{2\sigma_R^2}{R_{\max}}\right) \ . \\
&= \frac{1}{2} + \frac{d}{2n}\frac{\sigma^2}{R_{\max}} - \exp\left(-\frac{R_{\max}}{2\sigma^2}\right)\left(1 + \frac{2\sigma^2}{R_{\max}}\right) \ ,
\end{aligned} \tag{E.6}$$

where we use (3.1).

The Mutual information between a test sample S_J and the model parameters $\widehat{\theta}(\mathbf{Y})$ given the sensitive attribute T is,

[1] See proof of Theorem 4 in C.

$$
\begin{aligned}
I(S_J;\widehat{\theta}(\mathbf{Y})|T) &= \sum_{t\in\{0,1\}} I(S_J;\widehat{\theta}(\mathbf{Y})|T=t) \\
&= \mathrm{P}\{T=1\} I(S_J;\widehat{\theta}(\mathbf{Y})|T=1) \\
&= \mathrm{P}\{T=1\}\frac{1}{n}\sum_{j=1}^{n}\int Q_j(\theta|s)P_{Y_j}(s)\log\left[\frac{Q_j(\theta|s)P_{Y_j}(s)}{Q(\theta)P_{Y_j}(s)}\right]d\theta ds \\
&= \mathrm{P}\{T=1\}\frac{1}{n}\sum_{j=1}^{n}\int Q_j(\theta|s)P_{Y_j}(s)\log\left[\frac{Q_j(\theta|s)}{Q(\theta)}\right]d\theta ds \\
&= \mathrm{P}\{T=1\}\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}_{S\sim P_{Y_j}}\left[D_{\mathrm{KL}(Q_j(\theta|s)|Q(\theta))}\right] \\
&= \mathrm{P}\{T=1\}\frac{1}{2n}\sum_{j=1}^{n}\mathbb{E}_{S\sim P_{Y_j}}\Big[\mathrm{Tr}\big(\Sigma^{-1}\Sigma_j\big)-\log\left(\frac{|\Sigma_j|}{|\Sigma|}\right)-d \\
&\qquad +(\mu_j(S)-\beta)^T\Sigma^{-1}(\mu_j(S)-\beta)\Big] \\
&= \mathrm{P}\{T=1\}\frac{1}{2n}\sum_{j=1}^{n}\left(\mathrm{Tr}\big(\Sigma^{-1}\Sigma_j\big)-\log\left(\frac{|\Sigma_j|}{|\Sigma|}\right)-d+x_j^T\overline{x}^{-1}x_j\right) \\
&= \mathrm{P}\{T=1\}\frac{1}{2n}\sum_{j=1}^{n}\log\left(\frac{|\Sigma|}{|\Sigma_j|}\right)\ , \qquad \text{(E.7)}
\end{aligned}
$$

with $\mu_j(s)\triangleq\beta+\overline{x}^{-1}x_j(s-x_j^T\beta)$, $\Sigma_j\triangleq\sigma^2\overline{x}^{-1}(\mathbb{I}^{d\times d}-x_jx_j^T\overline{x}^{-1})\big)$ and, $\Sigma=\overline{x}^{-1}\sigma^2$. Using the upper bound $I(S_J;\widehat{\theta}(\mathbf{Y})|T)\geq I(T;\widehat{\theta}(\mathbf{Y})|S_J)$ in combination with (2.20), we can estimate an upper bound on the probability of success of the Bayesian attacker.

Proof of (3.1). Recall the definition of the generalization gap, substituting the MSE and the model into the definition, we obtain,

$$
\begin{aligned}
\mathbb{E}\left[\mathcal{E}_{\mathrm{G}}(\mathcal{A},\mathbf{Y})\right] &= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\ell(f_{\widehat{\theta}(\mathbf{Y})}(x_i),Y_i')-\frac{1}{n}\sum_{i=1}^{n}\ell(f_{\widehat{\theta}(\mathbf{Y})}(x_i),Y_i)\right] \\
&= \frac{1}{n}\mathbb{E}\left[\left\|\widehat{\theta}(\mathbf{Y})^T\mathbf{x}-\mathbf{Y}'\right\|^2-\left\|\widehat{\theta}(\mathbf{Y})^T\mathbf{x}-\mathbf{Y}\right\|^2\right]\ , \qquad \text{(E.8)}
\end{aligned}
$$

Let $\overline{x}\triangleq\mathbf{x}\mathbf{x}^T$, then,

$$
\begin{aligned}
\mathbb{E}\left[\left\|\widehat{\theta}(\mathbf{Y})^T\mathbf{x}-\mathbf{Y}'\right\|^2\right] &= \mathbb{E}\left[\left(\mathbf{Y}\mathbf{x}^T\overline{x}^{-1}\overline{x}\overline{x}^{-1}\mathbf{x}\mathbf{Y}^T-2\mathbf{Y}'\mathbf{x}^T\overline{x}^{-1}\mathbf{x}\mathbf{Y}^T+\|\mathbf{Y}'\|^2\right)\right] \\
&= \mathbb{E}\left[\mathbf{W}\mathbf{x}^T\overline{x}^{-1}\mathbf{x}\mathbf{W}^T-\beta^T\overline{x}\beta-2\mathbf{W}'x^T\overline{x}^{-1}\mathbf{x}\mathbf{W}^T+\|\mathbf{Y}'\|^2\right] \\
&= \mathbb{E}\left[\mathbf{W}\mathbf{x}^T\overline{x}^{-1}\mathbf{x}\mathbf{W}^T-\beta^T\overline{x}\beta+\|\mathbf{Y}'\|^2\right]\ , \qquad \text{(E.9)}
\end{aligned}
$$

Note that $\mathbb{E}\left[2\mathbf{W}'x^T\overline{x}^{-1}\mathbf{x}\mathbf{W}^T\right]=0$; since $\mathbb{E}[W]=0$ and $\mathbf{W}'$ is independent from $\mathbf{W}$.

On the other hand,

$$\begin{aligned}\mathbb{E}\left[\left\|\widehat{\theta}(\mathbf{Y})^T\mathbf{x}-\mathbf{Y}\right\|^2\right] &= \mathbb{E}\left[\left(\|\mathbf{Y}\|^2-\mathbf{Y}\mathbf{x}^T\overline{x}^{-1}\mathbf{x}\mathbf{Y}^T\right)\right]\\ &= \mathbb{E}\left[\left(\|\mathbf{Y}\|^2-\beta^T\overline{x}\beta-\mathbf{W}\mathbf{x}^T\overline{x}^{-1}\mathbf{x}\mathbf{W}^T\right)\right] \qquad \text{(E.10)}\end{aligned}$$

Note that $\mathbb{E}[\|\mathbf{Y}\|^2] = \mathbb{E}[\mathbf{Y}'^2]$, since $\mathbf{Y}$ and $\mathbf{Y}'$ are i.i.d. copies of the same random vector. Hence,

$$\mathbb{E}\left|\mathcal{E}_{\mathrm{G}}(\mathcal{A},\mathbf{Y})\right| = \frac{2}{n}\mathbb{E}\left[\mathbf{W}\mathbf{x}^T\overline{x}^{-1}\mathbf{x}\mathbf{W}^T\right] \ , \qquad \text{(E.11)}$$

Taking the trace of the remaining term in the expectation,

$$\begin{aligned}\frac{2}{n}\mathbb{E}\left[\mathbf{W}\mathbf{x}^T\overline{x}^{-1}\mathbf{x}\mathbf{W}^T\right] &= \frac{2}{n}\mathbb{E}\left[\mathrm{Tr}(\mathbf{W}\mathbf{x}^T\overline{x}^{-1}\mathbf{x}\mathbf{W}^T)\right]\\ &= \frac{2}{n}\mathbb{E}\left[\mathrm{Tr}(\mathbf{x}^T\overline{x}^{-1}\mathbf{x}\mathbf{W}^T\mathbf{W})\right]\\ &= \frac{2}{n}\mathrm{Tr}(\mathbf{x}^T\overline{x}^{-1}\mathbf{x}\,\mathbb{E}\left[\mathbf{W}^T\mathbf{W}\right])\\ &= \frac{2}{n}\mathrm{Tr}(\sigma^2\mathbf{x}^T\overline{x}^{-1}\mathbf{x})\\ &= \frac{2}{n}\mathrm{Tr}(\sigma^2\mathbb{I}^{d\times d}) = \frac{2d\sigma^2}{n}\ . \qquad \text{(E.12)}\end{aligned}$$

which gives the desired result. □

ECOLE DOCTORALE

Titre : Fuite de Données Sensibles Dans les Modèles d'Apprentissage Profond

Mots clés : Apprentissage Profond, Privacy, Attaques d'Inférence de Membres

Résumé : Il a été démontré que les modèles d'apprentissage automatique (ML) peuvent divulguer des informations sur leurs ensembles d'apprentissage. Il s'agit d'un problème critique lorsque les données d'apprentissage sont de nature sensible, par exemple dans les applications médicales où les données appartiennent à des patients.
Une approche populaire pour mesurer la fuite d'informations des modèles de ML consiste à effectuer des attaques d'inférence contre les modèles. L'objectif de cette approche est de mesurer la confidentialité du système en fonction de sa robustesse aux attaques par inférence. Ces attaques sont principalement classées en attaques d'inférence de membres (MIA) et en attaques d'inférence d'attributs (AIA). L'objectif d'une MIA est de déterminer si un échantillon ou un groupe d'échantillons fait partie de l'ensemble d'apprentissage du modèle, tandis qu'une AIA tente de déduire ou de reconstruire un échantillon à partir du modèle d'apprentissage.
Bien qu'il existe d'autres méthodes pour mesurer la confidentialité en ML, comme la confidentialité différentielle, cette thèse se concentre principalement sur les attaques par inférence.
Ce travail est divisé en trois grands chapitres. Le premier chapitre présente la motivation de notre travail, l'énoncé du problème, l'examen de l'état de l'art et définit la notation et le cadre théorique qui seront utilisés dans les chapitres suivants. Le deuxième chapitre contient nos principaux résultats théoriques et fournit une taxonomie des attaques d'inférence de membres et d'attributs. Le troisième chapitre fournit une description détaillée de nos expériences et une discussion sur les résultats.
Nos résultats théoriques concernant les attaques par inférence sont décrits comme suit: Tout d'abord, nous dérivons des limites théoriques sur le taux de réussite d'un attaquant. Ce résultat fournit une limite supérieure à la probabilité de succès d'une attaque par inférence dans le cas spécifique où l'attaquant a accès aux paramètres du modèle entraîné, et donc dans tout autre scénario où l'attaquant possède moins d'informations. Deuxièmement, nous dérivons des limites qui relient l'écart de généralisation d'un modèle ML au taux de réussite d'un attaquant contre ce modèle. Ce résultat suggère qu'un modèle ML qui se généralise mal sera susceptible de faire l'objet de MIA. Cependant, l'inverse n'est pas toujours vrai, comme nous le prouvons à l'aide d'un exemple pertinent. Troisièmement, nous dressons une liste de résultats qui relient l'information mutuelle entre le modèle entraîné et son ensemble d'entraînement à l'écart de généralisation et au taux de réussite de l'attaquant.
Nous utilisons notre cadre théorique pour décrire les stratégies de MIA existant dans la littérature et nous proposons plusieurs nouvelles stratégies. Nous explorons l'utilisation de techniques de détection de distribution et de mesures de diversité pour les MIA. Nous proposons également une technique basée sur la norme de la perturbation minimale nécessaire pour qu'un modèle modifie sa prédiction à l'aide d'une attaque adversariale. En outre, nous utilisons notre cadre pour décrire un ensemble d'AIA.
Nos résultats théoriques sont illustrés à l'aide d'un scénario fictif. La limite inférieure reliant l'écart de généralisation au taux de réussite de l'attaquant est testée et comparée à l'état de l'art des MIAs dans un scénario plus réaliste.
La majeure partie de nos expériences est consacrée à l'évaluation comparative des performances des différentes stratégies de MIA contre des modèles de classification d'images les plus récents. Nous décrivons et classons les stratégies existantes dans l'état de l'art. Nous comparons l'efficacité des nouvelles stratégies proposées dans ce travail à l'état de l'art. Nous montrons empiriquement que le fait d'avoir accès à des échantillons supplémentaires pouvant être utilisés comme données d'entraînement pour l'attaquant n'offre pas d'avantage par rapport aux stratégies qui ne nécessitent pas de données supplémentaires. Nous classons les différentes stratégies en fonction de leurs performances contre les modèles de classification d'images les plus récents. Ce résultat fournit des indications sur la manière de mesurer la robustesse d'un modèle de ML en matière de protection de la vie privée.
Enfin, nous testons l'efficacité des AIA contre un modèle entraîné à classer les chiffres manuscrits. L'ensemble de données contient l'identité des auteurs et nous l'utilisons comme information sensible à déterminer par les AIA.
Nous montrons avec rigueur mathématique et de manière empirique que les réseaux neuronaux profonds sont sensibles aux attaques d'inférence de membres, même lorsqu'ils généralisent bien. Nous montrons empiriquement que les stratégies de MIA coûteuses en ressources ne sont pas plus efficaces que les stratégies qui interrogent une seule fois le modèle ML cible. Ce résultat suggère que les informations les plus pertinentes pour déterminer l'appartenance sont contenues dans les dernières couches du modèle cible.

Title : Leakage of Sensitive Data from Deep Neural Networks

Keywords : Deep Learning, Privacy, Membership Inference Attacks

Abstract : It has been shown that Machine Learning (ML) models can leak information about their training sets. This is a critical issue in the case where the training data is of a sensitive nature, e.g., medical applications where the data belongs to patients.
A popular approach for measuring the leakage of information from ML models is to perform inference attacks against the models. The goal of this approach is to measure the privacy of the system as the robustness to inference attacks. These attacks are mainly categorized into Membership Inference Attacks (MIAs) and Attribute Inference Attacks (AIAs). The goal of a MIA is to determine if a sample or group of samples are part of the training set of the model, while an AIA tries to infer or reconstruct a sample from the trained model.
Although there exist other methods for measuring privacy in ML, such as differential privacy, the main focus of this thesis is on inference attacks.
This work is divided in three big chapters. The first chapter provides the motivation for our work, problem statement, review of the state-of-the-art and sets the notation and theoretical framework to be used in future chapters. The second chapter contains our main theoretical results and provides a taxonomy of membership and attribute inference attacks. The third chapter provides and thorough description of our experiments and a discussion on the results.
Our theoretical findings regarding inference attacks are described as follows: First, we derive theoretical bounds on the success rate of an attacker. This result provides an upper bound on the success probability of an inference attack in the specific case where the attacker has access to the model parameters of the trained model, and therefore in any other scenario where the attacker possesses less information. Second, we derive bounds that link the generalization gap of a ML model to the success rate of an attacker against this model. This result suggests that a ML that generalizes poorly will be susceptible to MIAs. However, the converse is not always true, as we prove with a pertinent example. Third, we derive a list of results that relate the mutual information between the trained model and its training set to the generalization gap and the success rate of the attacker.
We use our theoretical framework to describe the MIA strategies existing in the literature and we propose several novel strategies. We explore the use of out of distribution detection techniques and diversity measures for MIAs. We also propose a technique based on the norm of the minimum perturbation necessary to make a model change its prediction using an adversarial attack. Additionally, we use our framework to describe a set of AIAs.
Our theoretical results are illustrated in a toy scenario. The lower bound relating the generalization gap to the success rate is tested and compared to state of the art MIAs in a more realistic scenario.
The bulk of our experiments are dedicated to benchmark the performance of different MIAs strategies against state of the art image classification models. We describe and categorize the existing state of the art strategies. We compare the effectiveness of the novel strategies proposed in this work to the state of the art. We empirically show that having access to additional samples that can be used as training data for the attacker does not provide an advantage over strategies that do not require additional data. We rank different strategies based on their performance against state of the art image classification models. This result provides guidelines on how to measure the privacy robustness of a ML model.
Finally, we test the effectiveness of AIAs against a model trained to classify handwritten digits. The data set contains the identity of the writers, and we use this as the sensitive information to be determined by the AIAs.
We show with mathematical rigour and also empirically that Deep Neural Networks are susceptible to Membership Inference Attacks, even when they generalize well. Empirically, we show that resource hungry MIA strategies are not more effective than strategies that simply query the target ML model one time. This result suggests that the most relevant information to determine membership is contained in the last layers of the target model.

Institut Polytechnique de Paris
91120 Palaiseau, France