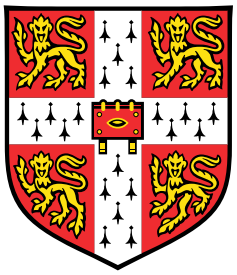
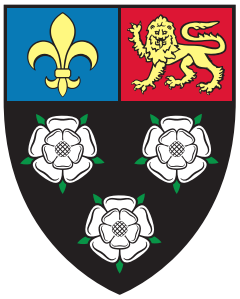


Revisiting Generalization for Deep Learning: PAC-Bayes, Flat Minima, and Generative Models



Gintare Karolina Dziugaite

Supervisor: Prof. Z. Ghahramani

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

King's College December 2018

I would like to dedicate this thesis to my children K.E. and L.A.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Gintare Karolina Dziugaite
December 2018

Acknowledgements

My first exposure to research was due to Wei Ji Ma, who accepted me for a summer internship in Computational Neuroscience despite the fact that I had only just completed my undergraduate degree in Mathematics and had no prior research experience. Working with Weiji changed my perspective on research as a career choice. I am grateful to Weiji for encouraging me to apply to PhD programs.

I would like to thank my supervisor Zoubin Ghahramani for supporting my decision to transition into machine learning research. Zoubin quickly directed me to interesting projects that resulted in publications, motivating me to continue my research career. I was very fortunate to have an advisor who was an expert in so many different areas of machine learning.

Much of the research reported in this thesis was carried out or initiated while I was a visiting student in the "Foundations of Machine Learning" program at the Simons Institute for the Theory of Computing at UC Berkeley. I would like to explicitly thank Peter Bartlett, Shai Ben-David, Dylan Foster, Matus Telgarsky, and Ruth Urner for helpful discussions. I would also like to thank Bharath Sriperumbudur for technical discussions regarding the work described in Chapter 5.

I am very grateful to my parents who were willing to help at any time and in any way they could, who visited me for months at a time, travelled with me to conferences, and helped care for my children while I was working unreasonable hours.

I am most thankful for my husband's never-ending support.

Last but not least, I would like to thank the hyper competitive deep learning research community for pushing me to exceed my rest-to-work limits while raising two small children. Under different, more relaxed, conditions, I might have gotten much less work done.

Abstract

In this work, we construct generalization bounds to understand existing learning algorithms and propose new ones. Generalization bounds relate empirical performance to future expected performance. The tightness of these bounds vary widely, and depends on the complexity of the learning task and the amount of data available, but also on how much information the bounds take into consideration. We are particularly concerned with data and algorithm-dependent bounds that are quantitatively nonvacuous. We begin with an analysis of stochastic gradient descent (SGD) in supervised learning. By formalizing the notion of flat minima using PAC-Bayes generalization bounds, we obtain nonvacuous generalization bounds for stochastic classifiers based on SGD solutions. Despite strong empirical performance in many settings, SGD rapidly overfits in others. By combining nonvacuous generalization bounds and structural risk minimization, we arrive at an algorithm that trades-off accuracy and generalization guarantees. We also study generalization in the context of unsupervised learning. We propose to use a two sample test statistic for training neural network generator models and bound the gap between the population and the empirical estimate of the statistic.

Table of contents

List of figures

List of tables

Notation

$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$	naturals, integers, rationals, reals, respectively
$\mathscr{M}_1(S)$	probability measures on the space S
$\mathscr{H}$	hypothesis class
$\ell(h,z)$	loss of a hypothesis h on a data point z
$\check{\ell}$	convex surrogate to the 0–1 loss (logistic loss)
$R_{\mathscr{D}}(h)$	risk of a hypothesis h
$\hat{R}_S(h)$	empirical risk of a hypothesis h
$I(X,Y)$	mutual information between two random variables X and Y
$\hat{\mathscr{R}}_m(S,\mathscr{H})$	empirical Rademacher complexity of $\mathscr{H}$ on a dataset S
$\mathscr{R}_m(\mathscr{H})$	Rademacher complexity of $\mathscr{H}$
$\mathrm{kl}(q\|\|p)$	KL divergence between two Bernoulli random variables with probability of success q and p
$\mathrm{KL}^{-1}(q\|c)$	defined in Eq. (2.1)
$\mathscr{N}_{w,s}$	Gaussian distribution with mean w and variance s

Introduction

Neural networks have enjoyed several waves of popularity. One of the defining properties of the most recent resurgence—the "deep learning" era—is the use of large data sets and much larger networks. Neural network approaches now dominate in fields such as vision, natural language processing, and many others. Despite this success, the generalization properties of deep learning algorithms are yet to be fully understood: there is, as yet, no complete theory explaining why common algorithms work in practice. Instead, guidelines for choosing and tuning common learning algorithm are based on empirical experience. Such practice is problematic. In some applications, standard algorithms like stochastic gradient descent (SGD) reliably return solutions with low test error. In other applications, these same algorithms rapidly overfit.

Our goal is to improve our understanding of generalization of neural networks in the deep learning regime, empowering us to:

1. explain when a learning algorithm can be expected to work well; and

2. design improved algorithms with provable generalization guarantees.

Any attempt to explain generalization must grapple with the fact that the hypothesis class induced by standard neural network models is huge, as measured by standard complexity measures, such as VC dimension and Rademacher complexity. One of the defining properties of deep learning is that models are chosen to have many more parameters than available training data. In practice, the number of available training instances is too small in comparison to the size of the network to yield generalization guarantees without taking into consideration the learning algorithm or depending on the complexity of the learned hypothesis.

This fact has long been appreciated and Bartlett (1998) is essentially credited with solving this problem in 1998. In his seminal work, Bartlett introduced fat shattering risk bounds where the fat shattering dimension of the network is controlled by the norms of the weights. One of the key contributions of this thesis is revisiting this and later developments and questioning whether they solve the problem at hand of understanding generalization in deep learning. One might hope that the generalization properties of SGD could be explained by

showing that SGD performs implicit regularization of the weight norms. However, Bartlett's learning bounds are numerically vacuous, i.e., greater than the upper bound on the loss, when applied to modern networks learned by SGD. Logically, in order to explain generalization, we need nonvacuous bounds.

Nonetheless, many authors are actively working on understanding implicit regularization and several empirical studies suggest that the implicit regularization idea may be promising (Neyshabur, Tomioka, and Srebro, 2014; Neyshabur, 2017; Zhang et al., 2017; Shwartz-Ziv and Tishby, 2017; Bartlett, Foster, and Telgarsky, 2017a; Gunasekar et al., 2017). For example, Zhang et al. (2017) demonstrate that, without regularization, SGD can achieve zero training error on standard benchmark datasets, like MNIST and CIFAR. At the same time, SGD obtains weights with very small generalization error with the original labels. As a simplified model, Zhang et al. study SGD in a linear model, and show that SGD obtains the minimum norm solution, and thus performs implicit regularization. They suggest a similar phenomenon may occur when using SGD to training neural networks. Indeed, earlier work by Neyshabur, Tomioka, and Srebro (2014) observes similar phenomena and argues for the same point: implicit regularization underlies the ability of SGD to generalize, even under massive overparametrization. Subsequent work by Neyshabur, Tomioka, and Srebro (2015) introduces "path norms" as a better measure of the complexity of ReLU networks. The authors bound the Rademacher complexity of classes of neural networks with bounded path norm. Despite progress, we show that these new bounds probably will not lead to nonvacuous generalization bounds.

The generalization question may also be addressed by studying properties of learning algorithms, such as algorithmic stability. However, existing stability guarantees typically diminish rapidly with training time. The experiments by Neyshabur, Tomioka, and Srebro (2014) demonstrate that the classification error, as estimated on the test set, does not get worse if we run SGD to convergence. On the other hand, Zhang et al. shows that SGD rapidly overfits when handed randomized labels. This example highlights the impediment of using label-independent stability analyses to explaining generalization, as noted by Zhang et al. (2017).

In summary, the theoretical relevance of many learning bounds has been assessed by looking whether the bound captures some implicit regularization, structural properties of the solution, and/or properties of the data distribution. Progress was made by eliminating or improving the dependencies on the number of layers, depth, and parameters in the neural network. Empirical relevance was commonly illustrated via plots that show the generalization bounds tracking the estimated generalization error.

In our work, we take one step further by seeking to obtain generalization bounds that are numerically close to the estimate of generalization on held-out data. Our approach combines Probably Approximately Correct (PAC) Bayes (McAllester, 1999a) framework with nonconvex optimization and other computational techniques. This direction turns out to be fruitful for several reasons. For instance, it yields insight into what empirical risk surface structure found by SGD leads to good generalization and it also allows us to interpret other existing optimizers as risk bound minimizers, analyze and control their generalization guarantees in theory and in practice.

Publications and Collaborations

This thesis brings together a number of results obtained in collaboration with my supervisor Zoubin Ghahramani and Daniel M. Roy. In particular, the main contributions of this thesis appeared in the following publications and preprints:

- Gintare Karolina Dziugaite and Daniel M. Roy (2018a). "Data-dependent PAC-Bayes priors via differential privacy". *Advances in Neural Information Processing Systems (NIPS)*. vol. 29. Cambridge, MA: MIT Press. arXiv: 1802.09583 (Chapter 3).

- Gintare Karolina Dziugaite and Daniel M. Roy (2018b). "Entropy-SGD optimizes the prior of a PAC-Bayes bound: Generalization properties of Entropy-SGD and data-dependent priors". *Proceedings of the 35th International Conference on Machine Learning (ICML)*. arXiv: 1712.09376 (Chapter 4).

- Gintare Karolina Dziugaite and Daniel M. Roy (2017). "Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data". *Proceedings of the 33rd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. arXiv: 1703.11008 (Chapter 2).

- Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani (2015). "Training Generative Neural Networks via Maximum Mean Discrepancy Optimization". *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*. UAI'15. Amsterdam, Netherlands: AUAI Press, pp. 258–267 (Chapter 5).

The core ideas were developed during discussions with my collaborators Zoubin Ghahramani and Daniel M. Roy, and, thus, these contributions are shared among the authors of each of the papers listed above. I carried out all the empirical and theoretical work, with one exception: the results in Section 3.4.1 were derived jointly in collaboration with Daniel M. Roy.

Outline and Contributions

In addition to describing the results appearing in the articles listed above, this thesis contains a more indepth literature review as well as an overview of relevant aspects of statistical learning theory, one of the main tools used in my research.

We start Chapter 1 by formally introducing tools from statistical learning theory that are used for our research. This background material should facilitate the reader in fully understanding the analyses reported in this thesis.

In Chapter 2, we then address the question of nonvacuous generalization bounds for SGD trained networks. We return to an idea by Langford and Caruana (2001), who used PAC-Bayes bounds to compute nonvacuous numerical bounds on generalization error for *stochastic* two-layer two-hidden-unit neural networks via a sensitivity analysis. By optimizing the PAC-Bayes bound directly, we are able to extend their approach and obtain nonvacuous generalization bounds for deep stochastic neural network classifiers with millions of parameters trained on only tens of thousands of examples. We connect our findings to recent and old work on flat minima and MDL-based explanations of generalization.

The PAC-Bayes framework can incorporate knowledge about the learning algorithm and data distribution through the use of distribution-dependent priors, yielding tighter generalization bounds on data-dependent posteriors. Using this flexibility, however, is difficult, especially when the data distribution is presumed to be unknown. In Chapter 3 we show how an ε-differentially private choice of the prior yields a valid PAC-Bayes bound, and then show how non-private mechanisms for choosing priors obtain the same generalization bound provided they converge weakly to the private mechanism. As a consequence, we show that a Gaussian prior mean chosen via stochastic gradient Langevin dynamics (SGLD; Welling and Teh, 2011) leads to a valid PAC-Bayes bound, despite SGLD only converging weakly to an ε-differentially private mechanism. As the bounds are data-dependent, we use empirical results on standard neural network benchmarks to illustrate the gains of data-dependent priors over existing distribution-dependent PAC-Bayes bound.

In Chapter 4, we show that Entropy-SGD (Chaudhari et al., 2017), when viewed as a learning algorithm, optimizes a PAC-Bayes bound on the risk of a Gibbs (posterior) classifier, i.e., a randomized classifier obtained by a risk-sensitive perturbation of the weights of a learned classifier. Entropy-SGD works by optimizing the bound's prior, violating the hypothesis of the PAC-Bayes theorem that the prior is chosen independently of the data. Indeed, available implementations of Entropy-SGD rapidly obtain zero training error on random labels and the same holds of the Gibbs posterior. In order to obtain a valid generalization bound, we rely on our result showing that data-dependent priors obtained by SGLD yield valid PAC-Bayes bounds. We observe that test error on MNIST and CIFAR10

falls within the (empirically nonvacuous) risk bounds computed under the assumption that SGLD reaches stationarity. In particular, Entropy-SGLD can be configured to yield relatively tight generalization bounds and still fit real labels, although these same settings do not obtain state-of-the-art performance.

In Chapter 5, we consider training a deep neural network to generate samples from an unknown distribution given i.i.d. data. We frame learning as an optimization minimizing a two-sample test statistic—informally speaking, a good generator network produces samples that cause a two-sample test to fail to reject the null hypothesis. As our two-sample test statistic, we use an unbiased estimate of the maximum mean discrepancy, which is the centerpiece of the nonparametric kernel two-sample test proposed by Gretton et al. (Gretton et al., 2012). We compare to the *generative adversarial nets* framework introduced by Goodfellow et al. (Goodfellow et al., 2014), in which learning is a two-player game between a generator network and an adversarial discriminator network, both trained to outwit the other. From this perspective, the MMD statistic plays the role of the discriminator. In addition to empirical comparisons, we prove bounds on the generalization error incurred by optimizing the empirical MMD.

Chapter 1

Statistical Learning Theory

In this chapter we introduce basic concepts from statistical learning theory. Informally, learning uses data to improve performance on one or more tasks. Statistical learning theory provides a formal framework for defining learning problems, measuring their complexity, and anticipating the performance of learning algorithms.

In this section, our focus will be on the *batch supervised learning* setting, and, in particular, classification. In supervised learning, we want to learn a mapping from a space $\mathbb{R}^k$ of inputs to a space K of labels. The goal is to find the best predictor, or hypothesis, h, in some hypothesis class $\mathscr{H} \subseteq \mathbb{R}^k \to K$, on the basis of example input–output pairs. In the batch setting, these examples are presented all at once to the learner and are assumed to be independent and identically distributed (i.i.d.).

In the batch setting, the i.i.d. assumption allows us to link (past) empirical performance to (future/expected) performance. Empirical performance is thus a key measure of the quality of a proposed hypothesis. However, empirical performance can be misleading, depending on the quantity of data and the learning algorithm. Generalization bounds relate empirical performance to performance on unseen data, and can give us confidence using a learning algorithm. These bounds may depend on a number of factors, including:

1. the hypothesis class the learner is considering (Section 1.1.3);
2. the underlying data distribution (Section 1.1.3);
3. the properties of the chosen predictor and/or prior knowledge expressed by the learner (Section 1.2);
4. the properties of the algorithm used by the learner (Sections 1.3 and 1.4), such as its stability, the property that the distribution of the algorithm's output does not change when small modifications are made to the training data.

Bounds that depend only on (1) exist, but they capture only worst-case scenarios. Taking the data distribution into considerations often yields tighter bounds. Bounds combining all factors listed above are specific to the particular problem being studied and yield the tightest known bounds.

1.1 Supervised learning

Let Z be a measurable space, and let $\mathscr{D}$ be an unknown distribution on Z. Consider the batch supervised learning setting under a loss function bounded below: having observed $S \sim \mathscr{D}^m$, i.e., m independent and identically distributed samples from $\mathscr{D}$, we aim to choose a predictor belonging to some hypothesis class $\mathscr{H}$ and parameterized by a weight vector $\mathbf{w} \in \mathbb{R}^p$, with minimal *risk*

$$R_{\mathscr{D}}(\mathbf{w}) := \mathop{\mathbb{E}}_{z\sim\mathscr{D}}(\ell(\mathbf{w}, z)), \tag{1.1}$$

where $\ell : \mathbb{R}^p \times Z \to \mathbb{R}$ is measurable and bounded below. (We ignore the possibility of constraints on the weight vector for simplicity.)

We will also consider randomized predictors. A randomized predictor is represented by an element Q in the space $\mathscr{M}_1(\mathbb{R}^p)$ of probability measure on $\mathbb{R}^p$. The risk of randomized predictors is defined via averaging:

$$R_{\mathscr{D}}(Q) := \mathop{\mathbb{E}}_{\mathbf{w}\sim Q}(R_{\mathscr{D}}(\mathbf{w})) = \mathop{\mathbb{E}}_{z\sim\mathscr{D}}\Big(\mathop{\mathbb{E}}_{\mathbf{w}\sim Q}(\ell(\mathbf{w}, z))\Big), \tag{1.2}$$

where the second equality follows from Fubini's theorem and the fact that ℓ is bounded below.

Let $S_m = (z_1, \ldots, z_m)$ denote a training set of size m. We will often drop the subscript m and simply write S, unless the training set size is not obvious. Let $\hat{\mathscr{D}} := \frac{1}{m}\sum_{i=1}^m \delta_{z_i}$ be the empirical distribution. Given a weight distribution Q, such as that chosen by a learning algorithm on the basis of data S, its *empirical risk*,

$$\hat{R}_S(Q) := R_{\hat{\mathscr{D}}}(Q) = \frac{1}{m}\sum_{i=1}^m \mathop{\mathbb{E}}_{\mathbf{w}\sim Q}(\ell(\mathbf{w}, z_i)), \tag{1.3}$$

will be studied as a stand-in for its risk, which we cannot compute. While $\hat{R}_S(Q)$ is easily seen to be an unbiased estimate of $R_{\mathscr{D}}(Q)$ when Q is independent of S, our goal is to characterize the (one-sided) *generalization error* $R_{\mathscr{D}}(Q) - \hat{R}_S(Q)$ when Q is random and dependent on S.

Note, that throughout this thesis, we will often use a term *generalization bounds*, by which we refer to a bound on the difference between the risk and empirical risk.

1.1.1 Loss functions

One of our focuses will be on classification, where $Z = X \times K$, with K a finite set of classes/labels. A product measurable function $h : \mathbb{R}^p \times X \to K$ maps weight vectors $\mathbf{w}$ to classifiers $h(\mathbf{w}, \cdot) : X \to K$. The loss function is given by $\ell(\mathbf{w}, (x,y)) = g(h(\mathbf{w},x),y)$ for some $g : K \times K \to \mathbb{R}$. In this setting, 0–1 loss corresponds to $g(y',y) = 1$ if and only if $y' \neq y$, and otherwise $g(y',y) = 0$. In binary classification, we take $K = \{-1,1\}$. Thus, under 0–1 loss $\ell : \mathbb{R}^p \times (X,K) \to \{0,1\}$ satisfies

$$\ell(\mathbf{w}, (x,y)) = \mathbb{I}(\text{sign}(h(\mathbf{w},x)) \neq y).$$

In Chapter 2, for $K = \{\pm 1\}$, we also make use of the logistic loss $\breve{\ell} : \mathbb{R}^p \times (X, \{\pm 1\}) \to \mathbb{R}_+$

$$\breve{\ell}(\mathbf{w}, (x,y)) = \frac{1}{\log(2)} \log\big(1 + \exp(-h(\mathbf{w},x)y)\big),$$

which serves as a convex surrogate (i.e., upper bound) to the 0–1 loss.

We also consider parametric families of probability-density-valued classifiers $h : \mathbb{R}^p \times X \to [0,1]^K$. For every input $x \in X$, the output $h(\mathbf{w},x)$ determines a probability distribution on K. In this setting, $\ell(\mathbf{w}, (x,y)) = g(h(\mathbf{w},x),y)$ for some $g : [0,1]^K \times K \to \mathbb{R}$. The standard loss is then the cross entropy, given by $g((p_1,\ldots,p_K),y) = -\log p_y$. (Under cross entropy loss, the empirical risk is, up to a multiplicative constant, a negative log likelihood.) In the special case of binary classification, the output can be represented simply by an element of $[0,1]$, i.e., the probability the label is one. The binary cross entropy, ℓ_{BCE}, is given by $g(p,y) = -y\log(p) - (1-y)\log(1-p)$. Note that cross entropy loss is merely bounded below. We consider bounded modifications in Section 4.7.2.

In the context of supervised learning, we will sometimes refer to elements of $\mathbb{R}^p$ and $\mathscr{M}_1(\mathbb{R}^p)$ as classifiers and randomized classifiers, respectively. Likewise, under 0–1 loss, we will sometimes refer to (empirical) risk as (empirical) error.

In summary, we define the following notions of risk that we use throughout:

- $\hat{R}_S(\mathbf{w}) = \dfrac{1}{m}\sum_{i=1}^{m} \ell(\mathbf{w}, z_i)$ empirical risk of (hypothesis indexed by parameters) $\mathbf{w}$ on training data S (under 0–1 loss, training error);
- $\breve{R}_S(\mathbf{w}) = \dfrac{1}{m}\sum_{i=1}^{m} \breve{\ell}(\mathbf{w}, z_i)$ empirical risk under the surrogate loss (or surrogate error) of $\mathbf{w}$ on training data S. We use this empirical risk for training purposes in Chapter 2 when we need our objective to be differentiable;

- $R_{\mathscr{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{S\sim\mathscr{D}^m}[\hat{R}_{\mathbf{w}}(S)]$ risk (error) under (the data distribution) $\mathscr{D}$ for $\mathbf{w}$ (in Chapter 2, we often drop the subscript $\mathscr{D}$ and just write $R(\mathbf{w})$);
- $\hat{R}_S(Q) = \mathop{\mathbb{E}}_{\mathbf{w}\sim Q}[\hat{R}_S(\mathbf{w}))]$ empirical risk (error) of (randomized classifier) Q on training data S;
- $R(Q)) = \mathop{\mathbb{E}}_{\mathbf{w}\sim Q}[R_{\mathscr{D}}(\mathbf{w})]$ risk (error) of Q under $\mathscr{D}$

1.1.2 Empirical risk minimization and PAC-learning

In the rest of this section, we introduce basic principles from statistical learning theory. This development mirrors the development in the book of Shalev-Shwartz and Ben-David (2014).

Empirical risk minimization (ERM) is an idealized learning algorithm that chooses the parameters $\mathbf{w}^*$ whose risk matches the best possible risk

$$\inf_{\mathbf{w}\in\mathscr{H}} \hat{R}_S(\mathbf{w}). \tag{1.4}$$

The properties of ERM are well studied. A large hypothesis class $\mathscr{H}$ may contain a predictor with minimal empirical risk but large true risk, leading to a high generalization error, i.e., *overfitting*. Choosing a more restrictive hypothesis class may prevent overfitting in exchange for bias. Sharp results exist relating the possibility of obtaining uniform bounds on the generalization error to properties of the hypothesis class $\mathscr{H}$ and the 0–1 loss function ℓ.

Definition 1.1.1 (Agnostic PAC Learnable)**.** Let an algorithm map a training sample to a hypothesis, i.e., $\mathscr{A} : Z^m \to \mathscr{H}$. A hypothesis class $\mathscr{H}$ is called *(agnostic) PAC learnable* if there exists an algorithm $\mathscr{A}$ and a function $m_{\mathscr{H}}(\varepsilon,\delta) : (0,1)^2 \to \mathbb{N}$, such that for every $(\varepsilon,\delta) \in (0,1)^2$ and a data distribution $\mathscr{D}$, for all $m > m_{\mathscr{H}}(\varepsilon,\delta)$, with probability at least $1-\delta$ over the training sample S,

$$R_{\mathscr{D}}(\mathscr{A}(S)) \leq \min_{\mathbf{w}\in\mathscr{H}} R_{\mathscr{D}}(\mathbf{w}) + \varepsilon. \tag{1.5}$$

We call $m_{\mathscr{H}}(\varepsilon,\delta)$ a *sample complexity* for $\mathscr{H}$, i.e., it is the number of examples needed to guarantee PAC learnability of $\mathscr{H}$.

In other words, if a PAC learning algorithm is given enough data, for any data distribution it is guaranteed with high probability to return a predictor that is nearly as good as the best predictor in the hypothesis class. It is also known that a $\mathscr{H}$ is PAC learnable if and only if it is learnable by ERM.

Uniform convergence

There are many ways to understand PAC learnability, one of the key ways is through uniform convergence. This property guarantees that our generalization error converges to zero with the sample size, and quantifies the rate of that convergence.

Definition 1.1.2 (Uniform Convergence)**.** A hypothesis class $\mathscr{H}$ has the *uniform convergence* (UC) property if there exists a function $m^{\mathrm{UC}}_{\mathscr{H}}(\varepsilon,\delta) : (0,1)^2 \to \mathbb{N}$, such that for every $(\varepsilon,\delta) \in (0,1)^2$ and data distribution $\mathscr{D}$, for all $m \geq m^{\mathrm{UC}}_{\mathscr{H}}(\varepsilon,\delta)$, with probability at least $1-\delta$ over the training sample S,

$$\left| R_{\mathscr{D}}(\mathbf{w}) - \hat{R}_S(\mathbf{w}) \right| \leq \varepsilon. \tag{1.6}$$

We define the sample complexity of $\mathscr{H}$ to be the smallest function m satisfying the definition of $m^{\mathrm{UC}}_{\mathscr{H}}$.

A fundamental result in statistical learning is that a class is PAC learnable if and only if it has the UC property.

1.1.3 Bounding the sample complexity: VC dimension and Rademacher complexity

Hypothesis classes with the UC property vary in terms of their sample complexity. Here we introduce two ways to measure the complexity of the hypothesis class that allows us to bound the sample complexity.

VC dimension

One measure of complexity of the hypothesis class is expressed in terms of the number of ways it can label the dataset.

Definition 1.1.3 (Shattering)**.** Let $\mathscr{H}$ be a class of binary predictors parametrized by $\mathbf{w}$, i.e., each $\mathbf{w} \in \mathscr{H}$ maps X to $K = \{0,1\}$. Let $X_m = \{x_1,...,x_m\} \in X^m$. Then $\mathscr{H}$ is said to shatter X_m if

$$\{(h(\mathbf{w},x_1),..,h(\mathbf{w},x_n)) : \mathbf{w} \in \mathscr{H}\} \supseteq \{0,1\}^m \tag{1.7}$$

Definition 1.1.4 (VC dimension)**.** The VC dimension of a binary $\mathscr{H}$, denoted by $\mathrm{VCdim}_{\mathscr{H}}$, is the maximum size of a set that it can shatter.

Thus if $\mathscr{H}$ has VC dimension d, we know that there is a hypothesis $\mathbf{w}$ in the class that achieves zero empirical risk for every labelling of some dataset of size d. A key result in statistical learning is that $\mathscr{H}$ has uniform convergence property if and only if it has a finite VC dimension.

The VC dimension can be used to bound the sample complexity, and thus the generalization error. However, VC bounds on the generalization error are loose in practice as they account for worst case data distributions. These bounds are valid for any data distribution $\mathscr{D}$ and any ERM learning algorithm, and thus it needs to give a bound that holds for an adversarially crafted hypothesis for the data distribution.

One can bound the VC dimension of neural network hypothesis space. The first bounds in the literature are reported in (Goldberg and Jerrum, 1995; Koiran and Sontag, 1996) and depend on the number of parameters in the network. Bartlett, Maiorov, and Meir (1999) improved these bounds by introducing the dependence on the number of layers. These bounds are of order $O(LW\log(W))$, where W is the number of parameters in the neural networks and L is the number of layers. In Chapter 2 we evaluate one of such bounds and show that for large neural networks used in practice, we get a very large upper bound on the VC dimension (more than 10^7). The large VC dimension combined with a relatively small number of training examples in the dataset results in generalization bounds that are orders of magnitude too big to make a nonvacuous guarantee on the risk.

Rademacher Complexity

The VC bounds discussed above ignore the distribution over the data. Thus one can possibly get tighter bounds by incorporating the data distribution. In this section, we describe the application of *Rademacher complexity* to statistical learning theory (Koltchinskii and Panchenko, 2002; Bartlett and Mendelson, 2003).

Definition 1.1.5 (Rademacher Complexity)**.** Fix a sample size $m \in \mathbb{N}$. Let $\sigma = \{\sigma_i\}_{i\in\mathbb{N}}$ be a sequence of *Rademacher random variables*, i.e., random variables taking values in $\{-1,1\}$ with equal probability. Fix a class of measurable functions $\mathscr{F} \subseteq Z \to \mathbb{R}$. For a sample $S \in Z^m$, the empirical Rademacher complexity of the class $\mathscr{F}$ is

$$\hat{\mathscr{R}}_m(S,\mathscr{F}) = \mathbb{E}_\sigma[\sup_{f\in\mathscr{F}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i f_i(z_i)]. \tag{1.8}$$

The Rademacher complexity (under a data distribution $\mathscr{D}$) of the class $\mathscr{F}$ is

$$\mathscr{R}_m(\mathscr{F}) = \mathbb{E}_{S\sim\mathscr{D}}\hat{\mathscr{R}}_m(S,\mathscr{F}). \tag{1.9}$$

Usually, one studies the Rademacher complexity of the loss class, and thus $\mathscr{F}$ refers to the loss function composed with $\mathscr{H}$, i.e., all $f \in \mathscr{F}$ are such that $f(x,y) = \ell(\mathbf{w},(x,y))$ for some $\mathbf{w} \in \mathscr{H}$. In this case, the Rademacher complexity can be used to obtain the mean of worst case generalization error:

Theorem 1.1.6.

$$\mathop{\mathbb{E}}_{S\sim\mathscr{D}}\Big[\sup_{w\in\mathbb{R}^p}\big|R_{\mathscr{D}}(\mathbf{w})-\hat{R}_S(\mathbf{w})\big|\Big] \leq 2\mathscr{R}_m(\{\ell(\mathbf{w},\cdot):\mathbf{w}\in\mathbb{R}^p\}) \tag{1.10}$$

This bound can be used to prove a bound on the expected generalization error and expected oracle risk (the gap between risks of an ERM classifier and the best classifier in the hypothesis class) of any ERM mechanism.

One of the key application of Rademacher complexity in statistical learning theory is to margin bounds for classifiers that output a real-valued prediction. The 0-1 loss of these predictors is obtained by thresholding the output. To obtain a margin bound, one studies the ramp loss, a Lipschitz upper bound to the 0–1 loss. One can show that the Rademacher of the loss class is no larger than the product of the Lipschitz constant and the Rademacher complexity of the real-valued hypothesis class. In this case, the $\mathscr{H}$ has a high Rademacher complexity if it is rich enough to contain predictors that can explain a random labelling of the data.

In Chapter 2 we present and evaluate a generalization error bound for neural network classifiers. Also, in Chapter 5 (Section 5.4), we introduce a neural network generative model and use the Rademacher complexity to obtain some generalization guarantees.

1.1.4 Structural Risk Minimization

The method of ERM finds a hypothesis minimizing the empirical risk. However, for very complex hypothesis classes, ERM may lead to overfitting: low empirical risk does not guarantee low risk. One solution is to introduce more structure in the hypothesis class that reflects the varying complexity of the hypothesis within the original hypothesis class. This is exactly the motivation behind *Structural Risk Minimization*, which was introduced by Vapnik and Chervonenkis (1974).

Fix a hypothesis class $\mathscr{H}$, let S be a dataset, and consider a bound on the risk of the form

$$\alpha\hat{R}_S(\mathbf{w}) + C(\mathbf{w}), \tag{1.11}$$

where $\alpha \in \mathbb{R}_+$ and the term $C(\mathbf{w})$ may depend on the hypothesis $\mathbf{w}$. In ERM, the term $C(\mathbf{w})$ is a constant, i.e., it does not depend on the particular hypothesis $\mathbf{w}$. In classical SRM, the

term $C(\mathbf{w})$ is not constant, and so Eq. (1.11) encodes a trade-off between the empirical risk and some notion of complexity for predictors.

Consider a hypothesis class $\mathscr{H} = \bigcup_{n\in\mathbb{N}} \mathscr{H}_n$ where, for each $n \in \mathbb{N}$, the subclass $\mathscr{H}_n$ has the uniform convergence property with sample complexity $m^{\mathrm{UC}}_{\mathscr{H}_n}(\varepsilon,\delta)$. Define the function $\varepsilon_n : \mathscr{M} \times \mathbb{N} \times (0,1) \to (0,1)$ as

$$\varepsilon_n(m,\delta) = \min\{\varepsilon \in (0,1) : m^{\mathrm{UC}}_{\mathscr{H}_n}(\varepsilon,\delta) \leq m\}. \tag{1.12}$$

Define a weight function $w : \mathbb{N} \to [0,1]$ over each hypothesis class $\{\mathscr{H}_n\}_{n\in\mathbb{N}}$ in $\mathscr{H}$, such that $\sum_{n\in\mathbb{N}} w(n) \leq 1$. The following generalization guarantee holds:

Theorem 1.1.7. *(Shalev-Shwartz and Ben-David, 2014) Let $\mathscr{H}$, w, and ε_n be defined as above. Then for all $\delta \in (0,1)$ and $\mathscr{D}$, for all $\mathbf{w} \in \mathscr{H}$, with probability at least $1-\delta$ over the choice of S,*

$$R_{\mathscr{D}}(\mathbf{w}) \leq \hat{R}_S(\mathbf{w}) + \min_{n:\,\mathbf{w}\in\mathscr{H}_n} \varepsilon_n(m, \delta \cdot w(n)). \tag{1.13}$$

Note that, the weight function, once normalized, has the same form as a prior distribution in Bayesian analysis. However, the guarantee holds for any weight function, provided it is fixed before seeing the data. The theorem makes it clear that we are incentivized to assign large weight to small subclasses $\mathscr{H}_n$ that we believe will fit the data well, and so there is an indirect link with the Bayesian prior.

The risk bound, Eq. (1.13), suggests a natural learning algorithm for any hypothesis space that can be written as a countable union of PAC learnable classes. The following paradigm is called Structural Risk Minimization (SRM):

1. Write $\mathscr{H} = \bigcup_{n\in\mathbb{N}} \mathscr{H}_n$ for some countable collection of PAC learnable subclasses;
2. Assign weight $w(n)$ to each $\mathscr{H}_n$, with $\sum_n w(n) \leq 1$;
3. Given data S and a confidence parameter $\delta \in (0,1)$, choose $\mathbf{w} \in \mathscr{H}$ that minimizes the right hand side of Eq. (1.13).

It is common to use a VC bound on $\varepsilon_{n(\mathbf{w})}(m, \delta w(n(\mathbf{w})))$.

Nonuniform learnability is a weaker notion than PAC learnability. The latter implies the former. In PAC learnability one is required to choose a single value for sample complexity $m_{\mathscr{H}}(\varepsilon,\delta)$ that works uniformly for all hypothesis in $\mathscr{H}$ (including the hypothesis that has the lowest true error). Nonuniform learnability relaxes this requirement by allowing to set m depending on $\mathscr{H}_n$.

Note, that while each hypothesis $\mathscr{H}_n$ in SRM is uniformly learnable, $\mathscr{H}$ is only nonuniformly learnable.

The SRM paradigm drives the learning towards the simplest hypothesis that achieves a relatively low empirical risk. This closely connects to the Minimum Description Length (MDL) and PAC-Bayes frameworks discussed in the following sections.

1.1.5 Minimum Description Length

The Minimum Description Length (MDL) principle can be seen as a special case of SRM. In MDL:

- each $\mathscr{H}_n$ is a singleton class. ByHoeffding's inequality, $\varepsilon_n(m,\delta)$ decays as $1/\sqrt{m}$;
- the weights $w(n)$ of each $\mathscr{H}_n$ are assigned based on an information-theoretic notion of complexity, namely the description length of a hypothesis.

In MDL, the learning task is viewed as a compression task.

See Grünwald (2005) for a tutorial and further references on MDL. The article also discusses how MDL connects to Bayes factor model selection and Bayesian inference. In Chapter 2 we mention how MDL connects to PAC-Bayes.

1.2 PAC-Bayes

PAC-Bayes theory was first developed by McAllester in 1999 with the goal to provide PAC learning guarantees for Bayesian algorithms. The first PAC analysis of Bayesian algorithms is due to Shawe-Taylor and Williamson. The developed theory reported in (Shawe-Taylor and Williamson, 1997) apply under more restrictive set of assumptions on the parameter space and prior measures, as compared to McAllester's work.

PAC-Bayes theorems give generalization and oracle bounds that can be used to build and analyze learning algorithms that are similar to SRM and MDL. In all three settings, the learner:

- fixes a prior/weight function over the hypothesis class, prior to seeing the data;
- optimizes a bound on the risk of the form Eq. (1.11).

The prior can also be interpreted as a data-independent randomized classifer.

Unlike in SRM or MDL, PAC-Bayes bounds are generally used to control the risk of data-dependent *randomized* classifiers. PAC-Bayes bounds lead to a natural variant of SRM where:

- as in MDL, the hypothesis class $\mathscr{H} = \bigcup_{h \in \mathscr{H}} \mathscr{H}_h$ is viewed as a (possibly uncountable) union of singleton classes $\mathscr{H}_h = \{\mathbf{w}\}$;
- the weight function in MDL is replaced by a (possibly atomless) probability measure P over $\mathscr{H}$;
- the search over classifiers, h, is replaced by one over *randomized* classifiers, Q, i.e., probability measures over $\mathscr{H}$

MDL is obtained as the special case where $\mathscr{H}$ is countable and the search is restricted to degenerate probability measures (i.e., ones assigning probability one to a single hypothesis). Note that P and Q are the same type of structure.

1.2.1 KL divergence and the PAC-Bayes theorem

Let Q, P be probability measures defined on $\mathbb{R}^p$, assume Q is absolutely continuous with respect to P, and write $\frac{\mathrm{d}Q}{\mathrm{d}P} : \mathbb{R}^p \to \mathbb{R}_+ \cup \{\infty\}$ for some Radon–Nikodym derivative of Q with respect to P. Then the Kullback–Liebler divergence (or relative entropy) of P from Q is defined to be

$$\mathrm{KL}(Q||P) := \int \log \frac{\mathrm{d}Q}{\mathrm{d}P} \mathrm{d}Q. \tag{1.14}$$

We are mostly concerned with KL divergences where Q and P are probability measures on Euclidean space, $\mathbb{R}^d$, absolutely continuous with respect to Lebesgue measure. Let q and p denote the respective densities. In this case, the definition of the KL divergence simplifies to

$$\mathrm{KL}(Q||P) = \int \log \frac{q(x)}{p(x)} q(x) \mathrm{d}x.$$

Of particular interest to us is the KL divergence between multivariate normal distributions in $\mathbb{R}^d$. Let $N_q = \mathscr{N}(\mu_q, \Sigma_q)$ be a multivariate normal with mean μ_q and covariance matrix Σ_q, let $N_p = \mathscr{N}(\mu_p, \Sigma_p)$, and assume Σ_q and Σ_p are positive definite. Then $\mathrm{KL}(N_q||N_p)$ is

$$\begin{aligned} \frac{1}{2}\Bigg(& \mathrm{tr}\left(\Sigma_p^{-1}\Sigma_q\right) - k + \left(\mu_p - \mu_q\right)^\top \Sigma_p^{-1}(\mu_p - \mu_q) \\ & + \ln\left(\frac{\det \Sigma_p}{\det \Sigma_q}\right)\Bigg). \end{aligned} \tag{1.15}$$

For $p, q \in [0,1]$, we abuse notation and define

$$\begin{aligned}\mathrm{kl}(q||p) &:= \mathrm{KL}(\mathscr{B}(q)||\mathscr{B}(p)) \\ &= q\log\frac{q}{p} + (1-q)\log\frac{1-q}{1-p},\end{aligned}$$

where $\mathscr{B}(p)$ denotes the Bernoulli distribution on $\{0,1\}$ with mean p.

1.2.2 Bounds

We now present a PAC-Bayes theorem, first established by McAllester (1999a). We focus on the setting of bounding the generalization error of a (randomized) classifier on a finite discrete set of labels K. The following variation is due to Langford and Seeger (2001) for 0–1 loss (see also (Langford, 2002) and (Catoni, 2007).)

Theorem 1.2.1 (PAC-Bayes (McAllester, 1999a; Langford and Seeger, 2001))**.** *Under 0–1 loss, for every $\delta > 0$, $m \in \mathbb{N}$, distribution $\mathscr{D}$ on $\mathbb{R}^k \times K$, and distribution P on $\mathbb{R}^p$,*

$$\mathop{\mathbb{P}}_{S\sim\mathscr{D}^m}\left((\forall Q)\ \mathrm{kl}(\hat{R}_S(Q)||R_{\mathscr{D}}(Q)) \le \frac{\mathrm{KL}(Q||P) + \log\frac{2m}{\delta}}{m-1}\right) \ge 1-\delta. \tag{1.16}$$

The application of PAC-Bayes bounds to learning algorithms is the following: pick a randomized classifier P before seeing the data, which is your prior; the learning algorithm chooses a randomized classifier Q based on the data; then with high probability, Q will have small risk if Q has small empirical risk and the KL divergence between Q and P is small relative to the number of data points.

We also use the following variation of a PAC-Bayes bound, where we consider any bounded loss function.

Theorem 1.2.2 (Linear PAC-Bayes Bound (McAllester, 2013; Catoni, 2007))**.** *Fix $\lambda > 1/2$ and assume the loss takes values in an interval of length L_{max}. For every $\delta > 0$, $m \in \mathbb{N}$, distribution $\mathscr{D}$ on $\mathbb{R}^k \times K$, and distribution P on $\mathbb{R}^p$,*

$$\mathop{\mathbb{P}}_{S\sim\mathscr{D}^m}\left((\forall Q)\ R_{\mathscr{D}}(Q) \le \frac{1}{1-\frac{1}{2\lambda}}\left(\hat{R}_S(Q) + \frac{\lambda L_{max}}{m}(\mathrm{KL}(Q||P) + \log\frac{1}{\delta})\right)\right) \ge 1-\delta. \tag{1.17}$$

1.2.3 Optimal Prior and Posterior

A *Gibbs posterior* (for a prior P) is a distribution that is absolutely continuous with respect to P with a Radon–Nikodym derivative proportional to $\exp(-\tau\hat{R}_S(\mathbf{w}))$. In statistical mechanics

literature, the parameter τ is referred to as *inverse temperature*. Gibbs distributions arise as the solutions of various optimization problems.

Gibbs posterior as the optimal posterior

Fix a prior distribution P. It is well known that the optimal data-dependent randomized classifier Q that minimizes the PAC-Bayes risk bound is a Gibbs posterior. The result appears in McAllester (2003, Thm. 2) for a PAC-Bayes bound that is valid for any measurable loss functions, but the hypothesis class is restricted to be finite. A more general result for uncountable $\mathscr{H}$ was developed by Catoni (2007, Lem. 1.1.3) in the case of bounded risk.

Thus for a given P and a bounded loss function, the optimal distribution Q minimizing the linear PAC-Bayes bound satisfies

$$\frac{dQ}{dP}(\mathbf{w}) = \frac{\exp(-\tau \hat{R}_S(\mathbf{w}))}{\int \exp(-\tau \hat{R}_S(\mathbf{w}))p(d\mathbf{w})}. \tag{1.18}$$

For some τ, λ in the linear PAC-Bayes bound stated in Theorem 1.2.2 can be expressed in terms of the inverse temperature parameter τ.

Local Priors

It is natural to consider a fixed data-dependent posterior $Q = Q(S)$ and ask what prior optimizes a PAC-Bayes bound. Catoni (2007) studied this question and showed that the optimal prior (in expectation) is

$$P = \mathop{\mathbb{E}}_{S\sim\mathscr{D}}[Q(S)]. \tag{1.19}$$

This prior is not available in practice because we do not know $\mathscr{D}$. Instead, one can study some non-optimal but data-distribution-dependent prior distributions.

In the case when $Q(\mathbf{w})$ is a Gibbs classifier with density proportional to

$$e^{-\tau \hat{R}_S(\mathbf{w}) - \gamma F_Q(S,\mathbf{w})}, \tag{1.20}$$

Lever, Laviolette, and Shawe-Taylor (2013) were able to bound the KL divergence with a distribution-dependent prior whose prior density is

$$e^{-\tau R(\mathbf{w}) - \gamma F_P(\mathscr{D},\mathbf{w})}. \tag{1.21}$$

While this prior choice is not optimal, it is an interesting case to study due to optimality of Gibbs distributions discussed above. The functions F_Q and F_P may be different and act as regularizers. They may depend on the parameters of the hypothesis or perform a data-dependent regularization. In (Lever, Laviolette, and Shawe-Taylor, 2013), the authors give localized PAC-Bayes bounds for these particular choices of P and Q.

The use of an informed prior gives a much tighter PAC-Bayes bound. A generic PAC-Bayes bound depends on $\mathrm{KL}(Q||P)$. This term may be very large when P is chosen poorly and is not tailored for the specific task at hand. In the analysis presented by Lever, Laviolette, and Shawe-Taylor (2013), they eliminate the KL term by replacing it with an upper bound when P and Q are chosen as described above. In Chapter 3 we study local prior bounds and develop a new data-dependent prior bound that depends on the properties of the algorithm used to choose P.

For a use of local priors, see, e.g., Parrado-Hernández et al. (2012). In this work, the authors provide tighter PAC-Bayes generalization bounds for SVM classifiers. They explore several strategies to achieve this. One idea is to use part of the dataset to learn a better prior for a PAC-Bayes bound, which the authors call a prior PAC Bayes bound. Another idea is to choose a Gaussian prior with the mean depending on the data distribution. Since the data distribution is not available, the authors instead use an empirical data distribution and then upper bound the difference between the expected parameter value under the empirical distribution and the true data distribution. This approach yields a bound named an expectation prior PAC-Bayes bound (Parrado-Hernández et al., 2012, Thm. 9). Their experiments demonstrate that the use of informative priors for SVM classifiers results in tighter PAC-Bayes bounds.

1.2.4 Gibbs posteriors in (generalized) Bayesian Inference

In Section 1.2.3 we introduced Gibbs distributions and discussed how they relate to PAC-Bayes generalization bounds. In this section we discuss other optimality properties of Gibbs distributions.

Bissiri, Holmes, and Walker (2016) introduce "general Bayesian updating", a framework that generalizes the classical Bayesian inference by replacing the negative log likelihood in the Bayesian update of the posterior with a more general loss function.

More precisely, if $P(\mathbf{w})$ represents your prior beliefs about the parameters $\mathbf{w}$, then for some loss function and the corresponding empirical risk $\hat{R}_S(\mathbf{w})$ on the observed data S, the

general Bayesian posterior is proportional to

$$e^{-m\hat{R}_S(\mathbf{w})}P(\mathbf{w}). \tag{1.22}$$

This is equivalent to Eq. (1.18) with a rescaled loss function, i.e., the density of a Gibbs posterior with $\tau = m$. In a special case, where the loss function is the negative log likelihood and $\tau = m$, the term $-\tau\hat{R}_S(\mathbf{w})$ is the expected log likelihood under Q. This demonstrates the connection among the PAC-Bayes optimal posteriors, general Bayesian inference, and classical Bayesian inference. However, note that the latter works under an assumption that the likelihood contains the true data generating distribution. In contrast, the former frameworks are valid under model misspecification. Most PAC-Bayes generalization bounds require a bounded loss function. Germain et al. (2016) study the connection between PAC-Bayes and Bayesian inference and extend PAC-Bayes generalization bounds for regression tasks and real-valued unbounded loss functions. Similar connections have been made by Zhang (2006a), Zhang (2006b), Jiang and Tanner (2008), Grünwald (2012), and Grünwald and Mehta (2016).

The log normalizing constant of a Gibbs posterior is

$$\log \int \exp(-\tau\hat{R}_S(\mathbf{w}'))P(\mathrm{d}\mathbf{w}'). \tag{1.23}$$

In the thermodynamics literature, this is an *entropy*. We encounter a local version of the entropy in Chapter 4 when we study Entropy-SGD. The entropy also arises in minimax regret analysis (Yamanishi, 1998; Yamanishi, 1999). There it is known as the extended stochastic complexity (ESC; Yamanishi 1999). The ESC is applied to both batch and online learning. For online learning algorithms, it can be shown that ESC minimizes the so-called worst case regret.[1] The ESC also asymptotically achieves minimax estimation error.[2] In Yamanishi (1999) ESC upper bounds the so-called Relative Cumulative Loss for an online learning algorithm. The Relative Cumulative Loss is the difference between the accumulated loss achieved by the online learning algorithm and the minimal accumulated empirical loss for the best hypothesis in the class (empirical risk times the number of examples). See (Yamanishi, 1998; Yamanishi, 1999) for more details. Yamanishi also connects his work to MDL and generalized Bayesian inference.

[1] The worst case regret is defined as the maximum difference between cumulative logarithmic loss of the online learning algorithm and the minimum cumulative logarithmic loss achievable by some $h \in \mathscr{H}$.

[2] The estimation error here is slightly different from the one defined in the beginning of the chapter and is adapted for online learning algorithms. Bounds on estimation error are also often referred to as excess risk or oracle bounds.

1.2.5 PAC-Bayes risk bound as an optimization objective

Variational Inference

Bayesian inference employs Bayes rule to express a posterior distribution over the hypothesis class. In most but the simplest scenarios, this distribution is computationally intractable. One can target this problem by building approximate samplers for the posterior distribution (MCMC methods). An alternative approach is provided by variational methods.

In variational Bayesian inference, one replaces the exact inference problem, with an approximate one. The approximate inference problem can then be angled as an optimization problem.

The density of the Bayesian posterior measure Q_{Bayes} on the hypothesis space is given by

$$p(\mathbf{w}|S) = \frac{p(\mathbf{w},S)}{\int p(\mathbf{w},S)\,\mathrm{d}\mathbf{w}}. \tag{1.24}$$

The integral in the denominator is the marginal density of the observations S, also called the evidence. The evidence is intractable for many models of interest. In variational inference, the goal is to find a distribution close to the Bayesian posterior Q_{Bayes}. We can formalize this as finding a distribution Q_{VI} (with density $q(\mathbf{w})$) within a tractable family of probability measures $\mathscr{Q}$ satisfying

$$Q_{\text{VI}} = \arg\min_{Q\in\mathscr{Q}} \mathrm{KL}(Q||Q_{\text{Bayes}}). \tag{1.25}$$

Computing this KL term is intractable. However, the optimization problem can be seen to be equivalent to

$$\begin{aligned} Q_{\text{VI}} &= \arg\max_{Q\in\mathscr{Q}} \log p(S) - \mathrm{KL}(Q||Q_{\text{Bayes}}) && (1.26)\\ &= \arg\max_{Q\in\mathscr{Q}} \mathbb{E}[\log p(\mathbf{w},S)] - \mathbb{E}[\log q(\mathbf{w})] && (1.27)\\ &= \arg\max_{Q\in\mathscr{Q}} \mathbb{E}[\log p(S|\mathbf{w})] - \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})). && (1.28) \end{aligned}$$

The objective in Eq. (1.28) is called the evidence lower bound objective (ELBO). In modern stochastic variational inference, one then proceeds by using stochastic gradient method to optimize the ELBO objective and find Q_{VI}. For a review on variational inference methods, see, e.g., Blei, Kucukelbir, and McAuliffe (2017).

The first term in the ELBO objective is the expected log likelihood, which is unknown. During optimization, it is usually replaced with its Monte Carlo estimate. Thus in variational

inference, we optimize the parameters of Q to maximize

$$\hat{R}_S(Q) - \beta \mathrm{KL}(Q||P). \tag{1.29}$$

for some prior P. The reader may recognize that the ELBO objective is of the same form as Eq. (1.11), and, in particular, a linear PAC-Bayes bound on the risk (Theorem 1.2.2), with the loss chosen to be log loss. The connection between minimizing PAC-Bayes bounds under log loss and maximizing log marginal densities is the subject of recent work by Germain et al. (2016).

Kingma, Salimans, and Welling (2015) use variational inference to improve the dropout technique in neural network training. The original Gaussian dropout (Srivastava et al., 2014) can be interpreted as sampling stochastic weights. The sampling distribution is isotropic gaussian, centered at the current weight values, with variance equal to the dropout rate (and thus the variance is equal for all weights in the network). Kingma, Salimans, and Welling (2015) treat the dropout rate as a variational parameter which is learned via optimization.

In summary, the variational dropout can be interpreted as optimizing the parameters of Q by minimizing a PAC-Bayes risk bound with a fixed prior. The posterior distribution on the parameters is a stochastic classifier Q, which is an isotropic Gaussian with variances tuned to the individual weights. This closely resembles our work presented on PAC-Bayes risk bound optimization and described in Chapter 2.

1.3 Algorithms and Stability

An algorithm is uniformly stable if its output is relatively insensitive to a change in the input. This notion of stability is fairly strong as it is independent of the data distribution and thus has to account for the worst case scenarios and very unlikely draws of the training data. Uniform stability of a learning algorithm allows one to bound the performance of the classifier returned by the algorithm.

Definition 1.3.1 (Uniform Stability; Bousquet and Elisseeff 2002)**.** An algorithm $\mathscr{A}$ is ε-uniformly stable with respect to a loss function ℓ if, for all pairs $S, S' \in Z^m$ that differ at only one coordinate, we have $\hat{R}_S(\mathscr{A}(S)) - \hat{R}_S(\mathscr{A}(S')) \leq \varepsilon$.

Note, that uniform stability depends on the dataset size m and requires the algorithm to be deterministic. One can easily extend this to randomized algorithms and obtain generalization in expectation. See (Hardt, Recht, and Singer, 2015) for more details.

We can relate stability to concentration bounds using a method of bounded differences. In particular, using McDiarmid's inequality, it is fairly straightforward to get a generalization bound for an output of a uniformly stable algorithm $\mathscr{A}$.

Theorem 1.3.2 (McDiarmid's Inequality (Mendelson, 2003))**.** *Let $s_1,..,s_m$ be random variables, and $S = (s_1,..,s_m)$. Assume vectors S and S_i differ at only one coordinate i. Let a function F map S to $\mathbb{R}$. Then if there exists $(c_1,...,c_m)$, such that F satisfies*

$$\sup_{S,S'} |F(S) - F(S')| \leq c_i \tag{1.30}$$

for all $i \in \{1,...,m\}$, then for all $\varepsilon > 0$

$$\mathbb{P}\Big(|F(S) - \mathop{\mathbb{E}}_{S}[F(S)]| > \varepsilon\Big) \leq \exp\left(\frac{-2\varepsilon^2}{\sum_{n=1}^{N} c_n^2}\right). \tag{1.31}$$

Now consider $F(S) = R(\mathscr{A}(S)) - \hat{R}_S(\mathscr{A}(S))$. Then it is straightforward to show that $\mathop{\mathbb{E}}_{S}[F(S)]$ is bounded by ε. Assume that the loss function ℓ is bounded and takes values in an interval of length $L_{\max}$. Then we can easily demonstrate that $|F(S) - F(S')|$ is bounded in terms of ε and $L_{\max}$. It is a straightforward exercise to apply McDiarmid's inequality to F to get a bound on $|F(S) - \mathop{\mathbb{E}}_{S}[F(S)]|$, and a bound on the risk follows.

Theorem 1.3.3 (Uniform stability implies generalization; Bousquet and Elisseeff 2002)**.** *Let an algorithm $\mathscr{A}$ be $\varepsilon-$uniformly stable. Assume the loss function takes values in an interval of length L_{max}. Then with probability at least $1-\delta$ over the training sample S,*

$$R(\mathscr{A}(S)) \leq \hat{R}_S(\mathscr{A}(S)) + 2\varepsilon + (2m\varepsilon + L_{max})\frac{\log\frac{1}{\delta}}{2m}. \tag{1.32}$$

There exists a number of other notions of stability, that either imply generalization with high probability, or in expectation. For a comparison, see, e.g. Bousquet and Elisseeff (2002) and Bassily et al. (2016).

1.3.1 Stochastic Gradient Descent

The workhorse of modern machine learning is stochastic gradient descent (SGD). In the context of supervised learning, SGD iteratively takes steps along unbiased (though, in practice, correlated) estimates of the gradient of a risk for a differentiable loss function l.

More concretely, given an initial hypothesis $\mathbf{w}_0 \in \mathbb{R}^p$, SGD repeatedly performs the updates

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \eta_n \frac{1}{k} \sum_{i=1}^{k} \nabla_{\mathbf{w}_n} \ell(\mathbf{w}_n, z_{j_i}) \tag{1.33}$$

where, on each round $n = 1, 2, \ldots$, some number $k < m$ indices $j_1, \ldots, j_k$ are chosen uniformly at random and without replacement from $[m] := \{1, 2, \ldots, m\}$.

1.3.2 Stability of SGD

In a recent paper by Hardt, Recht, and Singer (2015), the authors show that under some regularity conditions on the loss function and its gradients, SGD is an $\varepsilon-$uniformly stable algorithm. The ε decays at the rate of $1/m$, where m is the sample size. It depends on the step sizes (that have to be non-increasing during training). It also grows sub-linearly with the number of SGD steps taken.

Theorem 1.3.4. *(Hardt, Recht, and Singer, 2015) Let the loss function $l \in [0,1]$ be L-Lipschitz and β-smooth. Assume we run SGD for N steps, with step sizes $\eta_n < c/n$. Then SGD is ε-uniformly stable with*

$$\varepsilon \leq \frac{1 + 1/\beta c}{m-1} (2cL^2)^{\frac{1}{\beta c+1}} T^{\frac{\beta c}{\beta c+1}}. \tag{1.34}$$

While this is an interesting result regarding the stability of the algorithm, it only gives a generalization bound in expectation, rather than a high probability bound. Furthermore, as discussed above, uniform stability is a very strong requirement and the notion is independent of the data distribution, and thus the risk bounds obtained using uniform stability are very loose for "nice" datasets. It would allow us to make few SGD steps before the bound on the risk exceeds 0.5 and becomes trivial for binary classification tasks.

1.4 Differential privacy

Differential privacy is a formalization of the privacy an algorithm affords every individual in a data set ((Dwork, 2008a; Dwork and Roth, 2014a) for a survey). Differential privacy has important applications in real world problems when machine learning models are applied to datasets where it is necessary to maintain the anonymity of individual users in the dataset. For example, if a classifier is to be trained on sensitive data and then released publicly, it should not be possible to learn anything about any one user by studying the classifier.

Here we formally define some of the differential privacy related terms used in the main text. (See (Dwork, 2006; Dwork and Roth, 2014b) for more details.)

Let $U, U_1, U_2, \ldots$ be independent uniform $(0,1)$ random variables, independent also of any random variables introduced by $\mathbb{P}$ and $\mathbb{E}$, and let $\pi : \mathbb{N} \times [0,1] \to [0,1]$ satisfy $(\pi(1,U), \ldots, \pi(k,U)) \stackrel{d}{=} (U_1, \ldots, U_k)$ for all $k \in \mathbb{N}$. Write π_k for $\pi(k,\cdot)$.

Definition 1.4.1. A *randomized algorithm* $\mathscr{A}$ from R to T, denoted $\mathscr{A} : R \rightsquigarrow T$, is a measurable map $\mathscr{A} : [0,1] \times R \to T$. Associated to $\mathscr{A}$ is a (measurable) collection of random variables $\{\mathscr{A}_r : r \in R\}$ that satisfy $\mathscr{A}_r = \mathscr{A}(U,r)$. When there is no risk of confusion, we write $\mathscr{A}(r)$ for $\mathscr{A}_r$.

Definition 1.4.2. A randomized algorithm $\mathscr{A} : Z^m \rightsquigarrow T$ is (ε, δ)-*differentially private* if, for all pairs $S, S' \in Z^m$ that differ at only one coordinate, and all measurable subsets $B \subseteq T$, we have $\mathbb{P}(\mathscr{A}(S) \in B) \leq e^{\varepsilon}\, \mathbb{P}(\mathscr{A}(S') \in B) + \delta$.

We write ε-differentially private to mean $(\varepsilon, 0)$-differentially private algorithm.

Definition 1.4.3. Let $\mathscr{A} : R \rightsquigarrow T$ and $\mathscr{A}' : T \rightsquigarrow T'$. The *composition* $\mathscr{A}' \circ \mathscr{A} : R \rightsquigarrow T'$ is given by $(\mathscr{A}' \circ \mathscr{A})(u,r) = \mathscr{A}'(\pi_2(u), \mathscr{A}(\pi_1(u), r))$.

Lemma 1.4.4 (post-processing)**.** *Let $\mathscr{A} : Z^m \rightsquigarrow T$ be (ε, δ)-differentially private and let $F : T \rightsquigarrow T'$ be arbitrary. Then $F \circ \mathscr{A}$ is (ε, δ)-differentially private.*

Differential privacy can also be viewed as a very strong notion of algorithmic stability. In particular, every differentially private algorithm is uniformly stable. This sufficiency result is well known and follows immediately from the definition of differential privacy. In some work, differential provacy is also referred to as *Max-KL Stability* to highlight the connection to other stability definitions (Bassily et al., 2016).

We introduce several additional generalization bounds in Chapters 3 and 4, where we use differential privacy to derive a data dependent PAC-Bayes bound and to analyze an existing learning algorithm in terms of this new bound.

Chapter 2

Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks

2.1 Introduction

By optimizing a PAC-Bayes bound, we show that it is possible to compute nonvacuous numerical bounds on the generalization error of deep *stochastic* neural networks with millions of parameters, despite the training data sets being one or more orders of magnitude smaller than the number of parameters. To our knowledge, these are the first explicit and nonvacuous numerical bounds computed for trained neural networks in the modern deep learning regime where the number of network parameters eclipses the number of training examples.

The bounds we compute are data dependent, incorporating millions of components optimized numerically to identify a large region in weight space with low average empirical error around the solution obtained by stochastic gradient descent (SGD). The data dependence is essential: indeed, the VC dimension of neural networks is typically bounded below by the number of parameters, and so one needs as many training data as parameters before (uniform) PAC bounds are nonvacuous, i.e., before the generalization error falls below 1. To put this in concrete terms, on MNIST, having even 72 hidden units in a fully connected first layer yields vacuous PAC bounds.

Evidently, we are operating far from the worst case: observed generalization cannot be explained in terms of the regularizing effect of the size of the neural network alone. This is an old observation, and one that attracted considerable theoretical attention two decades ago: Bartlett (Bartlett, 1997; Bartlett, 1998) showed that, in large (sigmoidal) neural networks, when the learned weights are small in magnitude, the fat-shattering dimension is more important than the VC dimension for characterizing generalization. In particular,

Bartlett established classification error bounds in terms of the empirical margin and the fat-shattering dimension, and then gave fat-shattering bounds for neural networks in terms of the *magnitudes* of the weights and the depth of the network alone. Improved norm-based bounds were obtained using Rademacher and Gaussian complexity by Bartlett and Mendelson (2002) and Koltchinskii and Panchenko (2002).

These norm-based bounds are the foundation of our current understanding of neural network generalization. It is widely accepted that these bounds explain observed generalization, at least "qualitatively" and/or when the weights are explicitly regularized. Indeed, recent work by Neyshabur, Tomioka, and Srebro (2014) puts forth the idea that SGD performs implicit norm-based regularization. Somewhat surprisingly, when we investigated state-of-the-art Rademacher bounds for ReLU networks, the bounds were vacuous when applied to solutions obtained by SGD on real networks/datasets. We discuss the details of this analysis in Section 2.9. While most researchers assume these bounds are explanatory even if they are numerically loose, we argue that the bounds, being numerically vacuous, do not establish generalization on their own. It is worth highlighting that nonvacuous bounds may exist under hypotheses that currently only yield vacuous bounds. This is an important avenue to investigate.

2.1.1 Understanding SGD

Our investigation was instigated by recent empirical work by Zhang, Bengio, Hardt, Recht, and Vinyals (Zhang et al., 2017), who show that stochastic gradient descent (SGD), applied to deep networks with millions of parameters, is:

1. able to achieve ≈ 0 training error on CIFAR10 and IMAGENET and still generalize (i.e., test error remains small, despite the potential for overfitting);

2. still able to achieve ≈ 0 training error even after the labels are *randomized*, and does so with only a small factor of additional computational time.

Taken together, these two observations demonstrate that these networks have a tremendous capacity to overfit and yet SGD does not abuse this capacity as it optimizes the surrogate loss, despite the lack of explicit regularization.

It is a major open problem to explain this phenomenon. A natural approach would be to show that, under realistic hypotheses, SGD performs implicit regularization or tends to find solutions that possess some particular structural property that we already know to be connected to generalization. However, in order to complete the logical connection, we need an associated error bound to be nonvacuous in the regime of model size / data size where we hope to explain the phenomenon.

This work establishes a potential candidate, building off ideas by Langford (2002) and Langford and Caruana (2002a): On a binary class variant of MNIST, we find that SGD solutions are nearby to relatively large regions in weight space with low average empirical error. We find this structure by optimizing a PAC-Bayes bound, starting at the SGD solution, obtaining a nonvacuous generalization bound for a stochastic neural network. Across a variety of network architectures, our PAC-Bayes bounds on the test error are in the range 16–22%. These are far from nonvacuous but loose: Chernoff bounds on the test error based on held-out data are consistently around 3%. Despite the gap, theoreticians aware of the numerical performance of generalization bounds will likely be surprised that it is possible at all to obtain nonvacuous numerical bounds for models with such large capacity trained on so few training examples. While we cannot entirely explain the magnitude of generalization, we can demonstrate nontrivial generalization.

Our approach was inspired by a line of work in physics by Baldassi, Ingrosso, Lucibello, Saglietti, and Zecchina (Baldassi et al., 2015) and the same authors with Borgs and Chayes (Baldassi et al., 2016). Based on theoretical results for discrete optimization linking computational efficiency to the existence of nonisolated solutions, the authors propose a number of new algorithms for learning discrete neural networks by explicitly driving a local search towards nonisolated solutions. On the basis of Bayesian ideas, they posit that these solutions have good generalization properties. In a recent work with Chaudhari, Choromanska, Soatto, and LeCun (Chaudhari et al., 2017), they introduce local-entropy loss and EntropySGD, extending these algorithmic ideas to modern deep learning architectures with continuous parametrizations, and obtaining impressive empirical results.

In the continuous setting, nonisolated solutions correspond to "flat minima". The existence and regularizing effects of flat minima in the empirical error surface was recognized early on by researchers, going back at work by Hinton and Camp (1993) and Hochreiter and Schmidhuber (1997). Hochreiter and Schmidhuber discuss sharp versus flat minima using the language of minimum description length (MDL; (Rissanen, 1983; Grünwald, 2007)). In short, describing weights in sharp minima requires high precision in order to not incur nontrivial excess error, whereas flat minimum can be described with lower precision. A similar coding argument appears in (Hinton and Camp, 1993).

Hochreiter and Schmidhuber propose an algorithm to find flat minima by minimizing the training error while maximizing the log volume of a connected region of the parameter space that yields similar classifiers with similarly good training error. There are very close connections—at both the level of analysis and algorithms—with the work of Chaudhari et al. (2017) and close connections with the approach we take to compute nonvacuous error bounds by exploiting the local error surface. (We discuss more related work in Section 2.10.)

Despite the promising underpinnings, the generalization theorems given by (Chaudhari et al., 2017) have admittedly unrealistic assumptions, and fall short of connecting local-entropy minimization to observed generalization.

The goal of this work is to identify structure in the solutions obtained by SGD that provably implies small generalization error. Computationally, it is much easier to demonstrate that a randomized classifier will generalize, and so our results actually pertain to the generalization error of a *stochastic* neural network, i.e., one whose weights/biases are drawn at random from some distribution on every forward evaluation of the network. Under bounded loss, Fubini's theorem implies that we also obtain a bound on the expected error of a neural network whose weights have been randomly perturbed. It would be interesting to achieve tighter control on the distribution of error or on the error of the mean neural network.

Returning to the goal of explaining SGD and generalization in deep learning more generally, one could study whether the type of structure we exploit to obtain bounds necessarily arises from performing SGD under natural conditions. (We suspect one condition may be that the Bayes error rate is close to zero.) More ambitiously, perhaps the existence of the same structure can explain the success of SGD in practice.

2.1.2 Approach

Our working hypothesis is that SGD finds good solutions only if they are surrounded by a relatively large volume of solutions that are nearly as good. This hypothesis suggests that PAC-Bayes bounds may be fruitful: if SGD finds a solution contained in a large volume of equally good solutions, then the expected error rate of a classifier drawn at random from this volume should match that of the SGD solution. The PAC-Bayes theorem (McAllester, 1999a) bounds the expected error rate of a classifier chosen from a distribution Q in terms of the Kullback–Liebler divergence from some a priori fixed distribution P, and so if the volume of equally good solutions is large, and not too far from the mass of P, we will obtain a nonvacuous bound.

Our approach will be to use optimization to find a broad distribution Q over neural network parameters that minimizes the PAC-Bayes bound, in effect mapping out the volume of equally good solutions surrounding the SGD solution. This idea is actually a modern take on an old idea by Langford and Caruana (Langford and Caruana, 2002a), who apply PAC-Bayes bounds to small two-layer stochastic neural networks (with only 2 hidden units) that were trained on (relatively large, in comparison) data sets of several hundred labeled examples.

The basic idea can be traced back even further to work by Hinton and Camp (Hinton and Camp, 1993), who propose an algorithm for controlling overfitting in neural networks

via the minimum description length principle. In particular, they minimize the sum of the empirical squared error and the KL divergence between a prior and posterior distribution on the weights. Their algorithm is applied to networks with 100's of inputs and 4 hidden units, trained on several hundred labeled examples. Hinton and Camp do not compute numerical generalization bounds to verify that MDL principles alone suffice to *explain* the observed generalization.

Our algorithm more directly extends the work by Langford and Caruana, who propose to construct a distribution Q over neural networks by performing a sensitivity analysis on each parameter after training, searching for the largest deviation that does not increase the training error by more than, e.g., 1%. For Q, Langford and Caruana choose a multivariate normal distribution over the network parameters, centered at the parameters of the trained neural network. The covariance matrix is diagonal, with the variance of each parameter chosen to be the estimated sensitivity, scaled by a global constant. (The global scale is chosen so that the training error of Q is within, e.g., 1% of that of the original trained network.) Their prior P is also a multivariate normal, but with zero mean and covariance given by some scalar multiple of the identity matrix. By employing a union bound, they allow themselves to choose the scalar multiple in a data-dependent fashion to optimize the PAC-Bayes bound.

The algorithm sketched by Langford and Caruana does not scale to modern neural networks for several reasons, but one dominates: in massively overparametrized networks, individual parameters often have negligible effect on the training classification error, and so it is not possible to estimate the *relative* sensitivity of large populations of neurons by studying the sensitivity of neurons in isolation.

Instead, we use stochastic gradient descent to directly optimize the PAC-Bayes bound on the error rate of a stochastic neural network. At each step, we update the network weights and their variances by taking a step along an unbiased estimate of the gradient of (an upper bound on) the PAC-Bayes bound. In effect, the objective function is the sum of i) the empirical surrogate loss averaged over a random perturbation of the SGD solution, and ii) a generalization error bound that acts like a regularizer.

Having demonstrated that this simple approach can construct a witness to generalization, it is worthwhile asking whether these ideas can be extended to the setting of local-entropic loss (Chaudhari et al., 2017). If we view the distribution that defines the local-entropic loss as defining a stochastic neural network, can we use PAC-Bayes bounds to establish nonvacuous bounds on its generalization error?

2.2 Bounds

We will employ three probabilistic bounds to control generalization error: the union bound, a sample convergence bound derived from the Chernoff bound, and the PAC-Bayes bound due to McAllester (1999a). We state the union bound for completeness.

Theorem 2.2.1 (union)**.** *Let $E_1, E_2, \ldots$ be events. Then $\mathbb{P}(\bigcup_n E_n) \leq \sum_n \mathbb{P}(E_n)$.*

Recall that $\mathscr{B}(p)$ denotes the Bernoulli distribution on $\{0,1\}$ with mean $p \in [0,1]$. The following bound is derived from the KL formulation of the Chernoff bound:

Theorem 2.2.2 (sample convergence (Langford and Caruana, 2002a))**.** *For every $p, \delta \in (0,1)$ and $n \in \mathbb{N}$, with probability at least $1-\delta$ over $x \sim \mathscr{B}(p)^n$, $\mathrm{kl}(n^{-1}\sum_{i=1}^n x_i || p) \leq \frac{\log \frac{2}{\delta}}{n}$.*

For convenience, we state again a variant of McAllester's PAC-Bayes bound due to Langford (2002). [1]

Theorem 2.2.3 (PAC-Bayes (McAllester, 1999a; Langford and Seeger, 2001))**.** *For every $\delta > 0$, $m \in \mathbb{N}$, distribution $\mathscr{D}$ on $\mathbb{R}^k \times \{-1,1\}$, and distribution P on $\mathbb{R}^p$, with probability at least $1-\delta$ over $S_m \sim \mathscr{D}^m$, for all distributions Q on $\mathbb{R}^p$,*

$$\mathrm{kl}(\hat{R}_{S_m}(Q) || R(Q)) \leq \frac{\mathrm{KL}(Q||P) + \log \frac{2m}{\delta}}{m-1}.$$

2.2.1 Inverting KL bounds

In the following sections, we will encounter bounds on a quantity $p^* \in [0,1]$ of the form

$$\mathrm{kl}(q||p^*) \leq c$$

for some $q \in [0,1]$ and $c \geq 0$. Thus, we are interested in

$$\mathrm{KL}^{-1}(q|c) := \sup\{p \in [0,1] : \mathrm{kl}(q||p) \leq c\}. \tag{2.1}$$

We are not aware of a simple formula for $\mathrm{KL}^{-1}(q|c)$, although numerical approximations are readily obtained via Newton's method (Section 2.4). For the purpose of gradient-based optimization, we can use the well-known inequality, $2(q-p)^2 \leq \mathrm{kl}(q||p)$, to obtain a simple

[1] In this section we use a bound that is slightly tighter from the ones stated in Section 1.2.2. The latter has a $\log \frac{2m}{\delta}$ term rather than $\log \frac{m}{\delta}$ as in Theorem 2.2.3. However, note, that this term is very small when divided by $m-1$ ($\approx 1e-4$), and the effect on the bounds calculated is negligible.

upper bound

$$\mathrm{KL}^{-1}(q|c) \leq q + \sqrt{c/2}. \tag{2.2}$$

This bound is quantitatively loose near $q \approx 0$, because then $\mathrm{KL}^{-1}(q|c) \approx c$ for $c \ll 1$, versus the upper bound of $\Theta(\sqrt{c})$. On the other hand, when c is large enough that $q + \sqrt{\frac{c}{2}} > 1$, the derivative of $\mathrm{KL}^{-1}(q|c)$ is zero, whereas the upper bound provides a useful derivative.

2.2.2 Learning Algorithm

The PAC-Bayes bound presented in Theorem 3.3.1 leads to the following learning algorithm (McAllester, 1999a):

1. Fix a probability $\delta > 0$ and a distribution P on $\mathbb{R}^p$.
2. Collect an i.i.d. dataset S_m of size m.
3. Compute the optimal distribution Q on $\mathbb{R}^p$ that minimizes the error bound

$$\mathrm{KL}^{-1}\left(\hat{R}_{S_m}(Q) \middle| \frac{\mathrm{KL}(Q||P) + \log\frac{2m}{\delta}}{m-1}\right). \tag{2.3}$$

4. Return the randomized classifier given by Q.

In all but the simplest scenarios, making predictions according to the optimal Q is intractable. However, we can attempt to approximate it.

2.3 PAC-Bayes bound optimization

Let H be a parametric family of classifiers and write h_w for $H(w,\cdot)$. We will interpret h_w as a neural network with (weight/bias) parameters $w \in \mathbb{R}^d$, although the development below is more general.

Fix $\delta \in (0,1)$ and a distribution P on $\mathbb{R}^d$, and let $S_m \sim \mathscr{D}^m$ be m i.i.d. training examples. We aim to minimize the PAC-Bayes bound (Eq. (2.3)) with respect to Q.

For $w \in \mathbb{R}^d$ and $s \in \mathbb{R}^d_+$, let $\mathscr{N}_{w,s} = \mathscr{N}(w, \mathrm{diag}(s))$ denote the multivariate normal distribution with mean w and diagonal covariance $\mathrm{diag}(s)$. As our first simplifications, we replace the PAC-Bayes bound with the upper bound described by Eq. (2.2), replace the empirical loss with its convex surrogate, and restrict Q to the family of multivariate normal distributions

with diagonal covariance structure, yielding the optimization problem

$$\min_{w\in\mathbb{R}^d, s\in\mathbb{R}^d_+} \breve{R}_{S_m}(\mathscr{N}_{w,s}) + \sqrt{\frac{\mathrm{KL}(\mathscr{N}_{w,s}||P) + \log\frac{m}{\delta}}{2(m-1)}}.$$

2.3.1 The Prior

In order to obtain a KL divergence in closed form, we choose P to be multivariate normal. Symmetry considerations would suggest that we choose $P = \mathscr{N}(0, \lambda I)$ for some $\lambda > 0$, however there is no single good choice of λ. (We will also see that there are good reasons not to choose a zero mean, and so we will let w_0 denote the mean to be chosen a priori.)

In order to deal with the problem of choosing λ, we will follow Langford and Caruana (2002a) and use a union-bound argument to choose λ optimally from a discrete set, at the cost of a slight expansion to our generalization bound. In particular, we will take $\lambda = c\exp\{-j/b\}$ for some $j \in \mathbb{N}$ and fixed $b, c \geq 0$. (Hence, b determines a level of precision and c is an upper bound.) If the PAC-Bayes bound for each $j \in \mathbb{N}$ is designed to hold with probability at least $1 - \frac{6}{\pi^2 j^2}$, then, by the union bound (Theorem 2.2.1), it will hold uniformly for all $j \in \mathbb{N}$ with probability at least $1 - \delta$, as desired. During optimization, we will want to avoid discrete optimization, and so we will treat λ as if it were a continuous variable. (We will then discretize λ when we evaluate the PAC-Bayes bound after the fact.) Solving for j, we have $j = b\log\frac{c}{\lambda}$, and so we will replace j with this term during optimization. Taking into account the choice of P and the continuous approximation to the union bound, we have the following minimization problem:

$$\min_{w\in\mathbb{R}^d, s\in\mathbb{R}^d_+, \lambda\in(0,c)} \breve{R}_{S_m}(\mathscr{N}_{w,s}) + \sqrt{\frac{1}{2}B_{\mathrm{RE}}(w,s,\lambda;\delta)} \tag{2.4}$$

where $B_{\mathrm{RE}}(w,s,\lambda;\delta)$ is

$$\frac{\mathrm{KL}(\mathscr{N}_{w,s}||\mathscr{N}(w_0,\lambda I)) + 2\log(b\log\frac{c}{\lambda}) + \log\frac{\pi^2 m}{6\delta}}{m-1}, \tag{2.5}$$

and, using Eq. (1.15), the KL term simplifies to

$$\frac{1}{2}\left(\frac{1}{\lambda}\|s\|_1 - d + \frac{1}{\lambda}\|w - w_0\|_2^2 + d\log\lambda - 1_d\cdot\log s\right).$$

We fix $\delta = 0.025$, $b = 100$, and $c = 0.1$.

2.3.2 Stochastic Gradient Descent

We cannot optimize Eq. (2.4) directly because we cannot compute $\breve{R}_{S_m}(\mathscr{N}_{w,s})$ or its gradients efficiently. We can, however, compute the gradient of the unbiased estimate $\breve{R}_{S_m}(h_{w+\xi\odot\sqrt{s}})$, where $\xi \sim \mathscr{N}_{0,I_d}$. We will use an i.i.d. copy of ξ at each iteration. We did not experiment using mini-batches during bound optimization.

2.3.3 Final PAC-Bayes bound

While we treat λ as a continuous parameter during optimization, the union bound requires that λ be of the form $\lambda = c\exp\{-j/b\}$, for some $j \in \mathbb{N}$. We therefore round λ up or down, choosing that which delivers the best bound, as computed below.

According to the PAC-Bayes and union bound, with probability $1-\delta$, uniformly over all $w \in \mathbb{R}^d$, $s \in \mathbb{R}^d_+$, and λ of the form $c\exp\{-j/b\}$, for $j \in \mathbb{N}$, the error rate of the randomized classifier $Q = \mathscr{N}_{w,s}$ is bounded by

$$\mathrm{KL}^{-1}(\hat{R}_{S_m}(Q)|B_{\mathrm{RE}}(w,s,\lambda;\delta)).$$

We cannot compute this bound exactly because computing $\hat{R}_{S_m}(Q)$ is intractable. However, we can obtain unbiased estimates and apply the sample convergence bound (Theorem 2.2.2). In particular, given n i.i.d. samples $w_1,\ldots,w_n$ from Q, we produce the Monte Carlo approximation $\hat{Q}_n = \sum_{i=1}^n \delta_{w_i}$, for which $\hat{R}_{S_m}(\hat{Q}_n)$ is exactly computable, and obtain the bound

$$\begin{aligned}\hat{R}_{S_m}(Q) &\le \overline{\hat{R}^{n,\delta'}_{S_m}(Q)} \\ &:= \mathrm{KL}^{-1}(\hat{R}_{S_m}(\hat{Q}_n)|n^{-1}\log 2/\delta'),\end{aligned}$$

which holds with probability $1-\delta'$. By another application of the union bound,

$$R(Q) \le \mathrm{KL}^{-1}(\overline{\hat{R}^{n,\delta'}_{S_m}(Q)}|B_{\mathrm{RE}}(w,s,\lambda;\delta)), \tag{2.6}$$

with probability $1-\delta-\delta'$. We use this bound in our reported results.

2.4 Approximating $\mathrm{KL}^{-1}(q|c)$

There is no simple formula for $\mathrm{KL}^{-1}(q|c)$, but we can approximate it via root-finding techniques. For all $q \in (0,1)$ and $c \ge 0$, define $h_{q,c}(p) = \mathrm{KL}(q||p) - c$. Then $h'_{q,c}(p) = \frac{1-q}{1-p} - \frac{q}{p}$. Given a sufficiently good initial estimate p_0 of a root of $h_{q,c}(\cdot)$, we can obtain

improved estimates of a root via Newton's method:

$$p_{n+1} = \mathrm{N}(p_n; q, c) \text{ where } \mathrm{N}(p; q, c) = p - \frac{h_{q,c}(c)}{h'_{q,c}(p)}.$$

This suggests the following approximation to $\mathrm{KL}^{-1}(q|c)$:

1. Let $\tilde{b} = q + \sqrt{\frac{c}{2}}$.
2. If $\tilde{b} \geq 1$, then return 1.
3. Otherwise, return $\mathrm{N}^k(\tilde{b})$, for some integer $k > 0$.

Our reported results in Table 2.1 use five steps of Newton's method.

2.5 Network symmetries

In an ideal world, we would account for all the network symmetries when computing the KL divergence in the PAC-Bayes bound. However, it does not seem to be computationally feasible to account for the symmetries, as we discuss below. One consequence of randomly initializing a neural network's weights is that at least some symmetries are broken. If we do not expect SGD during training to completely change the direction from the origin as determined by the initial weight configuration, w_0, then w_0 will be a better mean for the PAC-Bayes prior P than the origin. In fact, taking into account some of the initial symmetry breaking in this way leads to much better bounds than setting the means to zero.

2.5.1 Bounds from mixtures

Fix a neural network architecture $H : \mathbb{R}^d \times \mathbb{R}^k \to \{-1, 1\}$ and write h_w for $H(w, \cdot)$. It has long been appreciated that distinct parametrizations $w, w' \in \mathbb{R}^d$ can lead to the same *functions* $h_w = h_{w'}$, and so the set $\mathbb{R}^p = \{h_w : w \in \mathbb{R}^d\}$ of classifiers defined by a neural network architecture is a quotient space of $\mathbb{R}^d$.

For the purposes of understanding the generalization error of neural networks, we would ideally work directly with $\mathbb{R}^p$. Let P, Q be a distributions on $\mathbb{R}^d$, i.e., stochastic neural networks. Then P and Q induce distributions on $\mathbb{R}^p$, which we will denote by $\bar{P}$ and $\bar{Q}$, respectively. For the purposes of the PAC-Bayes bound, it is the KL divergence $\mathrm{KL}(\bar{Q}||\bar{P})$ that upper bounds the performance of the stochastic neural network Q. In general, $\mathrm{KL}(\bar{Q}||\bar{P}) \leq \mathrm{KL}(Q||P)$, but it is difficult in practice to approximate the former because the quotient space is extremely complex.

One potential way to approach $\mathbb{R}^p$ is to account for symmetries in the parameterization. A *network symmetry* is a map $\sigma : \mathbb{R}^d \to \mathbb{R}^d$ such that, for all $w \in \mathbb{R}^d$, we have $h_w = h_{\sigma(w)}$. As an example of such a symmetry, in a fully connected network with identical activation functions at every unit, the function computed by the network is invariant to permuting the nodes with a hidden layer. Let $\mathbb{S}$ be any finite set of symmetries possessed by the architecture. For every distribution Q on $\mathbb{R}^d$ and network symmetry σ, we may define $Q_\sigma = Q \circ \sigma^{-1}$ to be the distribution over networks obtained by first sampling network parameters from Q and then applying the map σ to obtain a network that computes the same function.

Define $Q^{\mathbb{S}} = \frac{1}{|\mathbb{S}|} \sum_{\sigma \in \mathbb{S}} Q_\sigma$. Informally, Q and $Q^{\mathbb{S}}$ are identical when viewed as distributions on functions, yet $Q^{\mathbb{S}}$ spreads its mass evenly over equivalent parametrizations. In particular, for any data set S, we have $\hat{R}_S(Q) = \hat{R}_S(Q^{\mathbb{S}})$. We call $Q^{\mathbb{S}}$ a symmetrized version of Q. The following lemma states that symmetrized versions always have smaller KL divergence with respect to distributions that are invariant to symmetrization: Before stating the lemma, recall that the differential entropy of an absolutely continuous distribution Q on $\mathbb{R}^d$ with density q is $\int q(x) \log q(x) \mathrm{d}x \in \mathbb{R} \cup \{-\infty, \infty\}$.

Lemma 2.5.1. *Let $\mathbb{S}$ be a finite set of network symmetries, let P be an absolutely continuous distribution such that $P = P_\sigma$ for all $\sigma \in \mathbb{S}$, and define $Q^{\mathbb{S}}$ as above for some arbitrary absolutely continuous distribution Q on $\mathbb{R}^d$ with finite differential entropy. Then* $\mathrm{KL}(Q^{\mathbb{S}}||P) = \mathrm{KL}(Q||P) - \mathrm{KL}(Q||Q^{\mathbb{S}}) \leq \mathrm{KL}(Q||P)$.

The above lemma can be generalized to distributions over (potentially infinite) sets of network symmetries.

It follows from this lemma that one can do no worse by accounting for symmetries using mixtures, provided that one is comparing to a distribution P that is invariant to those symmetries. In light of the PAC-Bayes theorem, this means that a generalization bound based upon a KL divergence that does not account for symmetries can likely be improved. However, for a finite set $\mathbb{S}$ of symmetries, it is easy to show that the improvement is bounded by $\log |\mathbb{S}|$, which suggests that, in order to obtain appreciable improvements in a numerical bound, one would need to account for an exponential number of symmetries. Unfortunately, exploiting this many symmetries seems intractable. It is hard to obtain useful lower bounds to $\mathrm{KL}(Q||Q^{\mathbb{S}})$, while upper bounds from Jensen's inequality lead to negative (hence vacuous) lower bounds on $\mathrm{KL}(Q^{\mathbb{S}}||P)$.

In this work, we therefore take a different approach to dealing with symmetries. Neural networks are randomly initialized in order to *break* symmetries. Combined with the idea that the learned parameters will reflect these broken symmetries, we choose our prior P to be located at the random initialization, rather than at zero.

2.6 Experiments

Starting from neural networks whose weights have been trained by SGD (with momentum) to achieve near-perfect accuracy on a binary class variant of MNIST, we then optimize a PAC-Bayes bound on the error rate of stochastic neural network whose weights are random perturbations of the weights learned by SGD. We consider several different network architectures, varying both the depth and the width of the network.

Experiment	T-600	T-1200	T-300^2	T-600^2	T-1200^2	T-600^3	R-600
Train error	0.001	0.002	0.000	0.000	0.000	0.000	0.007
Test error	0.018	0.018	0.015	0.016	0.015	0.013	0.508
SNN train error	0.028	0.027	0.027	0.028	0.029	0.027	0.112
SNN test error	0.034	0.035	0.034	0.033	0.035	0.032	0.503
PAC-Bayes bound	0.161	0.179	0.170	0.186	0.223	0.201	1.352
KL divergence	5144	5977	5791	6534	8558	7861	201131
# parameters	471k	943k	326k	832k	2384k	1193k	472k
VC dimension	26m	56m	26m	66m	187m	121m	26m

Table 2.1 Results for experiments on binary class variant of MNIST. SGD is either trained on (T) true labels or (R) random labels. The network architecture is expressed as N^L, indicating L hidden layers with N nodes each. Errors are classification error. The reported VC dimension is the best known upper bound (in millions) for ReLU networks. The SNN error rates are tight upper bounds (see text for details). The PAC-Bayes bounds upper bound the test error with probability 0.965.

2.6.1 Dataset

We use the MNIST handwritten digits data set (LeCun, Cortes, and Burges, 2010) as provided in Tensorflow (Abadi et al., 2015), where the dataset is split into the training set (55000 images) and test set (10000 images). (We do not use the validation set.) Each MNIST image is black and white and 28-pixels square, resulting in a network input dimension of $k = 784$. MNIST is usually treated as a multiclass classification problem. In order to use standard PAC-Bayes bounds, we produce a binary classification problem by mapping numbers $\{0,\dots,4\}$ to label 1 and $\{5,\dots,9\}$ to label -1. In some experiments, we train on random labels, i.e., binary labels drawn independently and uniformly at random.

2.6.2 Initial network training by SGD

All experiments are performed on multilayer perceptrons, i.e., feed-forward neural networks with 2 or more layers, each layer fully connected to the previous and next layer. We choose a standard initialization scheme for the weights and biases: Weights are initialized randomly from a normal distribution (with mean zero and standard deviation $\sigma = 0.04$) that is truncated to $[-2\sigma, 2\sigma]$. Biases are initialized to a constant value of 0.1 for the first layer and 0 for the remaining layers. We let w_0 denote this random initialization of the weights (and biases).

We use REctified Linear Unit (RELU) activations at every hidden node. The last layer is linear. In order to train the weights, we minimize the logistic loss by SGD with momentum (learning rate 0.01; momentum 0.9). SGD is run in mini-batches of size 100. These settings are similar to those in (Zhang et al., 2017).

On our binary class variant of MNIST, we train several neural network architectures of varying depth and width (see Table 2.1). In each case, we train for a total of 20 epochs. We also train a small network (with one 600-unit hidden layer) on *random* labels, in order to demonstrate the large capacity of the network. Obtaining $\approx$0 training error requires 120 epochs. See the first two rows of Table 2.1 for the train/test error rates.

2.6.3 PAC-Bayes bound optimization

Starting from weights w learned by SGD, we construct a stochastic neural network with a multivariate normal distribution $Q = \mathcal{N}_{w,s}$ over its weights with mean w and covariance $\mathrm{diag}(s)$. We initialize s to $|w|$ and $|w|/10$ for true- and random-label experiments, respectively.

We optimize the PAC-Bayes bound (Eq. (2.4)) starting from an initial choice of e^{-6} for the prior variance λ and the prior mean fixed at the random initialization w_0. (See Section 2.5 for a discussion of this subtle but important innovation.) We transform the constrained optimization over $w \in \mathbb{R}^d$, $s \in \mathbb{R}^d_+$, and $\lambda \in (0, c)$, into an unconstrained optimization over w, $\frac{1}{2}\log(s)$, and $\frac{1}{2}\log(\lambda)$, respectively.

We optimize the objective by gradient descent with the RMSprop optimizer (with decay 0.9, as is typical). We use the unbiased estimate of the gradient of the empirical surrogate error of the randomized classifier $Q = \mathcal{N}_{w,s}$. We set the learning rate to 0.001 for the first 150000 iterations, and then lower it to 0.0001 for the final 50000 iterations. For the random-label experiment, we optimize the bound with a smaller 0.0001 learning rate for 500000 iterations. In both cases, the learning rate is tuned so that the objective decreases smoothly during learning.

Algorithm 1 is pseudo code for optimizing the PAC-Bayes bound. The code implements vanilla SGD, although it can be easily modified to use an optimizer like RMSprop.

2.6.4 Reported values

Reported error rates correspond to classification error. Train and test error rates are empirical averages for networks learned by SGD. In light of 10000 test data points and the observed error rates, upper bounds via Theorem 2.2.2 are only 0.005 higher.

Reported train and test error rates for the stochastic neural networks (abbreviated SNN) are upper bounds computed by an application of Theorem 2.2.2 as described in Section 2.3.3 with $\delta' = 0.01$ and $n = 150000$. These numbers produce estimates within 0.001–0.002.

The PAC-Bayes bound is computed as described in Section 2.3.3. Each bound holds with probability 0.965 over the choice of the training set and the draws from the learned SNN Q. For the random-label experiment, we report $\sqrt{\frac{1}{2}B_{\mathrm{RE}}(w,s,\lambda;\delta)}$ from Eq. (2.5), since the PAC-Bayes bound is vacuous when this quantity is greater than 1.

Our VC-dimension bounds for ReLU networks are computed from a formula communicated to us by Bartlett (2017). These bounds are in $O(LW\log W)$, where L is the number of layers and W is the total number of tunable parameters across layers.

2.7 Results

See Table 2.1.[Training results and bounds for MNIST trained networks] All SGD trained networks achieve perfect or near-perfect accuracy on the training data. On true labels, the SNN mean training error increases slightly as the weight distribution broadens to minimize the KL divergence. The SGD solution is close to mean of the SNN as measured with respect to the SNN covariance. (See Section 2.8 for a discussion.) For the random-label experiment, the SNN mean training error rises above 10%. Ideally, it might have risen to nearly 50%, while driving down the KL term to near zero, finding the optimal Q equal to P.

The empirical test error of the SGD classifiers does not change much across the different architectures, despite the potential for overfitting. This phenomenon is well known, though still remarkable. For the random-label experiment, the empirical test classification error of 0.508 represents lack of generalization, as expected. The same two patterns hold for the SNN test error too, with slightly higher error rates.

Remarkably, the PAC-Bayes bounds do not grow much despite the networks becoming several times larger, and all true label experiments have classification error bounded by 0.23. (This observation is consistent with (Neyshabur, Tomioka, and Srebro, 2014).) Since larger networks possess many more symmetries, the true PAC-Bayes bounds for our learned stochastic neural network classifier might be substantially smaller. (See Section 2.5 for a discussion.) While these bounds are several times larger than the test error estimated

Algorithm 1 PAC-Bayes bound optimization by SGD

Input:

$w_0 \in \mathbb{R}^d$ ▷ Network parameters (random init.)

$w \in \mathbb{R}^d$ ▷ Network parameters (SGD solution)

S_m ▷ Training examples

$\delta \in (0,1)$ ▷ Confidence parameter

$b \in \mathbb{N}, c \in (0,1)$ ▷ Precision and bound for λ

$\tau \in (0,1), T$ ▷ Learning rate; # of iterations

Output: Optimal w, s, λ ▷ Weights, variances

procedure PAC-BAYES-SGD

  $\varsigma \leftarrow \mathrm{abs}(w)$ ▷ where $s(\varsigma) = \mathrm{e}^{2\varsigma}$

  $\rho \leftarrow -3$ ▷ where $\lambda(\rho) = \mathrm{e}^{2\rho}$

  $\mathrm{B}(w, s, \lambda, w') := \breve{R}_{S_m}(w') + \sqrt{\frac{1}{2} B_{\mathrm{RE}}(w, s, \lambda)}$

  for $t \leftarrow 1, T$ **do** ▷ Run SGD for T iterations.

    Sample $\xi \sim \mathcal{N}(0, I_d)$

    $w'(w, \varsigma) = w + \xi \odot \sqrt{s(\varsigma)}$

    ▷ Gradient step

$$\begin{bmatrix} w \\ \varsigma \\ \rho \end{bmatrix} \mathrel{-}= \tau \begin{bmatrix} \nabla_w B(w, s(\varsigma), \lambda(\rho), w'(w, \varsigma)) \\ \nabla_\varsigma B(w, s(\varsigma), \lambda(\rho), w'(w, \varsigma)) \\ \nabla_\rho B(w, s(\varsigma), \lambda(\rho), w'(w, \varsigma)) \end{bmatrix}$$

  return $w, s(\varsigma), \lambda(\rho)$

on held-out data (approximately, 0.03), they demonstrate nontrivial generalization. The PAC-Bayes bound for the random-label experiment is vacuous.

The VC-dimension upper bounds indicate that data independent bounds will be vacuous by several orders of magnitude. Because the number of parameters exceeds the available training data, lower bounds imply that generalization cannot be explained in a data independent way.

2.8 Comparing weights before and after PAC-Bayes optimization

In the course of optimizing the PAC-Bayes bound, we allow the mean w to deviate from the SGD solution w_{SGD} that serves as the starting point. This is necessary to obtain bounds as

tight as those that we computed. Do the weights change much during optimization of the bound? How would we measure this change?

To answer these questions, we calculated the p-value of the SGD solution under the distribution of the stochastic neural network.

Let Q_{SNN} denote the distribution obtained by optimizing the PAC-Bayes bound, write w_{SNN} and Σ_{SNN} for its mean and covariance, and let $\|w\|_{\Sigma_{\mathrm{SNN}}} = w^T \Sigma_{\mathrm{SNN}}^{-1} w$ denote the induced norm. Using 10000 samples, we estimated

$$\mathop{\mathbb{P}}_{w\sim Q_{\mathrm{SNN}}}\Big(\|w - w_{\mathrm{SNN}}\|_{\Sigma_{\mathrm{SNN}}} < \|w_{\mathrm{SGD}} - w_{\mathrm{SNN}}\|_{\Sigma_{\mathrm{SNN}}}\Big).$$

The estimate was 0 for all true label experiments, i.e., w_{SGD} is less extreme of a perturbation of w_{SNN} than a typical perturbation. For the random-label experiments, w_{SNN} and w_{SGD} differ significantly, which is consistent with the bound being optimized in the face of random labels.

2.9 Evaluating Rademacher error bounds

Fix a class $\mathscr{F}$ of measurable functions from $\mathbb{R}^D$ to $\mathbb{R}$ and let $\mathscr{R}_m(\mathscr{F})$ denote the Rademacher complexity of $\mathscr{F}$ associated with m i.i.d. samples, as defined in Section 1.1.3. For $h \in \mathscr{F}$, we will obtain binary classifications (and measure error and empirical error) by computing the sign of its output, i.e., by thresholding. The following error bound is a straightforward adaptation of (Bartlett and Mendelson, 2002, Thm. 7), which is itself an adaptation of (Koltchinskii and Panchenko, 2002, Thm. 2).

Theorem 2.9.1. *For every $L > 0$, with probability at least $1-\delta$ over the choice of $S_m \sim \mathscr{D}^m$, for all $h \in \mathscr{F}$,*

$$R_{\mathscr{D}}(h) \le \hat{R}_S(h,L) + 2L\mathscr{R}_m(\mathscr{F}) + \sqrt{\frac{\log(\frac{2}{\delta})}{2m}}, \tag{2.7}$$

where

$$\hat{R}_S(h,L) = \frac{1}{m}\sum_{i=1}^{m} \max(\min(1 - Ly_i h(x_i), 1), 0).$$

In order to compute these bounds, we must compute (bounds on) the Rademacher complexity of appropriate function classes. To that end, we will use results by Neyshabur,

Tomioka, and Srebro (2015) for ReLU networks (i.e., multilayer perceptrons with ReLU activations).

Let w be the weights of a ReLU network and let $w_{i,j}^{(k)}$ denote the weight associated with the edge from neuron i in layer $k-1$ to neuron j in layer k. Neyshabur, Tomioka, and Srebro (2015) define the ℓ_1 path norm

$$\phi_1(w) = \sum_j \left[|w_{j,1}^{(2)}| \sum_i |w_{i,j}^{(1)}| \right], \tag{2.8}$$

stated here in the special case of a 2-layer network with 1 output neuron. For any number of layers, the path norm can be computed easily in a forward pass, requiring only a matrix–vector product at each layer.

Neyshabur, Tomioka, and Srebro also provide the follow Rademacher bound in terms of the path norm:

Theorem 2.9.2 ((Neyshabur, Tomioka, and Srebro, 2015, Cor. 7))**.** *Given m datapoints $x_1, \ldots, x_m \in \mathbb{R}^D$, the Rademacher complexity of the class of depth-d ReLU networks, whose ℓ_1 path norms are bounded by ϕ, is no greater than*

$$2^d \phi \sqrt{\frac{\log(2D)}{m}} \max_i \|x_i\|_\infty. \tag{2.9}$$

Let $w_j^{(k)}$ for the jth column of $w^{(j)}$, i.e., the vector of weights for edges from layer $k-1$ to neuron j in layer k. The ℓ_1 path norm is closely related to the norm

$$\gamma_{1,\infty}(w) = \prod_{i=1}^{d} \max_j \left\| w_j^{(k)} \right\|_1.$$

If the upper bound ϕ appearing in the bound of Theorem 2.9.2 is instead taken to be a bound on $\gamma_{1,\infty}(w)$, then one essentially obtains the Gaussian complexity bounds for neural networks established by Bartlett and Mendelson (2002) and Koltchinskii and Panchenko (2002). However, their bounds apply only to networks with bounded activation functions, ruling out ReLU networks.

Regardless, the path-norm bound is tighter for ReLU networks. In order to establish the connection, let $\mathscr{W}(w)$ denote the set of all weights w' obtained from redistributing the weights w across layers, i.e., by multiplying the weights $w^{(k-1)}$ in a layer by a constant $c > 0$ and multiplying the weights in the subsequent layer $w^{(k)}$ by c^{-1}. Note that the function computed by a ReLU network is invariant to this transformation. This is the key insight of Neyshabur, Tomioka, and Srebro. Obviously, $\phi_1(w) = \phi_1(w')$ for all $w' \in \mathscr{W}(w)$. Neyshabur,

Tomioka, and Srebro show that $\phi_1(w) = \inf_{w' \in \mathscr{W}(w)} \gamma_{1,\infty}(w')$, and so the path norm better captures the complexity of a ReLU network.

In our experiments, we will compute the bound obtained by combining Theorems 2.9.1 and 2.9.2.

Note that the constant L in Theorem 2.9.1 must be chosen independently of the data S_m. As in the original result (Koltchinskii and Panchenko, 2002, Thm. 2), one can use a union bound to allow oneself to choose L based on the data in order to minimize the bound. Even though the effect of this change is usually (relatively) small, its magnitude depends on the particular weight function employed in the union bound. Instead, we will apply the bound with an optimized L, yielding an optimistic bound (formally, a lower bound on any upper bound obtained from a union bound). We optimize L over a grid of values, and handle the vacuous edge cases analytically. Nevertheless, we will see even the resulting (optimistic) bound is vacuous.

2.9.1 Experiment details

We use SGD to train a two-layer 600-hidden-unit ReLU network on the same binary class variant of MNIST used to evaluate our PAC-Bayes bounds. We set the global learning rate to 0.005. As in our PAC-Bayes experiments, we optimize the average logistic loss during training. The random initializations commonly used for ReLU networks lead to initial path norms that produce vacuous error bounds. In order to visualize the behavior of the path-norm bound under SGD, we reduce the standard deviation of the truncated-normal initialization from 0.04 to 0.0001. As before, we use mini-batches of 100 training examples, yielding 550 iterations per epoch.

For comparison, we also train the same network architecture while explicitly regularizing the path norm. This is similar to the work done by Neyshabur, Salakhutdinov, and Srebro (2015).

2.9.2 Results

When the network is trained by optimizing the logistic cost function without regularization, the error bound becomes vacuous within a fraction of a single epoch. This occurs before the training error dips appreciable below chance. The bound's behavior is due to the path norm diverging. While the level sets $\hat{R}_S(h,\cdot)^{-1}$ of the empirical margin distribution are growing, they are not growing fast enough to counteract the growth of the path norm. (See the left column of Fig. 2.1.)

When the network is trained with explicit path-norm regularization, we obtain vacuous error bounds, unless we apply excessive amounts of regularization. We report results when the regularization parameter is 0.01 and 0.05. Both settings are clearly too large, as evidenced by the training error converging to ~20% and ~30%, respectively. A cursory study of overall ℓ_1 and ℓ_2 regularization produced qualitatively similar results. Further study is necessary.

2.10 Related work

As we mention in the introduction, our approach scales the ideas in (Hinton and Camp, 1993) and (Langford and Caruana, 2002a) to the modern deep learning regime where the networks have millions of parameters, but are trained on one or two orders of magnitude fewer training examples. The objective we optimize is an upper bound on the PAC-Bayes bound, which we know from the discussion in Section 2.2.1 will be very loose when the empirical classification error is approximately zero. Indeed, in that case, the PAC-Bayes bound is approximately

$$\hat{R}_{S_m}(\mathscr{N}_{w,s}) + \frac{\mathrm{KL}(\mathscr{N}_{w,s}||P) + \log\frac{m}{\delta}}{(m-1)}. \tag{2.10}$$

The objective optimized by Hinton and Camp is of the same essential form as this one, except for the choice of squared error and different prior and posterior distributions. We explored using Eq. (2.10) as our objective with a surrogate loss, but it did not produce different results.

In the introduction we discuss the close connection of our work to several recent papers (Baldassi et al., 2015; Baldassi et al., 2016; Chaudhari et al., 2017) that study "flat" or nonisolated minima on the account of their generalization and/or algorithmic properties.

Based on theoretical results for k-SAT that efficient algorithms find nonisolated solutions, Baldassi et al. (2016) model efficient neural network learning algorithms as minimizers of a *replicated* version of the empirical loss surface, which emphasizes nonisolated minima and deemphasizes isolated minima. They then propose several algorithms for learning discrete neural networks using these ideas.

In follow-up work with Chaudhari, Choromanska, Soatto, and LeCun (Chaudhari et al., 2017), they translate these ideas into the setting of continuously parametrized neural networks. They introduce an algorithm, called Entropy-SGD, which seeks out large regions of dense local minima: it maximizes the depth and flatness of the energy landscape. Their objective integrates both the energy of nearby parameters and the weighted distance to the parameters. In particular, rather than directly minimizing an error surface $w \mapsto \hat{R}_{S_m}(\mathbf{w})$, they propose the

following minimization problem over the so-called local-entropic loss [2]:

$$\min_{w\in\mathbb{R}^d} \quad \log \mathop{\mathbb{E}}_{W\sim\mathcal{N}_{w,1/\gamma}} [C(\gamma)\exp\{-\hat{R}_{S_m}(W)\}], \tag{2.11}$$

where $\gamma > 0$ is a parameter and $C(\gamma)$ a constant. In comparison, our algorithm can be interpreted as an optimization of the form

$$\min_{w\in\mathbb{R}^p, s\in\mathbb{R}^p_+} \quad \mathop{\mathbb{E}}_{W\sim\mathcal{N}_{w,s}} [\hat{R}_{S_m}(W)] + L(w,s) \tag{2.12}$$

where L serves as a regularizer that accounts for the generalization error by, roughly speaking, trying to expand the axis-aligned ellipsoid $\{x \in \mathbb{R}^d : (w-x)^T \mathrm{diag}(s)^{-1}(w-x) = 1\}$ and draw it closer to some point w_0 near the origin. Comparing Eqs. (2.11) and (2.12) highlights similarities and differences. The local-entropic loss is sensitive to the volume of the regions containing good solutions. While the first term in our objective function looks similar, it does not, on its own, account for the volume of regions. This role is played by the second term, which prefers large regions (but also ones near the initialization w_0). In our formulation, the first term is the empirical error of a stochastic neural network, which is precisely the term whose generalization error we are trying to bound. Entropy-SGD was not designed for the purpose of finding good stochastic neural networks, although it seems possible that having small local-entropic loss would lead to generalization for neural networks whose parameters are drawn from the local Gibbs distribution. Another difference is that, in our formulation, the diagonal covariance of the multivariate normal perturbation is learned adaptively, and driven by the goal of minimizing error. The shape of the normal perturbation is not learned, although the region whose volume is being measured is determined by the error surface, and it seems likely that this volume will be larger than that spanned by a multivariate Gaussian chosen to lie entirely in a region with good loss.

Chaudhari et al. (2017) give an informal characterization of the generalization properties of local-entropic loss in Bayesian terms by comparing the marginal likelihood of two Bayesian priors centered at a solution with small and large local-entropic loss. Informally, a Bayesian prior centered on an isolated solution will lead to small marginal likelihood in contrast to one centered in a wide valley. They give a formal result relying on the uniform stability of SGD (Hardt, Recht, and Singer, 2015) to show under some strong (and admittedly unrealistic) conditions that Entropy-SGD generalizes better than SGD. The key property is that the local-entropic loss surface is smoother than the original error surface.

[2] A detailed analysis of local entropy and Entropy-SGD algorithm is done in Chapter 4

Other authors have found evidence of the importance of "flat" minima: Recent work by Keskar, Mudigere, Nocedal, Smelyanskiy, and Tang (Keskar et al., 2017) finds that large-batch methods tend to converge to sharp / isolated minima and have worse generalization performance compared to mini-batch algorithms, which tend to converge to flat minima and have good generalization performance. The bulk of their paper is devoted to the problem of restoring good generalization behavior to batch algorithms.

Finally, our algorithm also bears resemblance to *graduated optimization*, an approach toward non-convex optimization attributed to Blake and Zisserman (1987) whereby a sequence of increasingly fine-grained versions of an optimization problem are solved in succession. (See (Hazan, Levy, and Shalev-Shwartz, 2016) and references therein.) In this context, Eq. (2.11) is the result of a local smoothing operation acting on the objective function $w \mapsto \breve{\ell}(\mathbf{w}, S_M)$. In graduate optimization, the effect of the local smoothing operation would be decreased over time, eventually disappearing. In our formulation, the act of balancing the empirical loss and generalization error serve to drive the evolution of the local smoothing in an adaptive fashion. Moreover, in the limit, the local smoothing does not vanish in our algorithm, as the volume spanned by the perturbations relates to the generalization error. Our results suggest that SGD solutions live inside relatively large volumes, and so perhaps SGD can be understood in terms of graduated optimization.

2.11 Discussion

We obtain nonvacuous generalization bounds for deep neural networks with millions of parameters trained on 55000 MNIST examples. These bounds are obtained by optimizing an objective derived from the PAC-Bayes bound, starting from the solution produced by SGD. Despite the weights changing, the SGD solution remains well within the 1% ellipsoidal quantile, i.e., the volume spanned by the stochastic neural network contains the original SGD solution. (When labels are randomized, however, optimizing the PAC-Bayes bound causes the solution to shift considerably.)

Our experiments look only at fully connected feed forward networks trained on a binary class variant of MNIST. It would be interesting to see if the results extend to multiclass classification, to other data sets, and to other types of architectures, especially convolutional ones.

Our PAC-Bayes bound can be tightened in several ways. Highly dependent weights constrain the size of the axis-aligned ellipsoid representing the stochastic neural network. We can potentially recognize small populations of highly dependent weights, and optimize their covariance parameters, rather than enforcing independence in the posterior.

One might also consider replacing the multivariate normal posterior with a distribution that is more tuned to the loss surface. One promising avenue is to follow the lines of Chaudhari et al. (2017) and consider (local) Gibbs distributions. If the solutions obtained by minimizing the local-entropic loss are flatter than those obtained by SGD, than we may be able to demonstrate quantitatively tighter bounds.

Finally, there is the hard work of understanding the generalization properties of SGD. In light of our work, it may be useful to start by asking whether SGD finds solutions in flat minima. Such solutions could then be lifted to stochastic neural networks with good generalization properties. Going from stochastic networks back to deterministic ones may require additional structure.

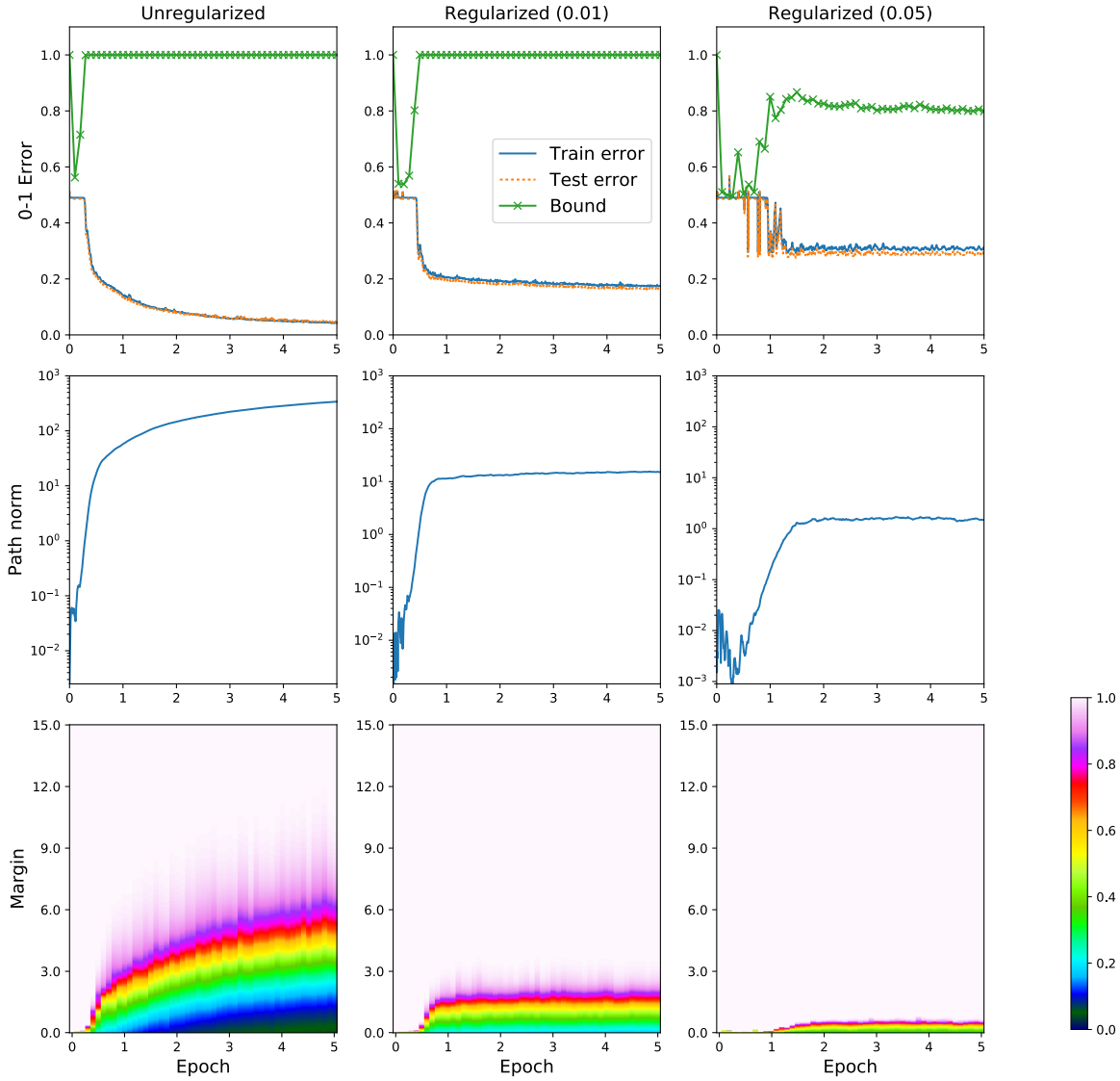


Fig. 2.1 Unregularized (**left column**) and path-norm regularized (**center and right columns** with regularization parameter specified in parenthesis) optimization of two-layer 600-hidden-unit ReLU network by SGD for 5 epochs. (We ran 20 epochs and found no new patterns. Plots for longer experiment obscured the initial behavior.) (**top row**) Training error, testing error, and error bounds versus (iterations measured in) epochs. Without regularization, the bounds are immediately vacuous once the network performance deviates from chance, and this remains true under regularization unless the explicit regularization is very strong. In this case, the bound is nonvacuous, but trivial in the sense that the error rate of guessing is 50%. Note that training/testing error is also very large in this case. (**center row**) Log plot of path norm versus epochs. Without regularization, the path norm diverges quickly. (**bottom row**) Empirical margin distributions versus epochs. The margin that attains a fixed average loss is growing, but not rapidly enough to counteract the rapidly increasing path norm.

Chapter 3

Data-dependent PAC-Bayes bounds

3.1 Introduction

There has been a resurgence of interest in PAC-Bayes bounds, especially towards explaining generalization in large-scale neural networks trained by stochastic gradient descent (Neyshabur et al., 2017b; Neyshabur et al., 2017a). See also (Bégin et al., 2016; Germain et al., 2016; Thiemann et al., 2017; Bartlett, Foster, and Telgarsky, 2017b; Raginsky, Rakhlin, and Telgarsky, 2017; Grünwald and Mehta, 2017; Smith and Le, 2018).

PAC-Bayes bounds control the generalization error of Gibbs classifiers (aka PAC-Bayes "posteriors") in terms of the Kullback–Leibler (KL) divergence to a fixed probability measure (aka PAC-Bayes "prior") on the space of classifiers. PAC-Bayes bounds depend on a tradeoff between the empirical risk of the posterior Q and a penalty $\frac{1}{m}\mathrm{KL}(Q||P)$, where P is the prior, fixed independently of the sample $S \in Z^m$ from some space Z of labelled examples. The KL penalty is typically the largest contribution to the bound and so finding the tightest possible bound generally depends on minimizing the KL term.

The KL penalty vanishes for $Q = P$, but typically P, viewed as a randomized (Gibbs) classifier, has poor performance since it has been chosen independently of the data. On the other hand, since P is chosen independently of the data, posteriors Q tuned to the data to achieve minimal empirical risk often bear little resemblance to the data-independent prior P, causing $\mathrm{KL}(Q||P)$ to be large. As a result, PAC-Bayes bounds can be loose or even vacuous.

The problem of excessive KL penalties is *not* inherent to the PAC-Bayes framework. Indeed, the PAC-Bayes theorem permits one to choose the prior P based on the distribution $\mathscr{D}$ of the data. However, since $\mathscr{D}$ is considered unknown, and our only insight as to $\mathscr{D}$ is through the sample S, this flexibility would seem to be useless, as P must be chosen independently of S in existing bounds. Nevertheless, it is possible to make progress in this direction, and it is likely the best way towards tighter bounds and deeper understanding.

There is a growing body of work in the PAC-Bayes literature on data-distribution-dependent priors (Catoni, 2007; Parrado-Hernández et al., 2012; Lever, Laviolette, and Shawe-Taylor, 2013). Our focus is on generalization bounds that use all the data, and in this setting, Lever, Laviolette, and Shawe-Taylor (2013) prove a remarkable result for Gibbs posteriors. Writing $\hat{R}_S$ for the empirical risk function with respect the sample S, Lever, Laviolette, and Shawe-Taylor study the randomized classifier Q with density (relative to some base measure) proportional to $\exp(-\tau\hat{R}_S)$. For large values of τ, Q concentrates increasingly around the empirical risk minimizers. As a prior, the authors consider the probability distribution with density proportional to $\exp(-\tau R_{\mathscr{D}})$, where the empirical risk has been replaced by its expectation, $R_{\mathscr{D}}$, the (true) risk. Remarkably, Lever, Laviolette, and Shawe-Taylor are able to upper bound the KL divergence between the two distributions by a term independent of $\mathscr{D}$, yielding the following result: Here $\mathrm{kl}(q||p)$ is the KL divergence between Bernoulli measures with mean q and p. See Section 3.3 for more details.

Theorem 3.1.1 (Lever, Laviolette, and Shawe-Taylor 2013)**.** *Fix $\tau > 0$. For $S \in Z^m$, let $Q(S) = P_{\exp(-\tau\hat{R}_S)}$ be a Gibbs posterior with respect to some base measure P on $\mathbb{R}^p$, where the empirical risk $\hat{R}_S$ is bounded in $[0,1]$. For every $\delta > 0$, $m \in \mathbb{N}$, distribution $\mathscr{D}$ on Z,*

$$\mathop{\mathbb{P}}_{S\sim\mathscr{D}^m}\left(\mathrm{kl}(\hat{R}_S(Q(S))||R_{\mathscr{D}}(Q(S))) \leq \frac{1}{m}\left(\tau\sqrt{\frac{2}{m}\ln\frac{2\sqrt{m}}{\delta} + \frac{\tau^2}{2m} + \ln\frac{2\sqrt{m}}{\delta}}\right)\right) \geq 1-\delta. \quad (3.1)$$

The dependence on the data distribution is captured through τ, which is ideally chosen as small as possible, subject to $Q(S)$ yielding small empirical risk. (One can use a union bound to tune τ based on S.) The fact that the KL bound does not depend on $\mathscr{D}$, other than through τ, implies that the bound *must* be loose for all τ such that there exists a distribution $\mathscr{D}$ that causes Q to overfit with high probability on size m datasets $S \sim \mathscr{D}^m$. In other words, for fixed τ, the bound is no longer distribution dependent. This would not be important if not for the following empirical finding: weights sampled according to high values of τ do not overfit on real data, but they do on data whose labels have been randomized. Thus these bounds are vacuous in practice when the generalization error is, in fact, small. Evidently, the KL bound gives up too much.

Our work launches a different attack on the problem of using distribution-dependent priors. Loosely speaking, if a prior is chosen on the basis of the data, but in a way that is very stable to perturbations of the data set, then the resulting data-dependent prior should reflect the underlying data distribution, rather than the data, resulting in a bound that should still hold, perhaps with smaller probability.

We formalize this intuition using differential privacy (Dwork, 2006; Dwork et al., 2015b). We show that an ε-differentially private prior mean yields a valid, though necessarily looser,

PAC-Bayes generalization bound. (See Theorem 3.3.2.) The result is a straightforward application of results connecting privacy and adaptive data analysis (Dwork et al., 2015b; Dwork et al., 2015a).

The real challenge is using such a result: In practice, ε-differentially private mechanisms are expensive to compute. In the context of generalization bounds for neural networks, we consider the possibility of using stochastic gradient Langevin dynamics (SGLD; Welling and Teh, 2011) to choose a data-dependent prior by way of stochastic optimization/sampling.

By various results, SGLD is known to produce an (ε, δ)-differentially private release (Mir, 2013; Bassily, Smith, and Thakurta, 2014; Dimitrakakis et al., 2014; Wang, Fienberg, and Smola, 2015; Minami et al., 2016) A gap remains between pure and approximate differential privacy. Even if this gap were to be closed, the privacy/accuracy tradeoff of these analyses is too poor because they do not take advantage of the fact that, under some technical conditions, the distribution of SGLD's output converges weakly towards a stationary distribution (Teh, Thiery, and Vollmer, 2016, Thm. 7), which is ε-differentially private. One can also bound the KL divergence (and then 2-Wasserstein distance) of SGLD to stationarity within a constant given an appropriate fixed step size (Raginsky, Rakhlin, and Telgarsky, 2017). Neither result implies that SGLD achieves pure ε-differential privacy.

We show that we can circumvent this barrier in our PAC-Bayes setting. We give a general result for non-private data-dependent priors and then an application to multivariate Gaussian priors with non-private data-dependent means, with explicit bounds for the case of Gibbs posteriors. In particular, conditional on a data set, if a data-dependent mean vector $\mathbf{w}$ is close in 2-Wasserstein distance to an ε-differentially private mean vector, then the generalization errors is close to that of the ε-differentially private mean. Neither the data-dependent mean $\mathbf{w}$, *nor its image* $\mathcal{N}(\mathbf{w}, \Sigma)$, is necessarily differentially private, even approximately. As an application of this result, SGLD can be used to optimize a data-dependent prior and still yield a valid PAC-Bayes bound.

3.2 Other Related Work

Our analysis relies on the stability of a data-dependent prior. Stability has long been understood to relate to generalization (Bousquet and Elisseeff, 2002). Our result relies on the connection between generalization and differential privacy (Dwork et al., 2015b; Dwork et al., 2015a; Bassily et al., 2016; Oneto, Ridella, and Anguita, 2017), which can be viewed as a particularly stringent notion of stability. See (Dwork, 2008b) for a survey of differential privacy.

Kifer, Smith, and Thakurta (2012, Thm. 1) also establish a "limit" theorem for differential privacy, showing that the almost sure convergence of mechanisms of the same privacy level admits a private mechanism of the same privacy level. Our result can be viewed as a significant weakening of the hypothesis to require only that the weak limit be private: no element on the sequence need be private.

The bounds we establish hold for bounded loss functions and i.i.d. data. Under additional assumptions, one can obtain PAC-Bayes generalization and excess risk bounds for unbounded loss with heavy tails (Catoni, 2007; Germain et al., 2016; Grünwald and Mehta, 2016; Alquier and Guedj, 2018). Alquier and Guedj (2018) also consider non-i.i.d. training data. Our approach to differentially private data-dependent priors can be readily extended to these settings.

3.2.1 Differential privacy

See Section 1.4 for basic definitions for differential privacy and our notation.

For our purposes, *max-information* is the key quantity controlled by differential privacy.

Definition 3.2.1 (Dwork et al. 2015a, §3)**.** Let $\beta \geq 0$, let X and Y be random variables in arbitrary measurable spaces, and let X' be independent of Y and equal in distribution to X. The *β-approximate max-information between X and Y*, denoted $I_\infty^\beta(X;Y)$, is the least value k such that, for all product-measurable events E,

$$\mathbb{P}\{(X,Y) \in E\} \leq \mathrm{e}^k \, \mathbb{P}\{(X',Y) \in E\} + \beta. \tag{3.2}$$

The *max-information* $I_\infty(X;Y)$ is defined to be $I_\infty^\beta(X;Y)$ for $\beta = 0$. For $m \in \mathbb{N}$ and $\mathscr{A} : Z^m \rightsquigarrow T$, the *$\beta$-approximate max-information of $\mathscr{A}$*, denoted $I_\infty^\beta(\mathscr{A},m)$, is the least value k such that, for all $\mathscr{D} \in \mathscr{M}_1(Z)$, $I_\infty^\beta(S;\mathscr{A}(S)) \leq k$ when $S \sim \mathscr{D}^m$. The β-approximate max-information of $\mathscr{A}$ is defined similarly.[1]

In Section 3.3.1, we consider the case where the dataset S and a data-dependent prior $\mathscr{P}(S)$ have small approximate max-information. The above definition tells us that we can almost treat the data-dependent prior as if it was chosen independently from S. The following is the key result connecting pure differential privacy and max-information:

Theorem 3.2.2 (Dwork et al. 2015a, Thms. 19–20)**.** *Fix $m \in \mathbb{N}$. Let $\mathscr{A} : Z^m \rightsquigarrow T$ be ε-differentially private. Then $I_\infty(\mathscr{A},m) \leq \varepsilon m$ and, for all $\beta > 0$, $I_\infty^\beta(\mathscr{A},m) \leq \varepsilon^2 m/2 + \varepsilon\sqrt{m\ln(2/\beta)/2}$.*

[1] Note that in much of the literature it is standard to express the max-information in *bits*, i.e., the factor e^k above is replaced by $2^{k'}$ with $k' = k\log_2 e$.

3.3 PAC-Bayes bounds

We repeat the following PAC-Bayes bound for convenience. The following PAC-Bayes bound for bounded loss is due to Maurer (2004). The same result for 0–1 loss was first established by Langford and Seeger (2001), building off the seminal work of McAllester (1999a). See also (Langford, 2002) and (Catoni, 2007).

Theorem 3.3.1 (PAC-Bayes; Maurer 2004, Thm. 5)**.** *Under bounded loss $\ell \in [0,1]$, for every $\delta > 0$, $m \in \mathbb{N}$, distribution $\mathscr{D}$ on Z, and distribution P on $\mathbb{R}^p$,*

$$\mathop{\mathbb{P}}_{S\sim\mathscr{D}^m}\Big((\forall Q)\ \mathrm{kl}(\hat{R}_S(Q)||R_{\mathscr{D}}(Q)) \le \frac{\mathrm{KL}(Q||P) + \ln\frac{2\sqrt{m}}{\delta}}{m}\Big) \ge 1-\delta. \tag{3.3}$$

One can use Pinsker's inequality to obtain a bound on the generalization error $|\hat{R}_S(Q) - R_{\mathscr{D}}(Q)|$, however this significantly loosens the bound, especially when $\hat{R}_S(Q)$ is close to zero. We refer to the quantity $\mathrm{kl}(\hat{R}_S(Q)||R_{\mathscr{D}}(Q))$ to the *KL-generalization error*. From a bound on this quantity, we can bound the risk as follows: given empirical risk q and a bound on the KL-generalization error c, the risk is bounded by the largest value $p \in [0,1]$ such $\mathrm{kl}(q||p) \le c$. See Chapter 2 for a discussion of this computation. When the empirical risk is near zero, the KL-generalization error is essentially the generalization error. As empirical risk increases, the bound loosens and the square root of the KL-generalization error bounds the generalization error.

3.3.1 Data-dependent priors

The prior P that appears in the PAC-Bayes generalization bound must be chosen independently of the data $S \sim \mathscr{D}^m$, but can depend on the data distribution $\mathscr{D}$ itself. If a data-dependent prior $\mathscr{P}(S)$ does not depend too much on any individual data point, it should behave as if it depends only on the distribution. Theorem 3.2.2 allows us to formalize this intuition: we can obtain new PAC-Bayes bounds that use data-dependent priors, provided they are ε-differentially private: We provide an example using the bound of Maurer (Theorem 3.3.1).

Theorem 3.3.2. *Fix a bounded loss $\ell \in [0,1]$. Let $m \in \mathbb{N}$, let $\mathscr{P}\colon Z^m \rightsquigarrow \mathscr{M}_1(\mathbb{R}^p)$ be an ε-differentially private data-dependent prior, let $\mathscr{D} \in \mathscr{M}_1(Z)$, and let $S \sim \mathscr{D}^m$. Then, with probability at least $1-\delta$,*

$$\forall Q \in \mathscr{M}_1(\mathbb{R}^p),\ \mathrm{kl}(\hat{R}_S(Q)||R_{\mathscr{D}}(Q)) \le \frac{\mathrm{KL}(Q||\mathscr{P}(S)) + \ln\frac{4\sqrt{m}}{\delta}}{m} + \varepsilon^2/2 + \varepsilon\sqrt{\frac{\ln(4/\delta)}{2m}}. \tag{3.4}$$

See Section 3.3.1 for a proof of a more general statement of the theorem and further discussion. The main innovation here is recognizing the potential to choose data-dependent priors using private mechanisms. The hard work is done by Theorem 3.2.2: obtaining differentially private versions of other PAC-Bayes bounds is straightforward.

When one is choosing the privacy parameter, ε, there is a balance between minimizing the direct contributions of ε to the bound (forcing ε smaller) and minimizing the indirect contribution of ε through the KL term for posteriors Q that have low empirical risk (forcing ε larger). The optimal value for ε is often much less than one, which can be challenging to obtain. We discuss strategies for achieving the required privacy in later sections.

Proof of Theorem 3.3.2

We prove a slightly more general result the the one stated in Theorem 3.3.2.

Theorem 3.3.3. *Fix a bounded loss $\ell \in [0,1]$. Let $m \in \mathbb{N}$, let $\mathscr{P}\colon Z^m \rightsquigarrow \mathscr{M}_1(\mathbb{R}^p)$ be an ε-differentially private data-dependent prior, let $\mathscr{D} \in \mathscr{M}_1(Z)$, and let $S \sim \mathscr{D}^m$. Then, for all $\delta \in (0,1)$ and $\beta \in (0,\delta)$, with probability at least $1-\delta$,*

$$\forall Q \in \mathscr{M}_1(\mathbb{R}^p),\ \mathrm{kl}(\hat{R}_S(Q)||R_{\mathscr{D}}(Q)) \leq \frac{\mathrm{KL}(Q||\mathscr{P}(S)) + \ln\frac{2\sqrt{m}}{\delta-\beta}}{m} + \varepsilon^2/2 + \varepsilon\sqrt{\frac{\ln(2/\beta)}{2m}}. \tag{3.5}$$

Proof. For every distribution P on $\mathbb{R}^p$, let

$$R(P) = \left\{S \in Z^m : (\exists Q)\ \mathrm{kl}(\hat{R}_S(Q)||R_{\mathscr{D}}(Q)) \geq m^{-1}\left(\mathrm{KL}(Q||P) + \ln\frac{2\sqrt{m}}{\delta'}\right)\right\}. \tag{3.6}$$

It follows from Theorem 3.3.1 that $\mathbb{P}_{S\sim\mathscr{D}^m}(S \in R(P)) \leq \delta'$. Let $\beta > 0$. Then, by the definition of approximate max-information, we have

$$\mathbb{P}_{S\sim\mathscr{D}^m}(S \in R(\mathscr{P}(S))) \leq e^{I_\infty^\beta(\mathscr{P};m)} \mathbb{P}_{(S,S')\sim\mathscr{D}^{2m}}(S \in R(\mathscr{P}(S'))) + \beta \tag{3.7}$$

$$\leq e^{I_\infty^\beta(\mathscr{P};m)}\delta' + \beta := \delta. \tag{3.8}$$

We have $\delta' = e^{-I_\infty^\beta(\mathscr{P};m)}(\delta-\beta)$. Therefore, with probability no more than δ over $S \sim \mathscr{D}^m$,

$$\exists Q \in \mathscr{M}_1(\mathbb{R}^p),\ \mathrm{kl}(\hat{R}_S(Q)||R_{\mathscr{D}}(Q)) \geq \frac{\mathrm{KL}(Q||\mathscr{P}(S)) + \ln\frac{2\sqrt{m}}{\delta-\beta} + I_\infty^\beta(\mathscr{P};m)}{m}. \tag{3.9}$$

The result follows from replacing the approximate max-information $I_\infty^\beta(\mathscr{P};m)$ with the bound provided by Theorem 3.2.2. □

The theorem leaves open the choice of $\beta < \delta$. For any fixed values for ε, m, and δ, it is easy to optimize β to obtain the tighest possible bound. In practice, however, the optimal bound is almost indistinguishable from that obtained by taking $\beta = \delta/2$. For the remainder of the chapter, we take this value for β, in which case, the r.h.s. of Eq. (3.4) is

$$\frac{\mathrm{KL}(Q||\mathscr{P}(S)) + \ln \frac{4\sqrt{m}}{\delta}}{m} + \varepsilon^2/2 + \varepsilon\sqrt{\frac{\ln(4/\delta)}{2m}}. \tag{3.10}$$

Note that the bound holds for all posteriors Q. In general the bounds are interesting only when Q is data dependent, otherwise one can obtain tighter bounds via concentration of measure results for empirical means of bounded i.i.d. random variables.

When one is choosing the privacy parameter, ε, there is a balance between minimizing the direct contributions of ε to the bound (forcing ε smaller) and minimizing the indirect contribution of ε through the KL term for posteriors Q that have low empirical risk (forcing ε larger). One approach is to compute the value of ε that achieves a certain bound on the excess generalization error. In particular, choosing $\varepsilon^2/2 = \alpha$ contributes an additional gap of α to the KL-generalization error. Choosing α is complicated by the fact that there is a non-linear relationship between the generalization error and the KL-generalization error, depending on the empirical risk. A better approach is often to attempt to balance the direct contribution with the indirect one. Regardless, the optimal value for ε is much less than one, which can be challenging to obtain. We discuss strategies for achieving the required privacy in later sections.

3.4 Weak approximations to ε-differentially private priors

Theorem 3.3.2 permits data-dependent priors that are chosen by ε-differentially private mechanisms. In this section, we discuss concrete families of priors and mechanisms for choosing among them in data-dependent ways. We also relax Theorem 3.3.2 to allow non-private mechanism.

We apply our main result to non-private data-dependent Gaussian priors with a fixed covariance matrix. Thus, we choose only the mean $\mathbf{w}_0 \in \mathbb{R}^p$ privately in a data-dependent way. We show that it suffices for the data-dependent mean to be merely close in 2-Wasserstein distance to a private mean to yield a generalization bound. (It is natural to consider also choosing a data-dependent covariance, but as it is argued below, the privacy budget we have in applications to generalization is very small.)

Ideally, we would choose a mean vector $\mathbf{w}_0$ that leads to a tight bound. A reasonable approach is to choose $\mathbf{w}_0$ by approximately minimizing the empirical risk $\hat{R}_S$ or surrogate risk

$\check{R}_S$, subject to privacy constraints. A natural way to do this is via an exponential mechanism. We pause to introduce some notation for Gibbs distributions: For a measure P on $\mathbb{R}^p$ and measurable function $g : \mathbb{R}^p \to \mathbb{R}$, let $P[g]$ denote the expectation $\int g(h)P(\mathrm{d}h)$ and, provided $P[g] < \infty$, let P_g denote the probability measure on $\mathbb{R}^p$, absolutely continuous with respect to P, with Radon–Nikodym derivative $\frac{\mathrm{d}P_g}{\mathrm{d}P}(h) = \frac{g(h)}{P[g]}$. A distribution of the form $P_{\exp(-\tau g)}$ is generally referred to as a Gibbs distribution with energy function g and inverse temperature τ. In the special case where P is a probability measure, we call $P_{\exp(-\tau \check{R}_S)}$ a "Gibbs posterior".

Lemma 3.4.1 (McSherry and Talwar 2007, Thm. 6)**.** *Let $q : Z^m \times \mathbb{R}^p \to \mathbb{R}$ be measurable, let P be a σ-finite measure on $\mathbb{R}^p$, let $\beta > 0$, and assume $P[\exp(-\beta\, q(S,\cdot))] < \infty$ for all $S \in Z^m$. Let $\Delta q := \sup_{S,S'} \sup_{\mathbf{w}\in\mathbb{R}^p} |q(S,\mathbf{w}) - q(S',\mathbf{w})|$, where the first supremum ranges over pairs $S, S' \in Z^m$ that disagree on no more than one coordinate. Let $\mathscr{A} : Z^m \rightsquigarrow \mathbb{R}^p$, on input $S \in Z^m$, output a sample from the Gibbs distribution $P_{\exp(-\beta q(S,\cdot))}$. Then $\mathscr{A}$ is $2\beta\Delta q$-differentially private.*

The following result is a straightforward application of Lemma 3.4.1 and essentially equivalent results have appeared in numerous studies of the differential privacy of Bayesian and Gibbs posteriors (Mir, 2013; Bassily, Smith, and Thakurta, 2014; Dimitrakakis et al., 2014; Wang, Fienberg, and Smola, 2015; Minami et al., 2016).

Corollary 3.4.2. *Let $\tau > 0$ and let $\check{R}_S$ denote the surrogate risk, taking values in an interval of length Δ. One sample from the Gibbs posterior $P_{\exp(-\tau \check{R}_S)}$ is $\frac{2\tau\Delta}{m}$-differentially private.*

3.4.1 Weak convergence yields valid PAC-Bayes bounds

Even for small values of the inverse temperature, it is difficult to implement the exponential mechanism because sampling from Gibbs posteriors exactly is intractable. On the other hand, a number of algorithms exist for generating approximate samples from Gibbs posteriors. If one can control the total-variation distance, one can obtain a bound like Theorem 3.3.2 by applying approximate max-information bounds for (ε, δ)-differential private mechanisms. However, many algorithms do not control the total-variation distance to stationarity.

The generalization properties of randomized classifiers are generally insensitive to small variations of the parameters, however, and so it stands to reason that our data-dependent prior need only be itself close to an ε-differentially private prior. We formalize this intuition here by deriving bounds on $\mathrm{KL}(Q||\mathscr{P}(S))$ in terms of a non-private data-dependent prior P^S. We start with an identity:

Lemma 3.4.3. *If $P' \ll P$ then $\mathrm{KL}(Q||P) = \mathrm{KL}(Q||P') + Q[\ln \frac{\mathrm{d}P'}{\mathrm{d}P}]$.*

The proof is straightforward, highlights the role of Q in judging the difference between P' and P, and leads immediately to the following corollary (see Section 3.5).

Lemma 3.4.4 (Non-private priors)**.** *Let $m \in \mathbb{N}$, let $\mathscr{P}: Z^m \rightsquigarrow \mathscr{M}_1(\mathbb{R}^p)$ be ε-differentially private, let $\mathscr{D} \in \mathscr{M}_1(Z)$, let $S \sim \mathscr{D}^m$, and let P^S be a data-dependent prior such that, for some $P^*(S)$ satisfying $\mathbb{P}[P^*(S)|S] = \mathbb{P}[\mathscr{P}(S)|S]$, we have $P^S \ll P^*(S)$ with probability at least $1-\delta'$. Then, with probability at least $1-\delta-\delta'$, Eq.* (3.4) *holds with* $\mathrm{KL}(Q||\mathscr{P}(S))$ *replaced by* $\mathrm{KL}(Q||P^S) + Q[\ln \frac{\mathrm{d}P^S}{\mathrm{d}P^*(S)}]$.

The conditions that $P^S \ll P^*(S)$ and $Q[\ln \frac{\mathrm{d}P^S}{\mathrm{d}P^*(S)}] \leq \infty$ for some Q do not constrain P^S to be differentially private. In fact, S could be P^S-measurable!

Lemma 3.4.4 is not immediately applicable because $P^*(S)$ is intractable to generate. The following application considers multivariate Gaussian priors, $N(w)$, indexed by their mean vectors $w \in \mathbb{R}^p$, with a fixed positive definite covariance matrix $\Sigma \in \mathbb{R}^{p\times p}$. We require two technical results: the first implies that we can bound $Q[\ln \frac{\mathrm{d}P^S}{\mathrm{d}P^*(S)}]$ by establishing concentration of Q near the non-private mean. the second characterizes this concentration for Gibbs posteriors built from bounded surrogate risks.

Lemma 3.4.5. $Q[\ln \frac{\mathrm{d}N(w')}{\mathrm{d}N(w)}] \leq \frac{1}{2}\|w-w'\|^2_{\Sigma^{-1}} + \|w-w'\|_{\Sigma^{-1}} \mathop{\mathbb{E}}_{v\sim Q}\|v-w'\|_{\Sigma^{-1}}$.

Lemma 3.4.6. *Let $P = N(w)$ and $Q = P_{\exp(h)}$ for $h \geq 0$. Then* $\mathop{\mathbb{E}}_{v\sim Q}\|v-w\|_{\Sigma^{-1}} \leq \sqrt{2\|h\|_{L^\infty(P)}} + \sqrt{2/\pi}$.

Corollary 3.4.7 (Gaussian means close to private means)**.** *Let $m \in \mathbb{N}$, let $\mathscr{D} \in \mathscr{M}_1(Z)$, let $S \sim \mathscr{D}^m$, let $\mathscr{A}: Z^m \rightsquigarrow \mathbb{R}^p$ be ε-differentially private, and let $w(S)$ denote a data-dependent mean vector such that, for some $w^*(S)$ satisfying $\mathbb{P}[w^*(S)|S] = \mathbb{P}[\mathscr{A}(S)|S]$, we have*

$$\|w(S) - w^*(S)\|_2^2 \leq C \tag{3.11}$$

with probability at least $1-\delta'$. Let $\sigma_{\min}$ be the minimum eigenvalue of Σ. Then, with probability at least $1-\delta-\delta'$, Eq. (3.4) *holds with* $\mathrm{KL}(Q||\mathscr{P}(S))$ *replaced by* $\mathrm{KL}(Q||N(w(S))) + \frac{1}{2}C/\sigma_{\min} + \sqrt{C/\sigma_{\min}}\,\mathbb{E}_{v\sim Q}\|v-w(S)\|_{\Sigma^{-1}}$. *In particular, for a Gibbs posterior $Q = P^S_{\exp(-\tau\check{R}_S)}$, we have $\mathbb{E}_{v\sim Q}\|v-w(S)\|_{\Sigma^{-1}} \leq \sqrt{2\tau\Delta} + \sqrt{2/\pi}$.*

See Section 3.5 for details and further discussion.

One way to achieve Eq. (3.11) is to construct $w_1(S), w_2(S), \ldots$ so that $\mathbb{P}[w_n(S)|S] \to \mathbb{P}[\mathscr{A}(S)|S]$ weakly with high probability. Skorohod's representation theorem then implies the existence of $w^*(S)$. One of the standard algorithms used to generate such sequences

for high-dimensional Gibbs distributions is stochastic gradient Langevin dynamics (SGLD; Welling and Teh, 2011).

A nonasymptotic approach is to control the p-Wasserstein distance, given by $(\mathscr{W}_p(\mu,\nu))^p = \inf_\gamma \int \|v-w\|_2^p \,\mathrm{d}\gamma(v,w)$ where the infimum runs over couplings of μ and ν, i.e., distributions $\gamma \in \mathscr{M}_1(\mathbb{R}^p \times \mathbb{R}^p)$ with marginals μ and ν, respectively.

Some existing results on SGLD control 2-Wasserstein distance to stationarity. One could take a result controlling the 2-Wasserstein distance and combine it with our results to get a guarantee of the following form:

Corollary 3.4.8 (SGLD PAC-Bayes Bound)**.** *Consider SGLD sampling the Gibbs posterior $P_{\exp(-\tau \check{R}_S)}$ with Gaussian P and smooth surrogate risk $\check{R}_S$ taking values in a length-Δ interval. Assume that for every $C > C' > 0$, there is a choice of step size η and number of SGLD iterations n, such that the n-th iterate $w(S) \in \mathbb{R}^p$ produced by SGLD satisfies $\mathscr{W}_2\big(\mathbb{P}[w(S)|S], \mathbb{P}[\mathscr{A}(S)|S]\big) \le C'$, where $\mathscr{A}(S)$ is a $\frac{2\tau\Delta}{m}$-differentially private vector distributed according to the Gibbs posterior given S. By Markov's inequality and the definition of $\mathscr{W}_2$, there exists $w^*(S)$ as in Corollary 3.4.7 such that, with probability $1-\delta'$, $\|w(S) - w^*(S)\|_2^2 \le C/\delta'$.*

The dependence on δ' appears to be poor. However, Markov chain algorithms are often geometrically ergodic, in which case C decays exponentially fast in the number of Markov chain steps, allowing one to spend computation to control the $1/\delta'$ term.

3.5 Proofs for Section 3.4.1

Proof of Lemma 3.4.3. Assume $Q \ll P'$, for otherwise the bound is trivial as $\mathrm{KL}(Q||P') = \infty$. Then $Q \ll P$, because $P' \ll P$, and so $\frac{\mathrm{d}Q}{\mathrm{d}P} = \frac{\mathrm{d}Q}{\mathrm{d}P'}\frac{\mathrm{d}P'}{\mathrm{d}P}$ and

$$\begin{aligned}
\mathrm{KL}(Q||P) - \mathrm{KL}(Q||P') &= Q\Big[\ln \frac{\mathrm{d}Q}{\mathrm{d}P}\Big] - \mathrm{KL}(Q||P') && (3.12)\\
&= Q\Big[\ln \frac{\mathrm{d}Q}{\mathrm{d}P'} + \ln \frac{\mathrm{d}P'}{\mathrm{d}P}\Big] - \mathrm{KL}(Q||P') && (3.13)\\
&= \mathrm{KL}(Q||P') + Q\Big[\ln \frac{\mathrm{d}P'}{\mathrm{d}P}\Big] - \mathrm{KL}(Q||P') && (3.14)\\
&= Q\Big[\ln \frac{\mathrm{d}P'}{\mathrm{d}P}\Big]. && (3.15)
\end{aligned}$$

□

Proof of Lemma 3.4.4. Let $P^*(S)$ satisfy the conditions in the statement of the theorem. Then $P^*(S)$ is ε-differentially private. By Theorem 3.3.2, the bound in Eq. (3.4) holds with

probability at least $1-\delta$ for the data-dependent prior $P^*(S)$ and all posteriors Q. By hypothesis, with probability $1-\delta-\delta'$, $P^S \ll P^*(S)$, and so, by Lemma 3.4.3, $\mathrm{KL}(Q||P^*(S)) = \mathrm{KL}(Q||P^S) + Q[\ln \frac{\mathrm{d}P^S}{\mathrm{d}P^*(S)}]$. □

Proof of Lemma 3.4.5. Expanding the log ratio of Gaussian densities and then applying Cauchy–Schwarz, we obtain

$$\ln \frac{\mathrm{d}N(w')}{\mathrm{d}N(w)}(v) = \frac{1}{2}\left(\|w-v\|_{\Sigma^{-1}}^2 - \|w'-v\|_{\Sigma^{-1}}^2\right) \tag{3.16}$$

$$= \langle w'-w, v\rangle_{\Sigma^{-1}} + \frac{1}{2}\left(\|w\|_{\Sigma^{-1}}^2 - \|w'\|_{\Sigma^{-1}}^2\right) \tag{3.17}$$

$$= \frac{1}{2}\langle w'-w, 2v\rangle_{\Sigma^{-1}} - \frac{1}{2}\langle w'-w, w+w'\rangle_{\Sigma^{-1}} \tag{3.18}$$

$$= \frac{1}{2}\langle w'-w, 2v-w-2w'+w'\rangle_{\Sigma^{-1}} \tag{3.19}$$

$$= \frac{1}{2}\langle w'-w, 2(v-w')+w'-w\rangle_{\Sigma^{-1}} \tag{3.20}$$

$$\leq \frac{1}{2}\|w'-w\|_{\Sigma^{-1}}^2 + \|w'-w\|_{\Sigma^{-1}}\,\|v-w'\|_{\Sigma^{-1}}. \tag{3.21}$$

The result follows by taking the expectation with respect to $v \sim Q$. □

Proof of Lemma 3.4.6. Let $g = \frac{\mathrm{d}Q}{\mathrm{d}P} := \frac{\mathrm{e}^h}{P[\mathrm{e}^h]}$. Then $\|g\|_{L^1(P)} = 1$ and $\|g\|_{L^\infty(P)} \leq \mathrm{e}^{\|h\|_{L^\infty(P)}}$. Let $f(v) = \|v-w\|_{\Sigma^{-1}}$. Then $\mathop{\mathbb{E}}_{v\sim Q}\|v-w\|_{\Sigma^{-1}} = \|f\|_{L^1(Q)} = \|fg\|_{L^1(P)}$. Finally, let χ be the indicator function for the ellipsoid $\{v : \|v-w\|_{\Sigma^{-1}} \leq R\}$, and let $\bar{\chi} = 1-\chi$. Then $\|f\chi\|_{L^\infty(P)} \leq R$ and

$$\|fg\|_{L^1(P)} = \|fg\chi\|_{L^1(P)} + \|fg\bar{\chi}\|_{L^1(P)} \tag{3.22}$$

$$\leq \|f\chi\|_{L^\infty(P)}\,\|g\|_{L^1(P)} + \|f\bar{\chi}\|_{L^1(P)}\,\|g\|_{L^\infty(P)} = R + \sqrt{\tfrac{2}{\pi}}\,\mathrm{e}^{-\frac{R^2}{2}}\mathrm{e}^{\|h\|_{L^\infty(P)}}, \tag{3.23}$$

where the inequalities follow from two applications of Hölder's inequality. Choosing $R = \sqrt{2\|h\|_{L^\infty(P)}}$ gives $\|f\|_{L^1(Q)} \leq \sqrt{2\|h\|_{L^\infty(P)}} + \sqrt{2/\pi}$. □

Proof of Corollary 3.4.7. Let $P^S = N(w(S))$ and $P^*(S) = N(w^*(S))$. By the closure of ε-differential privacy under composition, $P^*(S)$ is ε-differentially private and is absolutely continuous with respect to $N(w)$ for all w, and so satisfies the conditions of Lemma 3.4.4. In particular, with probability $1-\delta$, Eq. (3.4) holds with $\mathrm{KL}(Q||P^*(S))$ replaced by $\mathrm{KL}(Q||P^S) + Q[\ln \frac{\mathrm{d}P^S}{\mathrm{d}P^*(S)}]$.

By hypothesis, with probability at least $1-\delta-\delta'$, it also holds that $\|w(S)-w^*(S)\|_2^2 \leq C$. Then, by Lemma 3.4.5,

$$Q\left[\ln \frac{\mathrm{d}P^S}{\mathrm{d}P^*(S)}\right] \leq \frac{1}{2}\|w(S)-w^*(S)\|_2^2/\sigma_{\min} + \|w(S)-w^*(S)\|_2/\sqrt{\sigma_{\min}} \mathop{\mathbb{E}}_{v\sim Q}\|v-w(S)\|_{\Sigma^{-1}} \tag{3.24}$$

$$\leq \frac{1}{2}C/\sigma_{\min} + \sqrt{C/\sigma_{\min}} \mathop{\mathbb{E}}_{v\sim Q}\|v-w(S)\|_{\Sigma^{-1}}. \tag{3.25}$$

By Lemma 3.4.6 $\mathop{\mathbb{E}}_{v\sim Q}\|v-w(S)\|_{\Sigma^{-1}}$ is bounded for Gibbs measures based on a surrogate risk taking values in a length-Δ interval by $\sqrt{2\tau\Delta}+\sqrt{2/\pi}$. □

3.6 Empirical studies

In Sections 3.3 and 3.4 we presented data-dependent bounds and so it is necessary to study them empirically to evaluate their usefulness. The goal of this section is to make a number of arguments. First, it is an empirical question as to what value of the inverse temperature τ is sufficient to yield small empirical risk from a Gibbs posterior. Indeed, we compare to the bound of Lever, Laviolette, and Shawe-Taylor (2013), presented above as Theorem 3.1.1. As Lever, Laviolette, and Shawe-Taylor point out, this bound depends explicitly on τ, where it plays an obvious role as a measure of complexity. Second, despite how tight this bound is for small values of τ, the bound must become vacuous before the Gibbs posterior *would have* started to overfit on random labels because this bound holds for *all* data distributions. We demonstrate that this phase transition happens well before the Gibbs posterior achieves its minimum risk on true labels. Third, because our bound retains the KL term, we can potentially identify easy data. Indeed, our risk bound decreases beyond the point where the same classifier begins to overfit to random labels. Finally, our results suggest that we can use the property that SGLD converges weakly to investigate the generalization properties of Gibbs classifiers. More work is clearly needed to scale our study to full fledged deep learning benchmarks. (See Section 3.6.1 for details on how we compute the KL term and challenges there due to Gibbs posteriors Q not be exactly samplable.)

More concretely, we perform an empirical evaluation using SGLD to approximate simulating from Gibbs distributions. We are justified in taking this step due to our main result (Corollary 3.4.7), which shows that a slight weakening of the PAC-Bayes generalization bound is valid provided that SGLD converges weakly to its stationary distribution. However, because we cannot measure our convergence in practice, it is an empirical question as to whether our samples are accurate enough.

Violated bounds would be an obvious sign of trouble. We expect the bound on the classification error not to go below the true error as estimated on the heldout test set (with high probability). We perform an experiment on a MNIST (and CIFAR10, with the same conclusion so we have not included it) using true and random labels and find that no bounds are violated. The results suggest that it may be useful to empirically study bounds for Gibbs classifiers using SGLD.

Our main focus is a sythentic experiment comparing the bounds of Lever, Laviolette, and Shawe-Taylor (2013) to our new bounds based on privacy. The main finding here is that, as expected, the bounds by Lever, Laviolette, and Shawe-Taylor must explode when the Gibbs classifier begins to overfit random labels, whereas our bounds, on true labels, continue to track the training error and bound the test error.

3.6.1 Setup

Our focus is on classification by neural networks into K classes. Thus $Z = X \times [K]$.

We use neural networks that output probability vectors over these K classes. Given weights $\mathbf{w} \in \mathbb{R}^p$ and input $x \in X$, the probability vector output by the network is $p(\mathbf{w},x) \in [0,1]^K$. Networks are trained by minimizing cross entropy loss: $\ell(\mathbf{w},(x,y)) = g(p(\mathbf{w},x),y)$, where $g((p_1,\ldots,p_K),y) = -\ln p_y$. Note that cross entropy loss is merely bounded below. We report results in terms of $\{0,1\}$-valued classification error: $\ell(\mathbf{w},(x,y)) = 0$ if and only if $p(\mathbf{w},x)$ takes its maximum value only at coordinate y,

We sometimes refer to elements of $\mathbb{R}^p$ and $\mathscr{M}_1(\mathbb{R}^p)$ as classifiers and randomized classifiers, respectively, and refer to the (empirical) 0–1 risk as the (empirical) error. We train two different architectures using SGLD on MNIST and a synthetic dataset, SYNTH. The experimental setup is explained in Section 3.6.1.

One-stage training procedure We run SGLD for T training epochs with a fixed value of the parameter τ. We observe that convergence appears to occur within 10 epochs, but use a much larger number of training epochs to potentially expose nonconvergence behavior. The value of the inverse temperature τ is fixed during the whole training procedure.

Two-stage training procedure In order to evaluate our private PAC-Bayes bound (Theorem 3.3.2), we perform a two-stage training procedure:

- **Stage One.** We run SGLD for T_1 epochs with inverse temperature τ_1, minimizing the standard cross entropy objective. Let w_0 denote the neural network weights after stage one.

- **Transition.** We restart the learning rate schedule and continue SGLD for T_1 epochs with linearly annealing the temperature between τ_1 and τ_2, i.e., inverse temperature $\tau_t = ((t-T_1)\tau_2 + (2T_1 - t) * \tau_1)/T_1$, where t is the current epoch number. The objective at $\mathbf{w}$ is the cross entropy loss for $\mathbf{w}$ plus a weight-decay term $\frac{\gamma}{2}\|\mathbf{w}-\mathbf{w}_0\|_2^2$.
- **Stage Two.** We continue SGLD for $T_2 - T_1$ epochs with inverse temperature τ_2. The objective is the same as in the transition stage.

During the first stage, the k-step transitions of SGLD converge weakly towards a Gibbs distribution with a uniform base measure, producing a random vector $\mathbf{w}_0 \in \mathbb{R}^p$. The private data-dependent prior $P_{\mathbf{w}_0}$ is the Gaussian distribution centred at $\mathbf{w}_0$ with diagonal covariance $\frac{1}{\gamma} I_p$.

During the second stage, SGLD converges to the Gibbs posterior with a Gaussian base measure $P_{\mathbf{w}_0}$, i.e., $Q_{\tau_2} = P_{\exp(-\tau_2 \hat{R}_S)}$.

Bound calculation Our experiments evaluate different values of the inverse temperature τ.

We evaluate Lever bounds for the randomized classifier Q_τ obtained by the one-stage training procedure, with $T = 1000$. We do so on both the MNIST and SYNTH datasets.

We also evaluate our private PAC-Bayes bound (Theorem 3.3.2) for the randomized classifier Q_{τ_2} and the private data-dependent prior $P_{\mathbf{w}_0}$, where the privacy parameter depends on τ_1. The bound depends on the value of the $\mathrm{KL}(Q_{\tau_1}||P_{\mathbf{w}_0})$. The challenges of estimating this term are described in Section 3.6.1. We only evaluate the differentially private PAC-Bayes bounds on the small neural network and SYNTH dataset,

The parameter settings for SYNTH experiments are: $T1 = 100$, $T2 = 1000$, $\gamma = 2$; for MNIST: $T1 = 500$, $T2 = 1000$, $\gamma = 5$. When evaluating Lever bounds with a one-stage learning procedure for either datasets, $T = 1000$.

Other experimental setup details

Bounded loss While it is typical to train neural networks by minimizing cross entropy, this loss is unbounded and our theory is developed only for bounded loss. We therefore work with a bounded version of cross-entropy loss, which we obtain by preventing the network from producing extreme probabilities near zero and one. We describe our modification of the cross entropy in Section 3.6.1.

Datasets We use two datasets. The first is MNIST, which consists of handwritten digit images with labels in $\{0,...,9\}$. The dataset contains 50,000 training images and 10,000 validation images.

We also use a small synthetically generated dataset, which we refer to as SYNTH. The SYNTH dataset consists of 50 training data and 100 heldout data. Each input is a 4-dimensional vector sampled independently from a zero-mean Gaussian distribution with an identity covariance matrix. The true classifier is linear. The norm of the separating hyperplane is sampled from a standard normal.

The random label experiments are performed on a dataset where the labels are independently and uniformly generated and thus the risk is 0.5 under 0–1 loss.

Architectures We use SGLD without any standard modifications (such as momentum and batch norm) to ensure that the stationary distribution is that of SGLD. For MNIST, we use a fully connected neural network architecture. The network has 3 layers and 600 units in each hidden layer. The input is a 784 dimensional vector and the output layer has 10 units. For the SYNTH dataset, we use a fully connected neural network with 1 hidden layer consisting of 100 units. The input layer has 4 units, and the output layer is a single unit.

Learning rate At epoch t, the learning rate is $a_t = a_0 * t^{-b}$, where a_0 is the initial learning rate and b is the decay rate. We set $b = 0.5$ and use $a_0 = 10^{-5}$ for MNIST experiments and $a_0 = 10^{-3}$ for SYNTH experiments.

Minibatches An epoch refers to the full pass through the data in mini batches of size 128 for MNIST data, and 10 for SYNTH data.

Bounded cross entropy

In order to achieve differential privacy, we work with a bounded version of the cross entropy loss. The problem is associated with extreme probabilities near zero and one. Our solution is to remap the probabilities $p \mapsto \psi(p)$, where

$$\psi(p) = e^{-L_{\max}} + (1 - 2\mathrm{e}^{-L_{\max}})p \tag{3.26}$$

is an affine transformation that maps $[0,1]$ to $[e^{-L_{\max}}, 1 - e^{-L_{\max}}]$, removing extreme probability values. Cross entropy loss is then replaced by $g((p_1,\ldots,p_K),y) = -\ln \psi(p_y)$. As a result, cross entropy loss is contained in the interval $[0, L_{\max}]$. We take $L_{\max} = 4$ in our experiments.

Computing PAC-Bayes bounds for Gibbs posteriors

For a given PAC-Bayes prior P and dataset S, it is natural to ask which posterior $Q = Q(S)$ minimizes the PAC-Bayes bounds. In general, some Gibbs posterior (with respect to P) is the minimizer. We now introduce the Gibbs posterior and discuss how we can compute the term $\mathrm{KL}(Q||P)$ in the case of Gibbs posteriors.

For a σ-finite measure P over $\mathbb{R}^p$ and function $g : \mathbb{R}^p \to \mathbb{R}$, let $P[g]$ denote the expectation $\int g(h)P(\mathrm{d}h)$ and, provided $P[g] < \infty$, let P_g denote the probability measure on $\mathbb{R}^p$, absolutely continuous with respect to P, with Radon–Nikodym derivative $\frac{\mathrm{d}P_g}{\mathrm{d}P}(h) = \frac{g(h)}{P[g]}$. A distribution of the form $P_{\exp(-\tau g)}$ is generally referred to as a Gibbs distribution. A Gibbs *posterior* is a probability measure of the form $P_{\exp(-\tau \hat{R}_S)}$ for some constant $\tau > 0$.

The challenge of evaluating PAC-Bayes bounds for Gibbs posteriors is computing the KL term. We now describe a classical estimate and show that it is going to be an upper bound with high probability. Fix a prior P and $\tau \geq 0$, let $Q_\tau = P_{\exp(-\tau \hat{R}_S)}$, and let $Z_\tau = P[\exp(-\tau \hat{R}_S)]$. Then

$$\begin{aligned} \mathrm{KL}(Q_\tau||P) &= Q_\tau\Big[\ln \frac{\mathrm{d}Q_\tau}{\mathrm{d}P}\Big] && (3.27)\\ &= Q_\tau\Big[\ln \frac{\exp(-\tau \hat{R}_S)}{Z_\tau}\Big] && (3.28)\\ &= -\tau Q_\tau[\hat{R}_S] - \ln Z_\tau. && (3.29) \end{aligned}$$

Letting $W_1, \ldots, W_n \sim Q_\tau$, we have

$$Q_\tau[\hat{R}_S] = \sum_{i=1}^{n} Q_\tau[\hat{R}_S] = \mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^{n} \hat{R}_S(W_i)\Big]. \tag{3.30}$$

(The quantity within the expectation on the r.h.s. thus defines an unbiased estimator of Q_τ.) In the ideal case, the samples are independent, and then the variance decays at an n^{-1} rate. In practice, it is often difficult to even sample from Q_τ for high values of τ. Indeed, using this approach, we would generally overestimate the risk, which means that we do not obtain an upper bound on the KL term. So instead, we approximate $-\tau Q_\tau[\hat{R}_S] \approx 0$. Despite this, we obtain nonvacuous bounds.

The second term is challenging to estimate accurately, even assuming that P and Q_τ can be efficiently simulated. One tack is to consider i.i.d. samples $V_1, \ldots, V_n \sim P$, and note that

$$-\ln Z_\tau = -\ln P[\exp(-\tau \hat{R}_S)] = -\ln \mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^{n} \exp(-\tau \hat{R}_S(V_i))\Big] \tag{3.31}$$

$$\leq \mathbb{E}\Big[-\ln \frac{1}{n}\sum_{i=1}^{n} \exp(-\tau \hat{R}_S(V_i))\Big], \tag{3.32}$$

where the inequality follows from an application of Jensen's inequality. The quantity within the expectation on the r.h.s. thus forms an upper bound, and indeed, it is possible to show that it does not fall below the l.h.s. by ε with probability exponentially small in ε. Thus we have a high-probability (near) upper bound on the term in the KL. One might be inclined to compute a normalized importance sampler, but since Q cannot be effectively sampled, one does not obtain an upper bound with high probability.

The term $\ln Z_\tau$ is a generalized log marginal likelihood, which, in our experiments, we approximate by sampling from a Gaussian distribution P. Numerical integration techniques rapidly diminish in accuracy with increasing dimensionality of the parameter space.

Note, that due to the convexity of the exponential, samples $W_i \sim P$, for which $\hat{R}_S(W_i)$ is close to zero, will dominate Z_τ. Due to high dimensionality of the neural network parameter space, with high probability a random sample W_i from P will not be in the minima of the empirical loss surface and therefore $\hat{R}_S(W_i)$ will be high. As a results, in our experiments we obtain a very loose upper bound on the KL.

3.6.2 Results

Results are presented in Figs. 3.1 and 3.2. We never observe a violation of the PAC-Bayes bounds for Gibbs distributions. This suggests that our assumption that SGLD has nearly converged is accurate enough or the bounds are sufficiently loose that any effect from nonconvergence was masked.

Our MNIST experiments highlight that the Lever bounds must also upper bound the risk for every possible data distribution, including the random label distribution. In the random label experiment (right plot in Fig. 3.1), we observe that when τ gets close to the number of training samples, the generalization error starts increasing steeply. This phase transition is captured by the Lever bound. In the true label experiment (right plot), the generalization error does not rise with τ. Indeed, it continues to decrease, and so the Lever bound quickly becomes vacuous as we increase τ. The Lever bound cannot capture this behavior because it must capture also the behavior under random labels.

On the SYNTH dataset, we see the same phase transition under random labels and so Lever bounds remain vacuous after this point. In contrast, we see that our private PAC-Bayes bounds can track the error beyond the phase transition that occurs under random labels. (See Fig. 3.2.) At high values of τ, our KL upper bound becomes very loose.

Private versus Lever PAC-Bayes bound While Lever PAC-Bayes bound fails to explain generalization for high τ values, our private PAC-Bayes bound may remain nonvacuous. This is due to the fact that it retains the KL term, which is sensitive to the data distribution via Q, and thus it can be much lower than the upper bound on the KL in Lever, Laviolette, and Shawe-Taylor (2013) for datasets with small true Bayes risk. Two stage optimization, inspired by the DP PAC-Bayes bound, allows us to obtain more accurate classifiers by setting a higher inverse temperature parameter at the second stage, τ_2.

We do not plot DP PAC-Bayes bound for MNIST experiments due to the computational challenges approximating the KL term for a high-dimensional parameter space, as discussed in Section 3.6.1. We evaluate our private PAC-Bayes bound on MNIST dataset only for a combination of $\tau_1 = 10^3$ and $\tau_2 \in [3*10^3, 3*10^4, 10^5, 3*10^5]$. The values are chosen such that τ_1 gives a small penalty for using the data to learn the prior and τ_2 is chosen such that at $\tau = \tau_2$ Lever's bound returns a vacuous bound (as seen in Fig. 3.1). We use 10^5 number of samples from the DP Gaussian prior learnt in stage one to approximate $\log Z$ term that appears in the KL, as defined in Eq. (3.31).

The results are presented in the table below. While DP PAC-Bayes bound is very loose, it is still smaller than Lever's bound for high values of inverse temperature.

Note, that for smaller values τ_2, we can use Lever's upper bound on the KL term instead of performing a Monte Carlo approximation. Since τ_1 is small and adds only a small penalty ($\sim 1\%$), the DP PAC-Bayes bound is equal to Lever's bound plus a differential privacy penalty ($\sim 1\%$).

τ_2	$3*10^3$	$3*10^4$	10^5	$3*10^5$
Test	0.12	0.07	0.06	4
DP PAC-Bayes bound on test	0.21	0.35	0.65	1
Lever PAC-Bayes with $\tau = \tau_2$ on test	0.26	1	1	1

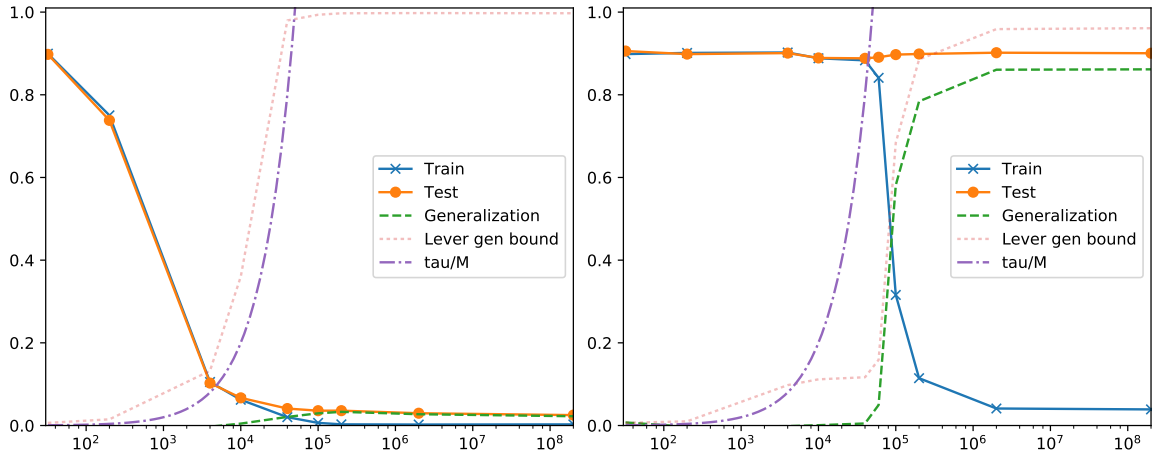


Fig. 3.1 Results for a fully connected neural network trained on MNIST dataset with SGLD and a fixed value of τ. We vary τ on the x-axis. The y-axis shows the average 0–1 loss. We plot the estimated generalization gap, which is the difference between the training and test errors. The left plot shows the results for the true label dataset. We observe, that the training error converges to zero as τ increases. Further, while the generalization error increases for intermediate values of τ (10^4 to 10^6), it starts dropping again as τ increases. We see that the Lever bound fails to capture this behaviour due to the monotonic increase with τ. The right hand side plot shows the results for a classifier trained on random labelling of MNIST images. The true error is around 0.9. For small values of τ (under 10^3) the network fails to learn and the training error stays at around 0.9. When τ exceeds the number of training points, the network starts to overfit heavily. The sharp increase of the generalization gap is predicted by the Lever bound.

3.7 Discussion

In this chapter we presented a valid PAC-Bayes bound that allows one to learn a prior from the training data. Our proposed bound works by requiring the prior to be differentially private. A prior learned with a differentially private algorithm can approximate a data-distribution-dependent prior. While the data dependency of the prior adds an additional penalty term, it may also allow us to get tighter generalization bounds by decreasing the KL term in the PAC-Bayes bound.

Achieving pure differential privacy is difficult. For example, sampling from an exponential mechanism is intractable. To achieve bounds for practical algorithms, we also derive a bound that is valid for algorithms that are weak approximations of exponential mechanism. This result is of practical importance and allows us to use of SGLD, a popular variant of SGD with stronger privacy guarantees.

In our empirical evaluation, we use SGLD to learn a PAC-Bayes prior. We empirically evaluate a two-stage learning procedure, which consists of initially running SGLD at high

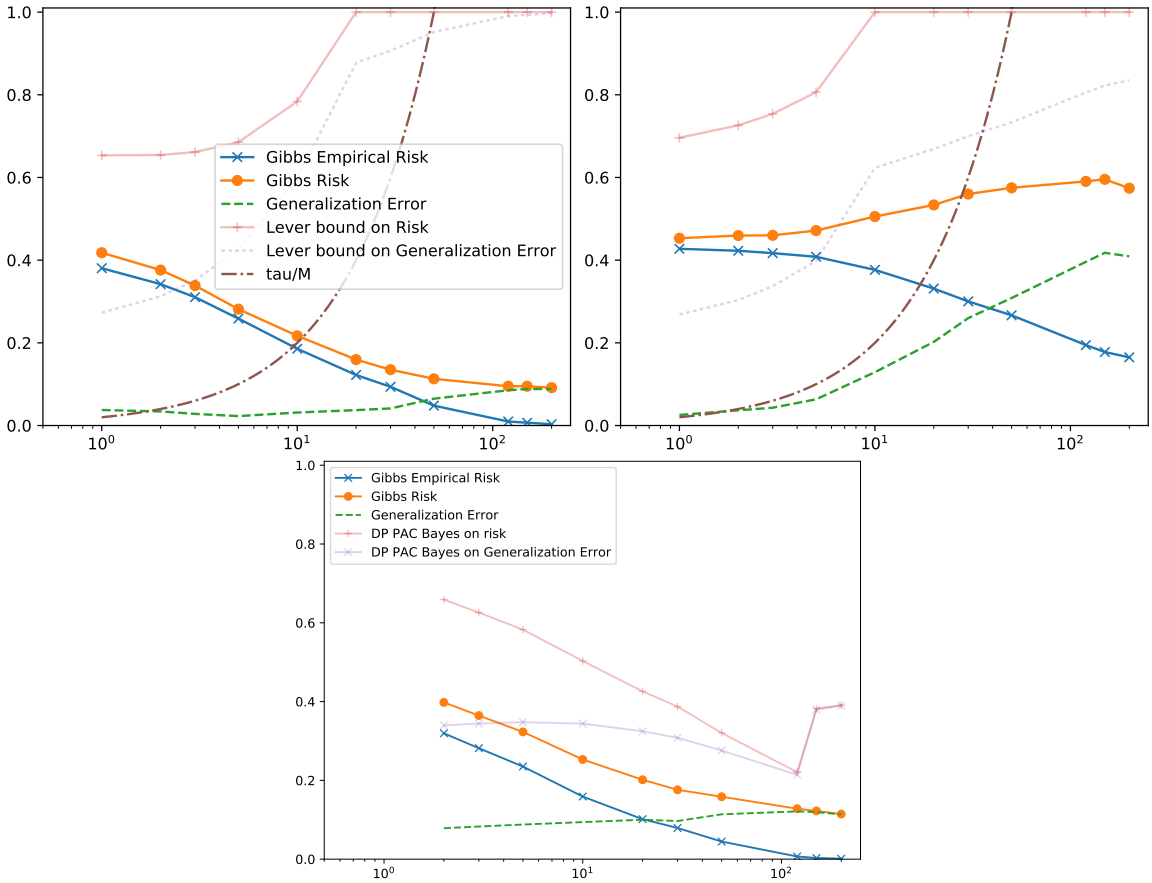


Fig. 3.2 Results for a small fully connected neural network trained on a synthetically generated dataset SYNTH, consisting of 50 training examples. The x-axis shows the τ value, and the y-axis the average 0–1 loss. To generate the top plots, we train the network with a one-stage SGLD. The top-left plot corresponds to the true label dataset, and the top-right to the random label dataset. Similarly as in MNIST experiments, we do not witness any violation of the Lever bounds. Once again, we notice that Lever bound gets very loose for larger values of τ in the true label case. The bottom plot demonstrates the results for the two-stage SGLD. In this case the x-axis plots the τ value used in the second-stage optimization. The first stage used $\tau_1 = 1$. The network is trained on true labels. We see that the differentially private PAC-Bayes bound yields a much tighter estimate of the generalization gap for larger values of τ than the Lever bound (top left). When τ becomes very large relative to the amount of training data, it becomes more difficult to sample from the Gibbs posterior. This results in a looser upper bound on the KL divergence between the prior and posterior.

temperature to learn the prior and then running SGLD at low temperature to obtain a posterior. Our results demonstrate the advantage of such two-stage training over learning at high temperature only, and provide evidence that our bounds can sometimes achieve much stronger guarantees than standard PAC-Bayes bounds, as well as those of Lever et al.

Chapter 4

Entropy-SGD optimizes the prior of a PAC-Bayes bound

4.1 Introduction

Optimization is central to much of machine learning, but generalization is the ultimate goal. Despite this, the generalization properties of many optimization-based learning algorithms are poorly understood. The standard example is stochastic gradient descent (SGD), one of the workhorses of deep learning, which has good generalization performance in many settings, even under overparametrization (Neyshabur, Tomioka, and Srebro, 2014), but rapidly overfits in others (Zhang et al., 2017). Can we develop high performance learning algorithms with provably strong generalization guarantees? Or is there a limit?

In this chapter, we study an optimization algorithm called Entropy-SGD (Chaudhari et al., 2017), which was designed to outperform SGD in terms of generalization error when optimizing an empirical risk. Entropy-SGD minimizes an objective $f : \mathbb{R}^p \to \mathbb{R}$ indirectly by performing (approximate) stochastic gradient ascent on the so-called local entropy

$$F(\mathbf{w}) := C + \log \underbrace{\mathbb{E}_{\xi\sim\mathcal{N}}[e^{-f(\mathbf{w}+\xi)}]}_{\int \exp(-f(\mathbf{w}+x))\mathcal{N}(\mathrm{d}x)}$$

where C is a constant and $\mathcal{N}$ denotes a zero-mean isotropic multivariate normal distribution on $\mathbb{R}^p$.

Our first contribution is connecting Entropy-SGD to results in statistical learning theory, showing that maximizing the local entropy corresponds to minimizing a PAC-Bayes bound (McAllester, 1999a) on the risk of the so-called Gibbs posterior. The distribution of $\mathbf{w} + \xi$ is the PAC-Bayesian "prior", and so optimizing the local entropy optimizes the bound's

prior. This connection between local entropy and PAC-Bayes follows from a result due to Catoni (2007, Lem. 1.1.3) in the case of bounded risk. (See Theorem 4.4.1.) In the special case where f is the empirical cross entropy, the local entropy is literally a Bayesian log marginal density. The connection between minimizing PAC-Bayes bounds under log loss and maximizing log marginal densities is the subject of recent work by Germain et al. (2016). Similar connections have been made by Zhang (2006a), Zhang (2006b), Grünwald (2012), and Grünwald and Mehta (2016).

Despite the connection to PAC-Bayes, as well as theoretical results by Chaudhari et al. suggesting that Entropy-SGD may be more stable than SGD, we demonstrate that Entropy-SGD (and its corresponding Gibbs posterior) can rapidly overfit, just like SGD. We identify two changes, motivated by theoretical analysis, that suffice to control generalization error, and thus prevent overfitting.

The first change relates to the stability of optimizing the prior mean. The PAC-Bayes theorem requires that the prior be independent of the data, and so by optimizing the prior mean, Entropy-SGD invalidates the bound. Indeed, the bound does not hold empirically. While a PAC-Bayes prior may not be chosen based on the data, it can depend on the data distribution. This suggests that if the prior depends only weakly on the data, it may be possible to derive a valid bound.

Indeed, in Chapter 3 we formalized this idea using differential privacy (Dwork, 2006; Dwork et al., 2015b) under the assumption of bounded risk. Using existing results connecting statistical validity and differential privacy (Dwork et al., 2015b, Thm. 11), we show that an ε-differentially private prior yields a valid, though looser, PAC-Bayes bound.

Achieving strong differential privacy can be computationally intractable. Motivated by this obstruction, in Chapter 3 we relax the privacy requirement in the case of Gaussian PAC-Bayes priors parameterized by their mean vector. We show that one need only sample a sequence that converges in distribution to the output of a differentially private mechanism. This allows one to use stochastic gradient Langevin dynamics (SGLD; Welling and Teh, 2011), which is known to converge weakly to its target distribution, under regularity conditions. We will refer to the Entropy-SGD algorithm as Entropy-SGLDwhen the SGD step on local entropy is replaced by SGLD.

The one hurdle to using data-dependent priors learned by SGLD is that we cannot easily measure how close we are to converging. Rather than abandoning this approach, we take two steps: First, we run SGLD far beyond the point where it appears to have converged visually. Second, we assume convergence, but then view/interpret the bounds as being optimistic. In effect, these two steps allow us to see the potential and limitations of using private data-dependent priors to study Entropy-SGLD.

We find that the resulting PAC-Bayes bounds are remarkably tight but still conservative in our experiments. On MNIST, when the privacy of Entropy-SGLD is tuned to contribute no more than $2\varepsilon^2 \times 100 \approx 0.2\%$ to the generalization error, the test error of the learned network is 3–8%, which is approximately 5–10 times higher than the state of the art, which for MNIST is between 0.2-1%, although the community has almost certainly overfit to MNIST.

The second change pertains to the stability of the stochastic gradient estimate made on each iteration of Entropy-SGD. This estimate is made using SGLD. (Hence Entropy-SGD is SGLD within SGD.) Chaudhari et al. make a subtle but critical modification to the noise term in the SGLD update: the noise is divided by a factor that ranges from 10^3 to 10^4. (This factor was ostensibly tuned to produce good empirical results.) Our analysis shows that, as a result of this modification, the Lipschitz constant of the objective function is approximately 10^6–10^8 times larger, and the conclusion that the Entropy-SGD objective is smoother than the original risk surface no longer stands. This change to the noise also negatively impacts the differential privacy of the prior mean. Working backwards from the desire to obtain tight generalization bounds, we are led to divide the SGLD noise by a factor of only $\sqrt[4]{m}$, where m is the number of data points. (For MNIST, $\sqrt[4]{m} \approx 16$.) The resulting bounds are nonvacuous and tighter than those we obtained in Chapter 2, although it must be emphasized that the bounds are optimistic because we assume SGLD has converged. The extent to which it has not converged would inflate the bound.

We begin by introducing sufficient background so that we can make a formal connection between local entropy and PAC-Bayes bounds. We discuss additional related work in Section 4.2. We then introduce several existing learning bounds that use differential privacy, including the PAC-Bayes bounds outlined above that use data-dependent priors. In Section 4.6, we present experiments on MNIST and CIFAR10, which provide evidence for our theoretical analysis. We close with a short discussion.

4.2 Related work

Entropy-SGD connects to early work by Hinton and Camp (1993) and Hochreiter and Schmidhuber (1997). Both sets of authors introduce regularization schemes based on minimum-description-length principles. Hinton and Camp aim to find weights with less information by finding weights whose empirical risk is insensitive to the addition of Gaussian noise; Hochreiter and Schmidhuber define an algorithm that explicitly seeks out "flat minima", i.e., a large connected region where the empirical risk surface is nearly constant. Building on work by Langford and Caruana (2002b), we revisited these ideas in Chapter 2, where a PAC-Bayes bound is minimized with respect to the posterior using nonconvex optimization.

(Achille and Soatto (2017) arrive at a similar objective from the information bottleneck perspective.) In contrast, our work shows that Entropy-SGD implicitly uses the optimal (Gibbs) posterior, and then optimizes the resulting PAC-Bayes bound with respect to the prior.

Our work also relates to renewed interest in nonvacuous generalization bounds (Langford, 2002; Langford and Caruana, 2002b), i.e., bounds on the numerical difference between the unknown classification error and the training error that are (much) tighter than the tautological upper bound of one. In Chapter 2, nonvacuous generalization bounds are demonstrated for MNIST. (Their algorithm can be viewed as variational dropout (Kingma, Salimans, and Welling, 2015) or information dropout (Achille and Soatto, 2018), with a proper data-independent prior but without local reparametrization.) Their work builds on core insights by Langford and Caruana (2002b), who computed nonvacuous bounds for neural networks five orders of magnitude smaller. Our work shows that Entropy-SGLD yields generalization bounds, though the value of these bounds depends on the degree to which SGLD is allowed to converge. Under the optimistic assumption that convergence has been achieved, we see that the resulting bounds are tighter than those computed by Chapter 2.

Our analysis of Entropy-SGLD exploits results in differential privacy (Dwork, 2008b) and its connection to generalization (Dwork et al., 2015b; Dwork et al., 2015a; Bassily et al., 2016; Oneto, Ridella, and Anguita, 2017). Entropy-SGLD is related to differentially private empirical risk minimization, which is well studied, both in the abstract (Chaudhuri, Monteleoni, and Sarwate, 2011; Kifer, Smith, and Thakurta, 2012; Bassily, Smith, and Thakurta, 2014) and in the particular setting of private training via SGD (Bassily, Smith, and Thakurta, 2014; Abadi et al., 2016). Given the connection between Gibbs posteriors and Bayesian posteriors, Entropy-SGLD also relates to the differential privacy of Bayesian and approximate sampling algorithms (Mir, 2013; Bassily, Smith, and Thakurta, 2014; Dimitrakakis et al., 2014; Wang, Fienberg, and Smola, 2015; Minami et al., 2016).

Finally, the local entropy should not be confused with the smoothed risk surface obtained by convolution with a Gaussian kernel: in that case, every point on this surface represents the risk of a randomized classifier, obtained by perturbing the parameters according to a Gaussian distribution. (This type of smoothing relates to the approach of in Chapter 2.) The local entropy also relates to a Gaussian perturbation, but the perturbation is either accepted or rejected based upon its relative performance (as measured by the exponentiated loss) compared with typical perturbations. Thus the local entropy perturbation concentrates on regions of weight space with low empirical risk, provided they have sufficient probability mass under the distribution of the random perturbation.

4.3 Preliminaries: Entropy-SGD

4.3.1 Entropy-SGD

Entropy-SGD is a gradient-based learning algorithm proposed by Chaudhari et al. (2017) as an alternative to stochastic gradient descent on the empirical risk surface $\hat{R}_S$. The authors argue that Entropy-SGD has better generalization performance. Part of that argument is a theoretical analysis of the smoothness of the local entropy surface that Entropy-SGD optimizes in place of the empirical risk surface, as well as a uniform stability argument that they admit rests on assumptions that are violated, but to a small degree empirically. As we have mentioned in the introduction, Entropy-SGD's modifications to the noise term in SGLD result in much worse smoothness. We will modify Entropy-SGD in order to stabilize its learning and control overfitting.

Entropy-SGD is stochastic gradient ascent applied to the optimization problem

$$\arg\max_{\mathbf{w}\in\mathbb{R}^p} F_{\gamma,\tau}(\mathbf{w};S), \tag{4.1}$$

where

$$F_{\gamma,\tau}(\mathbf{w};S) = \log \int_{\mathbb{R}^p} \underbrace{\exp(-\tau\hat{R}_S(\mathbf{w}') - \tau\frac{\gamma}{2}\|\mathbf{w}'-\mathbf{w}\|_2^2)}_{g_{\gamma,\tau}^{\mathbf{w},S}(\mathbf{w}')}\,\mathrm{d}\mathbf{w}'. \tag{4.2}$$

The objective $F_{\gamma,\tau}(\cdot;S)$ is known as the *local entropy*, and can be viewed as the log partition function of the unnormalized probability density function $g_{\gamma,\tau}^{\mathbf{w},S}$. (We will denote the corresponding distribution by $G_{\gamma,\tau}^{\mathbf{w},S}$.) Assuming that one can exchange differentiation and integration, it is straightforward to verify that

$$\nabla_{\mathbf{w}} F_{\gamma,\tau}(\mathbf{w};S) = \mathop{\mathbb{E}}_{\mathbf{w}'\sim G_{\gamma,\tau}^{\mathbf{w},S}}(\tau\gamma(\mathbf{w}-\mathbf{w}')), \tag{4.3}$$

and then the local entropy $F_{\gamma,\tau}(\cdot;S)$ is differentiable, even if the empirical risk $\hat{R}_S$ is not. Indeed, Chaudhari et al. show that the local entropy and its derivative are Lipschitz. Chaudhari et al. argue informally that maximizing the local entropy leads to "flat minima" in the empirical risk surface, which several authors (Hinton and Camp, 1993; Hochreiter and Schmidhuber, 1997; Baldassi et al., 2015; Baldassi et al., 2016) have argued is tied to good generalization performance (though none of these papers gives generalization bounds, vacuous or otherwise). Chaudhari et al. propose a Monte Carlo estimate of the gradient, $\nabla_{\mathbf{w}} F_{\gamma,\tau}(\mathbf{w};S) \approx \tau\gamma(\mathbf{w}-\mu_L)$, with $\mu_1 = \mathbf{w}_1$ and $\mu_{j+1} = \alpha\mathbf{w}'_j + (1-\alpha)\mu_j$, where $\mathbf{w}'_1, \mathbf{w}'_2, \ldots$ are (approximately) i.i.d. samples from $G_{\gamma,\tau}^{\mathbf{w},S}$ and $\alpha \in (0,1)$ defines a weighted average. Obtaining samples from $G_{\gamma,\tau}^{\mathbf{w},S}$ may be difficult when the dimensionality of the weight vector

Algorithm 2 One outerloop step of Entropy-SG(L)D

Input:

$\mathbf{w} \in \mathbb{R}^p$ ▷ Current weights

$S \in Z^m$ ▷ Data

$\ell : \mathbb{R}^p \times Z \to \mathbb{R}$ ▷ Loss

$\tau, \gamma, \eta, \eta', L, K, \alpha$ ▷ Parameters

Output: Weights $\mathbf{w}$ moved along stochastic gradient

1: **procedure** ENTROPY-SG(L)D-STEP
2: $\mathbf{w}', \mu \leftarrow \mathbf{w}$
3: **for** $i \in \{1, ..., L\}$ **do** ▷ Run SGLD for L iterations.
4: $\eta'_i \leftarrow \eta'/i$
5: $(z_{j_1}, \ldots, z_{j_K}) \leftarrow$ sample minibatch of size K
6: $\mathrm{d}\mathbf{w}' \leftarrow -\frac{\tau}{K} \sum_{i=1}^{K} \nabla_{\mathbf{w}'} \ell(\mathbf{w}', z_{j_i}) - \gamma\tau(\mathbf{w}' - \mathbf{w})$
7: $\xi \sim N(0, I_p)$ sample Gaussian noise
8: $\mathbf{w}' \leftarrow \mathbf{w}' + \frac{1}{2}\eta'_i \mathrm{d}\mathbf{w}' + \sqrt{\eta'_i}\xi)$
9: $\mu \leftarrow (1-\alpha)\mu + \alpha\mathbf{w}'$
▷ C.f. Eq. (4.3).
10: $\xi \sim N(0, I_p)$ sample Gaussian noise
11: $\mathbf{w} \leftarrow \mathbf{w} + \frac{1}{2}\eta\tau\gamma(\mathbf{w} - \mu) + \underbrace{\sqrt{\eta/\beta}\,\xi}_{\text{Entropy-SGLD only}}$
12: **return** $\mathbf{w}$

is large. Chaudhari et al. use Stochastic Gradient Langevin Dynamics (SGLD; Welling and Teh, 2011), which simulates a Markov chain whose long-run distribution converges to $G_{\gamma,\tau}^{\mathbf{w},S}$ and requires that the empirical risk be differentiable.[1] The final output of Entropy-SGD is the deterministic predictor corresponding to the final weights $\mathbf{w}^*$ achieved by several epochs of optimization.

Algorithm 2 gives a complete description of the stochastic gradient step performed by Entropy-SGD. If we rescale the learning rate, $\eta' \leftarrow \frac{1}{2}\eta'\tau$, lines 6 and 7 are equivalent to

$$6: \quad \mathrm{d}\mathbf{w}' \leftarrow -\tfrac{1}{K} \textstyle\sum_{i=1}^{K} \nabla_{\mathbf{w}'} \ell(\mathbf{w}', z_{j_i}) - \gamma(\mathbf{w}' - \mathbf{w})$$
$$7: \quad \mathbf{w}' \leftarrow \mathbf{w}' + \eta'_i \mathrm{d}\mathbf{w}' + \sqrt{\eta'_i}\sqrt{2/\tau}N(0, I_p)$$

Notice that the noise term is multiplied by a factor of $\sqrt{2/\tau}$. A multiplicative factor ε—called the "thermal noise", but playing exactly the same role as $\sqrt{2/\tau}$ here—appears in the original description of the Entropy-SGD algorithm given by Chaudhari et al. However,

[1] Chaudhari et al. take $L = 20$ steps of SLGD, using a constant step size $\eta'_j = 0.2$ and weighting $\alpha = 0.75$.

ε does not appear in the definition of local entropy used in their stability analysis. Our derivations highlights that scaling the noise term in SGLD update has a profound effect: the thermal noise exponentiates the density that defines the local entropy. The smoothness analysis of Entropy-SGD does not take into consideration the role of ε, which is critical because Chaudhari et al. take ε to be as small as 10^{-3} and 10^{-4}. Indeed, the conclusion that the local entropy surface is smoother no longer holds. We will see that τ controls the differential privacy and thus the generalization error of Entropy-SGD.

4.4 Maximizing local entropy minimizes a PAC-Bayes bound

We now present our first contribution, a connection between the local entropy and PAC-Bayes bounds. We begin with some notation for Gibbs distributions. For a measure P on $\mathbb{R}^p$ and function $g : \mathbb{R}^p \to \mathbb{R}$, let $P[g]$ denote the expectation $\int g(h)P(\mathrm{d}h)$ and, provided $P[g] < \infty$, let P_g denote the probability measure on $\mathbb{R}^p$, absolutely continuous with respect to P, with Radon–Nikodym derivative $\frac{\mathrm{d}P_g}{\mathrm{d}P}(h) = \frac{g(h)}{P[g]}$. A distribution of the form $P_{\exp(-\tau g)}$ is generally referred to as a Gibbs distribution. In the special case where P is a probability measure, we call $P_{\exp(-\tau \hat{R}_S)}$ a "Gibbs posterior".

Theorem 4.4.1 (Maximizing local entropy optimizes a PAC-Bayes bound's prior)**.** *Assume the loss takes values in an interval of length L_{max}, let $\tau = \frac{m}{\lambda L_{max}}$ for some $\lambda > 1/2$, Then the set of weight* $\mathbf{w}$ *maximizing the local entropy $F_{\gamma,\tau}(\mathbf{w}; S)$ equals the set of weights* $\mathbf{w}$ *minimizing the right hand side of Theorem 1.2.2 for $Q = G^{\mathbf{w},S}_{\gamma,\tau} = P_{\exp(-\tau \hat{R}_S)}$ and P a multivariate normal distribution with mean* $\mathbf{w}$ *and covariance matrix $(\tau\gamma)^{-1} I_p$.*

Proof of Theorem 4.4.1. Let m, δ, $\mathscr{D}$, and P be as in Theorem 1.2.2 and let $S \sim \mathscr{D}^m$. The linear PAC-Bayes bound (Theorem 1.2.2) ensures that for any fixed $\lambda > 1/2$ and bounded loss function, with probability at least $1 - \delta$ over the choice of S, the bound

$$\left(1 - \frac{1}{2\lambda}\right) \frac{m R_{\mathscr{D}}(Q)}{\lambda L_{\max}} \leq \frac{m \hat{R}_S(Q)}{\lambda L_{\max}} + \mathrm{KL}(Q||P) + g(\delta).$$

holds for all $Q \in \mathscr{M}_1(\mathbb{R}^p)$. Minimizing the upper bound on $R_{\mathscr{D}}(Q)$ is equivalent to the problem

$$\inf_Q \; Q[r] + \mathrm{KL}(Q||P) \tag{4.4}$$

with $r(h) = \frac{m}{\lambda L_{\max}} \hat{R}_S(h)$. By (Catoni, 2007, Lem. 1.1.3), for all $Q \in \mathscr{M}_1(\mathbb{R}^p)$ with $\mathrm{KL}(Q||P) < \infty$,

$$-\log P[\exp(-r)] = Q[r] + \mathrm{KL}(Q||P) - \mathrm{KL}(Q||P_{\exp(-r)}). \tag{4.5}$$

Using Eq. (4.5), we may reexpress Eq. (4.4) as

$$\inf_Q \ \mathrm{KL}(Q||P_{\exp(-r)}) - \log P[\exp(-r)].$$

By the nonnegativity of the Kullback–Liebler divergence, the infimum is achieved when the KL term is zero, i.e., when $Q = P_{\exp(-r)}$. Then

$$\left(1 - \frac{1}{2\lambda}\right) \frac{m}{\lambda L_{\max}} R_{\mathscr{D}}(P_{\exp(-r)}) \leq -\log P[\exp(-r)] + g(\delta).$$

Finally, it is plain to see that $F_{\gamma,\tau}(\mathbf{w};S) = C + \log P[\exp(-r)]$ when $C = \frac{1}{2} p \log(2\pi(\tau\gamma)^{-1})$ is a constant, $\tau = \frac{m}{\lambda L_{\max}}$, and $P = \mathscr{N}(\mathbf{w}, (\tau\gamma)^{-1} I_p)$ is a multivariate normal with mean $\mathbf{w}$ and covariance matrix $(\tau\gamma)^{-1} I$. □

The theorem requires the loss function to be bounded, because the PAC-Bayes bound we have used applies only to bounded loss functions. Germain et al. (2016) described PAC-Bayes generalization bounds for unbounded loss functions, though it requires that one make additional assumptions about the distribution of the empirical risk, which we would prefer not to make. (See Grünwald and Mehta (2016) for related work on excess risk bounds and further references).

4.5 Data-dependent PAC-Bayes priors

Theorem 4.4.1 reveals that Entropy-SGD is optimizing a PAC-Bayes bound with respect to the prior. As a result, the prior P depends on the sample S, and the hypotheses of the PAC-Bayes theorem (Theorem 1.2.2) are not met. Naively, it would seem that this interpretation of Entropy-SGD cannot explain its ability to generalize. Using tools from differential privacy, in Chapter 3 we show that if the prior term is optimized in a differentially private way, then a PAC-Bayes theorem still holds, at the cost of a slightly looser bound. We will assume basic familiarity with differential privacy. (See Section 1.4 for a basic summary.)

Using Theorem 3.2.2, one can compute tail bounds on the generalization error of fixed classifiers, and then, provided that a classifier is learned from data in a differentially private way, the tail bound holds on the classifier, with less confidence. The following two tail bounds are examples of this idea due to Oneto, Ridella, and Anguita (2017, Lem. 2 and Lem. 3).

Theorem 4.5.1. *Let $m \in \mathbb{N}$, let $\mathscr{A} : Z^m \rightsquigarrow \mathbb{R}^p$ be ε-differentially private, and let $\delta > 0$. Then $|R_{\mathscr{D}}(\mathscr{A}(S)) - \hat{R}_S(\mathscr{A}(S))| < \bar{\varepsilon} + m^{-\frac{1}{2}}$ with probability at least $1 - \delta$ over $S \sim \mathscr{D}^m$, where*

$\bar{\varepsilon} = \max\{\varepsilon, \sqrt{\frac{1}{m}\log\frac{3}{\delta}}\}$. *The same holds for the upper bound* $\sqrt{(6\hat{R}_S(\mathscr{A}(S)))}(\bar{\varepsilon} + m^{-\frac{1}{2}}) + 6(\bar{\varepsilon}^2 + m^{-1})$.

4.5.1 An ε-differentially private PAC-Bayes bound

The PAC-Bayes theorem allows one to choose the prior based on the data-generating distribution $\mathscr{D}$, but not on the data $S \sim \mathscr{D}^m$. In Theorem 3.3.2, we show that one can consider a data-dependent prior $\mathscr{P}(S)$ by using differential privacy.

Note that the bound stated in Theorem 3.3.2 holds for any posterior Q, including one obtained by optimizing a *different* PAC-Bayes bound. Note that, in realistic scenarios, δ is large enough relative to ε that an ε-differentially private prior $\mathscr{P}(S)$ contributes $2\varepsilon^2$ to the generalization error. Therefore, ε must be much less than one to not contribute a nontrivial amount to the generalization error. As discussed in Chapter 3, one can match the m^{-1} rate by which the KL term decays choosing $\varepsilon \in O(m^{-1/2})$. Our empirical studies use this rate.

4.5.2 Differentially private data-dependent priors

We have already explained that the weights learned by Entropy-SGD can be viewed as the mean of a data-dependent prior $\mathscr{P}(S)$. By Theorem 3.3.2 and the fact that post-processing does not decrease privacy, it would suffice to establish that the mean is ε-differentially private in order to obtain a risk bound on the corresponding Gibbs posterior classifier.

The standard (if idealized) approach for optimizing a data-dependent objective in a private way is to use the exponential mechanism (McSherry and Talwar, 2007). In the context of maximizing the local entropy, the exponential mechanism corresponds to sampling exactly from the "local entropy (Gibbs) distribution" $P_{\exp(\beta F_{\gamma,\tau}(\cdot;S))}$, where $\beta > 0$ and P is some measure on $\mathbb{R}^p$. (It is natural to take P to be Lebesgue measure, or a multivariate normal distribution, which would correspond to L2 regularization of the local entropy.) The following result establishes the privacy of a sample from the local entropy distribution:

Theorem 4.5.2. *Let $\gamma, \tau > 0$, and assume the range of the loss is contained in an interval of length L_{max}. One sample from the local entropy distribution $P_{\exp(\beta F_{\gamma,\tau}(\cdot;S))}$ is $\frac{2\beta L_{max}\tau}{m}$-differentially private.*

Proof of Theorem 4.5.2. The result follows immediately from Lemma 3.4.1 and the following lemma. □

Lemma 4.5.3. *Let $F_{\gamma,\tau}(\mathbf{w};S)$ be defined as Eq.* (4.2)*, assume the range of the loss is contained in an interval of length L_{max}, and define $q(S,\mathbf{w}) = -F_{\gamma,\tau}(\mathbf{w};S)$. Then $\Delta q := \sup_{S,S'}\sup_{\mathbf{w}\in\mathbb{R}^p} |q(S,h) - q(S',h)| \leq \frac{L_{max}\tau}{m}$.*

Proof. The proof essentially mirrors that of (McSherry and Talwar, 2007, Thm. 6). □

Sampling from exponential mechanisms exactly is generally intractable. We therefore rely on the result proved in Corollary 3.4.8, which allows us to use SGLD to produce an approximate sample and obtain the same bound up to a term that depends on the degree of convergence.

Remark 4.5.4. One must verify that SGLD produces a sequence that converges in distribution. Teh, Thiery, and Vollmer (2016) give sufficient conditions that suffice: they ensure that the gradients do not explode and cause the diffusion to fail to converge. There are simple examples where this behaviour can arise, but this problem does not appear to plague the application of SGLD to neural networks.

In summary, to obtain a data-dependent prior which yields a valid PAC-Bayes bound, we optimize the local entropy $F_{\gamma,\tau}(\cdot;S)$ using SGLD, repeatedly performing the update

$$\mathbf{w} \leftarrow \mathbf{w} + \tfrac{1}{2}\eta\hat{g}(\mathbf{w}) + \sqrt{\eta/\beta}\,N(0,I_p),$$

where at each round $\hat{g}(\mathbf{w})$ is an estimate of the gradient $\nabla_{\mathbf{w}} F_{\gamma,\tau}(\mathbf{w};S)$. (Recall the identity Eq. (4.3).) As in Entropy-SGD, we construct biased gradient estimates via an inner loop of SGLD. (We ignore this source of error.) In summary, the only change to Entropy-SGD is the addition of noise in the outer loop. We call the resulting algorithm Entropy-SGLD. (See Algorithm 2.)

As we run SGLD longer, we obtain a tighter bound that holds with at least probability approaching $1-\delta$. However, in practice we may not know the rate at which this convergence occurs. In our experiments, we use very long runs to approximate near-convergence and then only interpret the bounds as being optimistic. We return to these issues in Sections 4.6 and 4.10.

4.6 Numerical evaluations on MNIST

PAC-Bayes bounds for Entropy-SGLD are data-dependent and so the question of their utility is an empirical one that requires data. In this section, we perform an empirical study of SGD, SGLD, Entropy-SGD, and Entropy-SGLD on the MNIST and CIFAR10 data sets, using both convolutional and fully connected architectures, and comparing several numerical generalization bounds to test errors estimated based on held-out data.

The PAC-Bayes bounds we use depends on the privacy of a sample from the local entropy distribution. (Bounds for SGLD depend on the privacy of a sample from the Gibbs posterior.)

For the local entropy distribution, the degree ε of privacy is determined by the product of the τ and β parameters of the local entropy distribution. (Thermal noise is $\sqrt{2/\tau}$.) In turn, ε increases the generalization bound. For a fixed β, theory predicts that τ affects the degree of overfitting. We see this empirically. No bound we compute is violated more frequently then it is expected to be. The PAC-Bayes bound for SGLD is expanded by an amount ε' that goes to zero as SGLD converges. We assume SGLD has converged and so the bounds we plot are optimistic. We discuss this point below in light of our empirical results, and then return to this point in the discussion.

The weights learned by SGD, SGLD, and Entropy-SGD are treated differently from those learned by Entropy-SGLD. In the former case, the weights parametrize a neural network as usual, and the training and test error are computed using these weights. In the latter case, the weights are taken to be the mean of a multivariate normal prior, and we evaluate the training and test error of the associated Gibbs posterior (i.e., a randomized classifier). We also report the performance of the (deterministic) network parametrized by these weights (the "mean" classifier) in order to give a coarse statistic summarizing the local empirical risk surface.

Following Zhang et al. (2017), we study these algorithms on MNIST with the original ("true") labels, as well as on random labels. Parameter τ that performs very well in one setting often does not perform well in the other. Random labels mimic data where the Bayes error rate is high, and where overfitting can have severe consequences.

4.6.1 Details

We use a two-class variant of MNIST (LeCun, Cortes, and Burges, 2010).[2] Some experiments involve random labels, i.e., labels drawn independently and uniformly at random at the start of training. We study three network architectures, abbreviated FC600, FC1200, and CONV. Both FC600 and FC1200 are 3-layer fully connected networks, with 600 and 1200 units per hidden layer, respectively. CONV is a convolutional architecture. All three network architectures are taken from the MNIST experiments by Chaudhari et al. (2017), but adapted to our two-class version of MNIST.[3] Let S and S_{tst} denote the training and test sets, respectively. For all learning algorithms we track

(i) $R_S^{0\text{–}1}(\mathbf{w})$ and $R_{S_{\text{tst}}}^{0\text{–}1}(\mathbf{w})$, i.e., the training/test error for $\mathbf{w}$.

We also track

[2] The MNIST handwritten digits dataset (LeCun, Cortes, and Burges, 2010) consists of 60000 training set images and 10000 test set images, labeled 0–9. We transformed MNIST to a two-class (i.e., binary) classification task by mapping digits 0–4 to label 1 and 5–9 to label -1.

[3] We adapt the code provided by Chaudhari et al., with some modifications to the training procedure and straightforward changes necessary for our binary classification task.

(ii) estimates of $R_S^{0\text{–}1}(G_{\gamma,\tau}^{\mathbf{w},S})$ and $R_{S_{\text{tst}}}^{0\text{–}1}(G_{\gamma,\tau}^{\mathbf{w},S})$, i.e., the mean training and test error of the local Gibbs distribution, viewed as a randomized classifier ("Gibbs")

and, using the bound stated in Theorem 4.5.2, we compute

(iii) a PAC-Bayes bound on $R_{\mathscr{D}}^{0\text{–}1}(G_{\gamma,\tau}^{\mathbf{w},S})$ using Theorem 3.3.2 ("PAC-bound");

(iv) the mean of a Hoeffding-style bound on $R_{\mathscr{D}}^{r}(\mathbf{w}')$, where the underlying loss is the ramp loss with slope 10^6 and $\mathbf{w}' \sim P_{\exp(F_{\gamma,\tau}(\cdot;S))}$, using the first bound of Theorem 4.5.1 ("H-bound");

(v) an upper bound on the mean of a Chernoff-style bound on $R_{\mathscr{D}}^{r}(\mathbf{w}')$, where $\mathbf{w}' \sim P_{\exp(F_{\gamma,\tau}(\cdot;S))}$, using the second bound of Theorem 4.5.1 ("C-bound").

We also compute H- and C- bounds for SGLD, viewed as a sampler for $\mathbf{w}' \sim P_{\exp(-\tau\hat{R}_S)}$, where P is Lebesgue measure.

In order to get privacy guarantees for SGLD and Entropy-SGLD, we modify the cross entropy loss function to be bounded. (See Section 3.6.1). With the choice of $\beta = 1$ and $\tau = \sqrt{m}$, and the loss function taking values in an interval of length $L_{\max} = 4$, the local entropy distribution is an ε-differentially private mechanism with $\varepsilon \approx 0.0327$. See Section 4.7.2 for additional details. Note that, in the calculation of (iii), we do not account for Monte Carlo error in our estimate of $R_S^{0\text{–}1}(\mathbf{w})$. The effect is small, given the large number of iterations of SGLD performed for each point in the plot. Recall that

$$R_{\mathscr{D}}^{0\text{–}1}(G_{\gamma,\tau}^{\mathbf{w},S}) = \mathop{\mathbb{E}}_{\mathbf{w}' \sim G_{\gamma,\tau}^{\mathbf{w},S}}(R_{\mathscr{D}}^{0\text{–}1}(\mathbf{w}')),$$

and so we may interpret the bounds in terms of the performance of a randomized classifier or the mean performance of a randomly chosen classifier.

4.6.2 Results

Key results for the convolutional architecture (CONV) appear in Fig. 4.1. Results for FC600 and FC1200 appear in Fig. 4.2 of Section 4.7. (Training the CONV network produces the lowest training/test errors and tightest generalization bounds. Results and bounds for FC600 are nearly identical to those for FC1200, despite FC1200 having three times as many parameters.)

The top row of Fig. 4.1 presents the performance of SGLD for various levels of thermal noise $\sqrt{2/\tau}$ under both true and random labels. (Assuming SGLD is close to weak convergence, we may also use SGLD to directly perform a private optimization of the empirical risk surface. The level of thermal noise determines the differential privacy of SGLD's stationary distribution and so we expect to see a tradeoff between empirical risk and generalization error. Note that, algorithmically, SGD is SGLD with zero thermal noise.) SGD achieves

the smallest training and test error on true labels, but overfits the worst on random labels. In comparison, SGLD's generalization performance improves with higher thermal noise, while its risk performance worsens. At 0.05 thermal noise, SGLD achieves reasonable but relatively large risk but almost zero generalization error on both true and random labels. Other thermal noise settings have either much worse risk or generalization performance.

The middle row of Fig. 4.1 presents the performance of Entropy-SGD for various levels of thermal noise $\sqrt{2/\tau}$ under both true and random labels. As with SGD, Entropy-SGD's generalization performance improves with higher thermal noise, while its risk performance worsens. At the same levels of thermal noise, Entropy-SGD outperforms the risk and generalization error of SGD. At 0.01 thermal noise, Entropy-SGD achieves good risk and low generalization error on both true and random labels. However, the test-set performance of Entropy-SGD at 0.01 thermal noise is still worse than that of SGD. Whether this difference is due to SGD overfitting to the MNIST test set is unclear and deserves further study.

The bottom row of Fig. 4.1 presents the performance of Entropy-SGLD with $\tau = \sqrt{m}$ on true and random labels. (This corresponds to approximately 0.09 thermal noise.) On true labels, both the mean and Gibbs classifier learned by Entropy-SGLD have approximately 2% test error and essentially zero generalization error, which is less than predicted by the bounds evaluated. The differentially private PAC-Bayes risk bounds are roughly 3%. As expected by the theory, Entropy-SGLD, properly tuned, does not overfit on random labels, even after thousands of epochs.

We find that the PAC-Bayes bounds are generally tighter than the H- and C-bounds. All bounds are nonvacuous, though still loose. The error bounds reported here are tighter than those reported in Chapter 2. However, *the bounds are optimistic because they do not include the additional term which measure how far SGLD is from its weak limit.* Despite the bounds being optimistically tight, we see almost no violations in the data. (Many violations would undermine our assumption.) While we observe tighter generalization bounds than previously reported, and better test error, we are still far from the performance of SGD. The optimistic picture we get from the bounds suggests we need to develop new approaches. Weaker notions of stability/privacy may be necessary to achieve further improvement in generalization error and test error.

4.7 Two-class MNIST experiments

4.7.1 Architecture

We study three architectures: CONV, FC600, and FC1200.

CONV is a convolutional neural network, whose architecture is the same as that used by Chaudhari et al. (2017) for multiclass MNIST classification, except modified to produce a single probability output for our two-class variant of MNIST. In particular, CONV has two convolutional layers, a fully connected ReLU layer, and a sigmoidal output layer, yielding $126,711$ parameters in total.

FC600 and FC1200 are fully connected 3-layer neural networks, with 600 and 1200 hidden units, respectively, yielding $834,601$ and $2,385,185$ parameters in total, respectively. We use ReLU activations for all but the last layer, which was sigmoidal to produce an output in $[0,1]$.

In their MNIST experiments, Chaudhari et al. (2017) use dropout and batch normalization. We do not use dropout. The bounds we achieve with and without batch norm are very similar. Without batch norm, however, it is necessary to tune the learning rates. Understanding the combination of SGLD and batch norm and the limiting invariant distribution, if any, is an important open problem.

4.7.2 Training objective and hyperparameters for Entropy-SGLD

Epochs

Ordinarily, an epoch implies one pass through the entire data set. For SGD, each stochastic gradient step processes a minibatch of size $K = 128$. Therefore, an epoch is $m/K = 468$ steps of SGD. An epoch for Entropy-SGD and Entropy-SGLD is defined as follows: each iteration of the inner SGLD loop processes a minibatch of size $K = 128$, and the inner loop runs for $L = 20$ steps. Therefore, an epoch is $m/(LK)$ steps of the outer loop. In concrete terms, there are 20 steps of SGD per every one step of Entropy-SG(L)D. Concretely, the x-axis of our plots measure epochs divided by L. This choice, used also by Chaudhari et al. (2017), ensures that the wall-clock time of Entropy-SG(L)D and SGD align.

SGLD parameters: step sizes and weighted averages

The step sizes for SGLD must be square summable but not summable. The step sizes for the outer SGLD loop are of the form $\eta_t = \eta t^{-0.6}$, with $\eta = \frac{\eta'}{\gamma\tau}$, where $\eta' = 0.006$ and is called the base learning rate. The step sizes for the inner SGLD loop are of the form $\eta_t = \eta t^{-1}$, with $\eta = \frac{2}{\tau}$.

The estimate produced by the inner SGLD loop is computed using a weighted average (line 8) with $\alpha = 0.75$. We use SGLD again when computing the PAC-Bayes generalization bound (Section 4.7.3). In this case, SGLD is used to sample from the local Gibbs distribution when estimating the Gibbs risk and the KL term. We run SGLD for 1000 epochs to obtain

our estimate. Again, we use weighted averages, but with $\alpha = 0.005$, in order to average over a larger number of samples and better control the variance.

Gibbs classifier parameters

We set $\gamma = 1$ and $\tau = \sqrt{m}$ and keep the values fixed during optimization. By Theorem 4.5.2, the value of τ, $L_{\max}$, and β determine the differential privacy of sampling once from the local entropy distribution, which in turn affects the PAC-Bayes bounds for Entropy-SGLD. The differential privacy parameter ε and confidence parameter δ contribute

$$2\frac{\max\{\ln\frac{3}{\delta},\, m\varepsilon^2\}}{m} \tag{4.6}$$

to the bound on the KL-generalization error $\mathrm{kl}(R_S^{0\text{-}1}(Q)||R_{\mathscr{D}}^{0\text{-}1}(Q))$ in the differentially private PAC-Bayes bound (Theorem 3.3.2). Choosing $\tau = \sqrt{m}$, implies that the contribution coming from differential privacy decays at a rate of $1/m$. Numerically, given $L_{\max} = 4$ and $\beta = 1$, this contribution is 0.002.

4.7.3 Evaluating the PAC-Bayes bound

Inverting $\mathrm{kl}(q||p)$

In order to bound the risk using the differentially private PAC-Bayes bound, we must compute the largest value p such that $\mathrm{kl}(q||p) \leq c$. There does not appear to be a simple formula for this value. In practice, however, the value can be efficiently numerically approximated using, e.g., Newton's method. See Section 2.4.

Estimating the KL divergence

Let $\ell(\mathbf{w}) = \tau \breve{R}_S(\mathbf{w})$. By (Catoni, 2007, Lem. 1.1.3),

$$\mathrm{KL}(P_{\exp(-\ell)}||P) = \mathop{\mathbb{E}}_{\mathbf{w}\sim P_{\exp(-\ell)}}[-\ell(\mathbf{w})] - \log P[\exp(-\ell)]. \tag{4.7}$$

In Chapter 3 we make use of this to propose the two following Monte Carlo estimates:

$$\mathop{\mathbb{E}}_{\mathbf{w}\sim P_{\exp(-\ell)}}[-\ell(\mathbf{w})] \approx -\frac{1}{k'}\sum_{i=1}^{k'}\ell(\mathbf{w}') \tag{4.8}$$

where $\mathbf{w}'_1, \ldots, \mathbf{w}'_{k'}$ are taken from a Markov chain targeting $P_{\exp(-\ell)}$, such as SGLD run for $k' \gg 1$ steps (which is how we computed our bounds), and

$$\log P[\exp(-\ell)] = \log \int \exp\{-\ell(\mathbf{w})\} P(\mathrm{d}\mathbf{w}) \tag{4.9}$$

$$\gtrapprox \log \frac{1}{k} \sum_{i=1}^{k} \exp\{-\ell(\mathbf{w}_i)\}. \tag{4.10}$$

where $h_1, \ldots, h_k$ are i.i.d. P (which is a multivariate Gaussian in this case). In the latter case, due to the concavity of log, the estimate is a lower bound with high probability, yielding a high probability upper bound on the KL term.

4.8 Multiclass MNIST experiments

We evaluate the same generalization bounds on the standard MNIST classification task as in the MNIST binary labelling case. The results are presented in Fig. 4.4.

All the details of the network architectures and parameters are as stated in Section 4.7.2, with two exception: following Chaudhari et al. (2017), we use a fully connected network with 1024 hidden units per layer, denoted FC1024.

4.8.1 Objective

The neural network produces a probability vector $(p_1, \ldots, p_K)$ via a soft-max operation. Ordinarily, we then apply the cross entropy loss $-\log p_y$. When training privately, we use a bounded variant of the cross entropy loss, $-\log \psi(p_y)$, where ψ is defined as in Eq. (3.26).

4.9 CIFAR10 experiments

We train a convolutional neural network on CIFAR10 data (Krizhevsky, 2009) with true and random labels. The architecture of the network is identical to the one used in Chaudhari et al. (2017), but the training is performed with no dropout or weight decay.

Most of the experimental details are the same as in MNIST experiments described in Section 4.8. In particular, the training objective and the learning rate schedule is identical, with the initial outer loop base learning rate and decay both set to 0.1. The results reported in Figs. 4.5 and 4.6 are recorded after training for far more steps than necessary, in order to allow SGLD to get closer to its target distribution. In more detail, the results are obtained after 100 calls of the outerloop step (i.e., 2000 epochs), while we observe that the training error

converges very quickly, in most cases within the first 5 calls of the outerloop Entropy-SGLD step. In order to estimate the Gibbs randomized classifier error and KL divergence and evaluate the PAC-Bayes bound, we run SGLD for an extra 50 epochs at the end of training.

4.9.1 Privacy parameter experiments

We start by fixing $\beta = 1$ and experimenting with different values of the parameter τ. Recall that the product $\tau\beta$ determines the privacy level. The value $\tau = 10^8$ was used by Chaudhari et al. (2017) in their CIFAR10 experiments.

The top row of Fig. 4.5 presents the results for $\gamma = 0.03$, which is the same value used by Chaudhari et al. (2017). The random labels plot highlights a phase transition for τ values in the range 10^4 to 10^5. We observe that very little overfitting occurs for smaller values of τ on both true and random labels. When τ exceeds 10^5, the size of generalization gap appears to be data distribution dependent. For random label dataset with large true Bayes error, the classifier can achieve almost zero classification error, resulting in maximal generalization error. However, in the case of true label data, we see that the generalization error does not exceed 0.2.

Note, that for high values of τ, the differentially private PAC-Bayes bound is completely dominated by the differential privacy penalty. Effectively, the bound becomes data independent and thus cannot capture this difference in generalization error for true and random labels.

The bottom row contains results for $\gamma = 3$, which corresponds to shrinking the variance of the Gaussian prior and thus decreasing smoothing of the empirical error surface. One can recognize the same patterns as in the top row, but now the phase transition happens substantially earlier. Due to this shift, the DP-PAC-Bayes bound approaches the C-gen bound on the generalization error.

4.9.2 Prior variance experiments

The differential privacy of sampling from the local entropy distribution does not directly depend on γ. However, the optimization problem is clearly affected by the value of γ, and so the performance achievable within a gives privacy budget is affected by γ. We fixed the privacy level by taking $\tau\beta = 2000$, and experimented with different γ values. The results are presented in Fig. 4.6.

The left plot shows the results for $\beta = 1$, which is the same value used in Fig. 4.5 experiments and all MNIST experiments. For a large range of γ values (around 0.03 to 10), we achieve similar DP-PAC-Bayes bound on the generalization error. However, $\tau \in [1,3]$

yields the smallest bound on the risk and also the best performing classifiers, as judged by the risk evaluated on the test error. A value of $\tau < 0.01$ corresponds to large prior variance and excessive smoothing, which results in Entropy-SGLD finding a poor classifier.

On the right hand side plot, we reduce β to 0.004 to be able to increase τ and maintain the same level of privacy. This corresponds to higher SGLD noise on the outerloop step, and smaller noise on the inner SGLD step. Remember, that the prior variance is $(\gamma\tau)^{-1}$. Since τ is now a lot higher, $\gamma < 0.001$ results in less smoothing than in the $\beta = 1$ case and we see that Entropy-SGLD is now able to find a relatively good classifier while preserving the same level of privacy. In addition, we see further improvement in the bound on the risk and the test error for a larger range of γ values ($\gamma \in [0.0003, 20]$).

4.10 Discussion

Our work reveals that Entropy-SGD can be understood as optimizing a PAC-Bayes generalization bound in terms of the bound's prior. Because the prior must be independent of the data, the bound is invalid, and, indeed, we observe overfitting in our experiments with Entropy-SGD when the thermal noise $\sqrt{2/\tau}$ is set to 0.0001 as suggested by Chaudhari et al. for MNIST.

PAC-Bayes priors can, however, depend on the data distribution. This flexibility seems wasted, since the data sample is typically viewed as one's only view onto the data distribution. However, using results combining differential privacy and PAC-Bayes bounds, we arrive at an algorithm, Entropy-SGLD, that minimizes its own PAC-Bayes bound (though for a surrogate risk). Entropy-SGLD performs an approximately private computation on the data, extracting information about the underlying distribution, without undermining the statistical validity of its PAC-Bayes bound. The cost of using the data is a looser bound, but the gains in choosing a better prior make up for the loss. (The gains come from the KL term being much smaller on the account of the prior being better matched to the data-dependent posterior.)

Our bounds based on Theorem 3.3.2 are optimistic because we do not include the ε' term, assuming that SGLD has essentially converged. We do not find overt evidence that our approximation is grossly violated, which would be the case if we saw the test error repeatedly falling outside our confidence intervals. We believe that it is useful to view the bounds we obtain for Entropy-SGLD as being optimistic and representing the bounds we might be able to achieve rigorously should there be a major advance in private optimization. (No analysis of the privacy of SGLD takes advantage of the fact that it mixes weakly, in part because it's difficult to characterize how much it has converged in any real-world setting after a finite number of steps.) On the account of using private data-dependent priors (and

making optimistic assumptions), the bounds we observe for Entropy-SGLD are significantly tighter than those reported by Chapter 2. However, despite our bounds potentially being optimistic, the test set error we are able to achieve is still 5–10 times worse than that of SGD. Differential privacy may be too conservative for our purposes, leading us to underfit. We are able to achieve good generalization on both true and random labels under 0.01 thermal noise, despite this value of noise being too large for tight bounds. Identifying the appropriate notion of privacy/stability to combine with PAC-Bayes bounds is an important problem.

Despite Entropy-SGLD having much stronger generalization guarantees, Entropy-SGLD learns much more slowly than Entropy-SGD, the test error of Entropy-SGLD is far from state of the art, and the PAC-Bayes bounds, while much tighter than existing bounds, are still quite loose. It seems possible that we may be facing a fundamental tradeoff between the speed of learning, the excess risk, and the ability to produce a certificate of one's generalization error via a rigorous bound. Characterizing the relationship between these quantities is an important open problem.

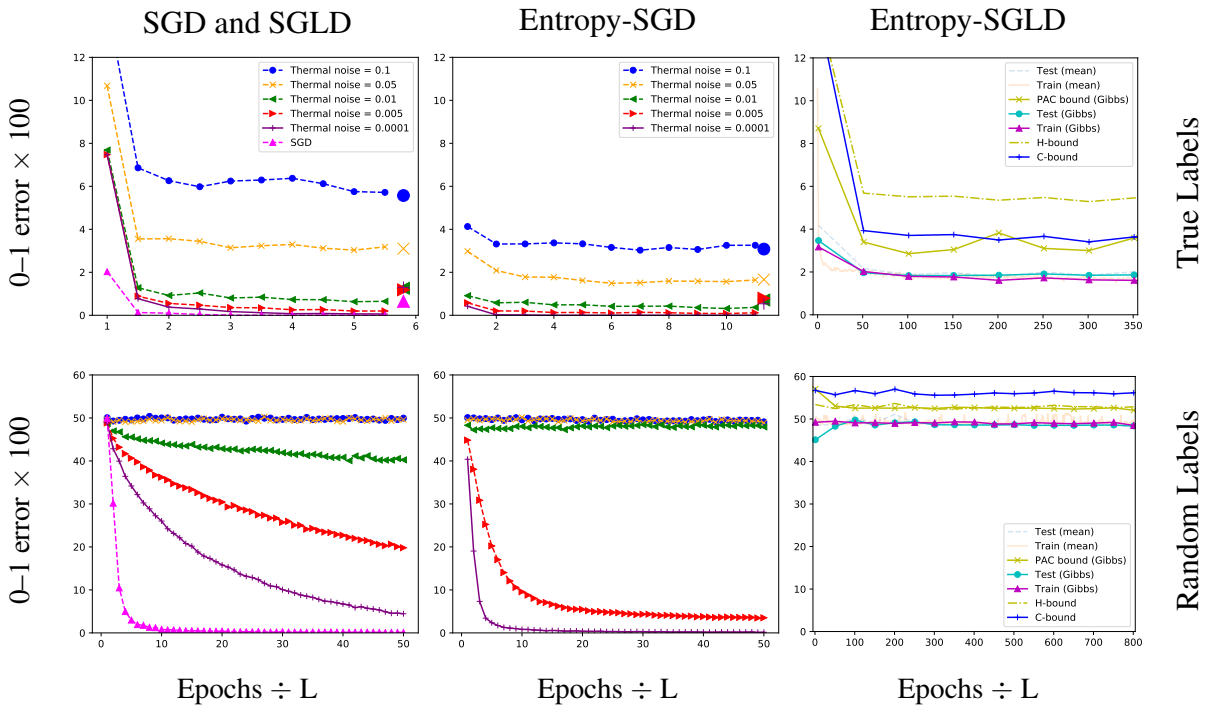


Fig. 4.1 Results on the CONV network on two-class MNIST. **(left column)** Training error (under 0–1 loss) for SGLD on the empirical risk $-\tau\hat{R}_S$ under a variety of thermal noise $\sqrt{2/\tau}$ settings. SGD corresponds to zero thermal noise. **(top-left)** The large markers on the right indicate test error. The gap is an estimate of the generalization error. On true labels, SGLD finds classifiers with relatively small generalization error. At low thermal noise settings, SGLD (and its zero limit, SGD), achieve small empirical risk. As we increase the thermal noise, the empirical 0–1 error increases, but the generalization error decreases. At 0.1 thermal noise, risk is close to 50%. **(bottom-left)** On random labels, SGLD has high generalization error for thermal noise values 0.01 and below. (True error is 50%). **(top-middle)** On true labels, Entropy-SGD, like SGD and SGLD, has small generalization error. For the same settings of thermal noise, empirical risk is lower. **(bottom-middle)** On random labels, Entropy-SGD overfits for thermal noise values 0.005 and below. Thermal noise 0.01 produces good performance on both true and random labels. **(right column)** Entropy-SGLD is configured to approximately sample from an ε-differentially private mechanism with $\varepsilon \approx 0.0327$ by setting $\tau = \sqrt{m}$, where m is the number of training samples. **(top-right)** On true labels, the generalization error for networks learned by Entropy-SGLD is close to zero. Generalization bounds are relatively tight. **(bottom-right)** On random label, Entropy-SGLD does not overfit. See Fig. 4.3 for SGLD bounds at same privacy setting.

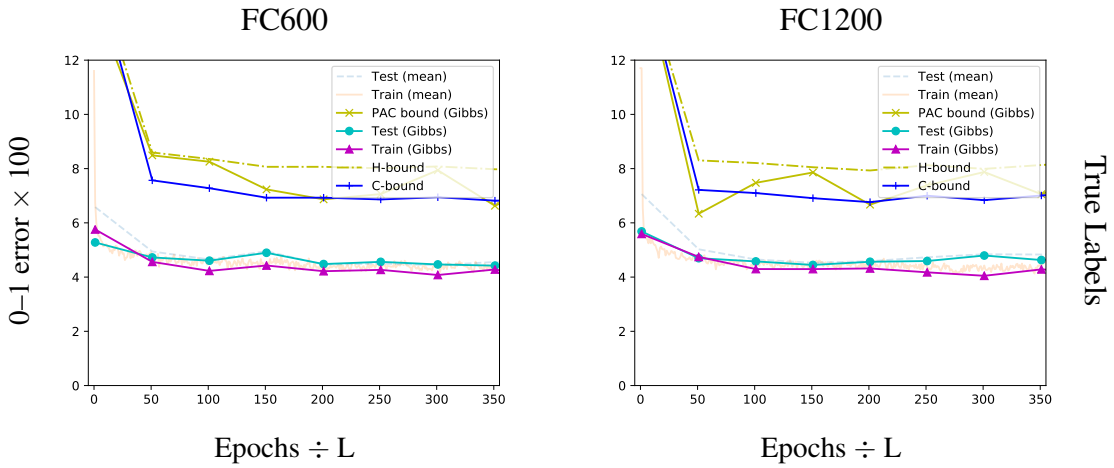


Fig. 4.2 Fully connected networks trained on binarized MNIST with a differentially private Entropy-SGLD algorithm. **(left)** Entropy-SGLD applied to FC600 network trained on true labels. **(right)** Entropy-SGLD applied to FC1200 network trained on true labels. Both training error and generalization error are similar for both network architectures. The true generalization gap is close to zero, since the test and train error overlaps. All the computed bounds on the test error are loose but nonvacuous.

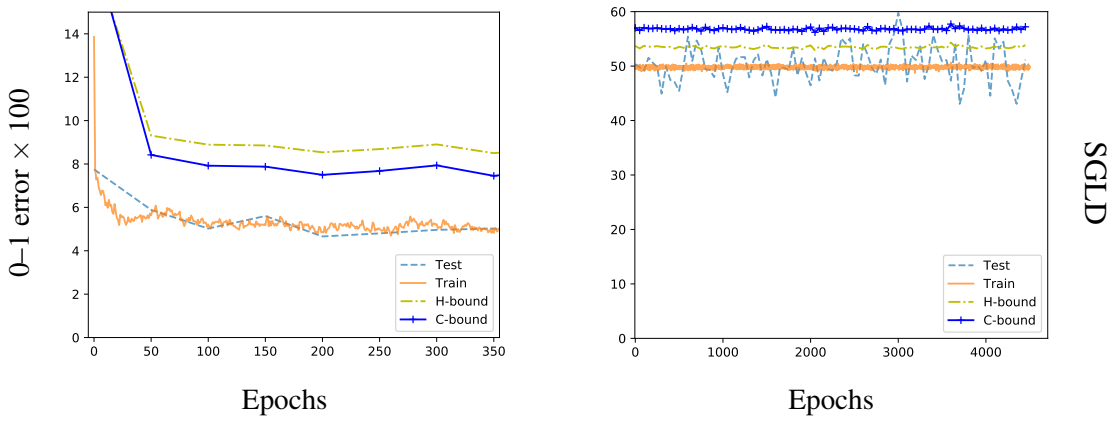


Fig. 4.3 Results on CONV architecture, running SGLD configured to have the same differential privacy as Entropy-SGLD with $\tau = \sqrt{m}$. On true labels, SGLD learns a network with approximately 3% higher training and test error than the mean and Gibbs networks learned by Entropy-SGLD. SGLD does not overfit on random labels, as predicted by theory. The C-bound on the true error of this network is around 8%, which is worse than the roughly 4% C-bound on the mean classifier.

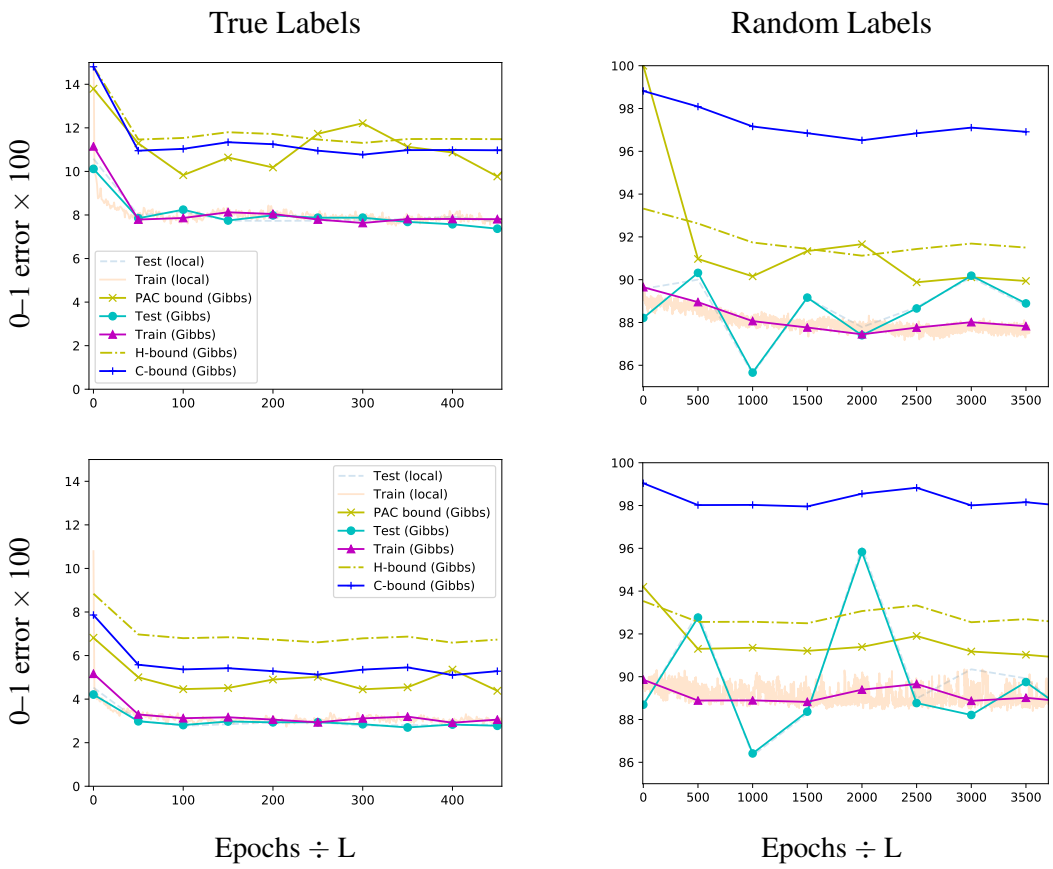


Fig. 4.4 ("Local" here refers to the mean classifier.) Entropy-SGLD results on MNIST. **(top-left)** FC1024 network trained on true labels. The train and test error suggest that the generalization gap is close to zero, while all three bounds exceed the test error by slightly more than 3%. **(bottom-left)** CONV network trained on true labels. Both the train and the test errors are lower than those achieved by the FC1024 network. We still do not observe overfitting. The C-bound and PAC-Bayes bounds exceed the test error by $\approx 3\%$. **(top-right)** FC1024 network trained on random labels. After approximately 1000 epochs, we notice overfitting by $\approx 2\%$. Running Entropy-SGLD further does not cause an additional overfitting. Theory suggests that our choice of τ prevents overfitting via differential privacy. **(bottom-right)** CONV network trained on random labels. We observe almost no overfitting (less than 1%). Both training and test error coincide and remain close to the guessing rate (90%).

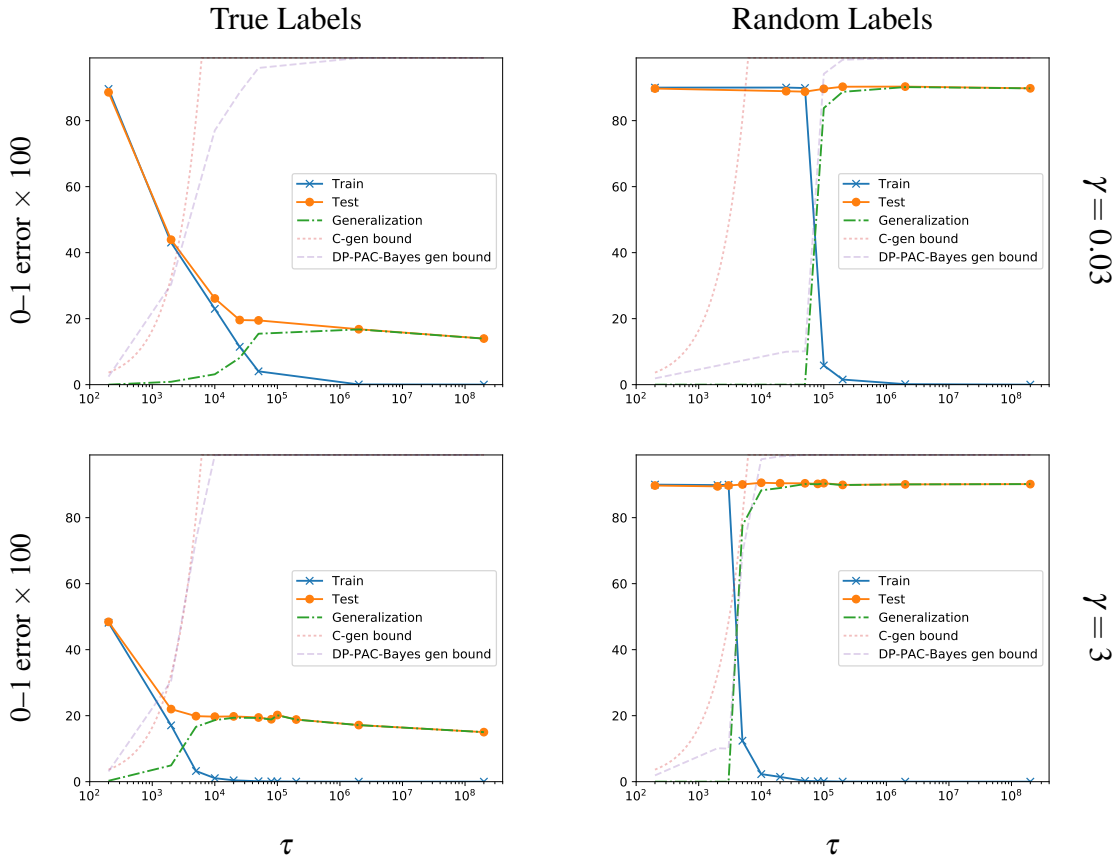


Fig. 4.5 Results for Entropy-SGLD trained on CIFAR10 data, with $\beta = 1$. **(top-left)** For this configuration of parameters, Entropy-SGLD finds good classifiers (with test error lower than 0.3) only at the values of τ for which both generalization bounds are vacuous. Note that, as τ increases, the test error keeps dropping, which cannot be captured by the risk bounds using differential privacy. However, this is only observed for the true label dataset, where the true Bayes error is small. **(top-right)** The generalization gap increases with τ as suggested by the risk bounds, and takes a maximum value for $\tau > 10^6$. **(bottom-left)** The pattern is similar to the $\gamma = 0.03$ case (top left plot). In contrast, Entropy-SGLD finds better classifiers (with test error lower than 0.3) for smaller values of τ. The PAC-Bayes and C-bounds are very close to each other. **(bottom-right)** As in the $\gamma = 0.03$ case (top right plot), we see maximal overfitting for large values of τ. However, Entropy-SGLD starts overfitting at a much lower τ value ($\tau > 2 * 10^2$) compared to the smaller γ case ($\tau > 5 * 10^4$). The PAC-Bayes bound approaches C-bound but no generalization bounds are violated.

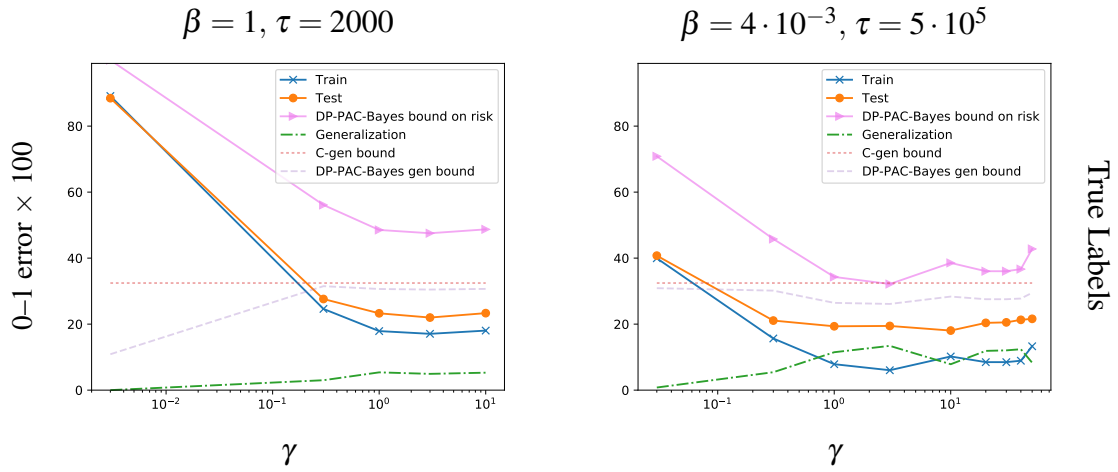


Fig. 4.6 Entropy-SGLD experiments on CIFAR10 data with a fixed level of privacy $\beta\tau = 2000$. The generalization bounds we evaluate do not depend on γ, therefore the C-gen bound takes a constant value. The DP-PAC-Bayes generalization bound corresponds to DP-PAC-Bayes bound on risk minus the empirical error (Train), and is tighter than the C-gen bound for all values of β, τ, γ that we tested. **(left)** β is set to 1 during optimization. A small value of $\gamma = 0.003$ corresponds to high prior variance and thus substantial smoothing of the optimization surface. In this case, Entropy-SGLD fails to find a good classifier. We hypothesize that this happens due to over-smoothing and information loss relating to the location of the actual empirical risk minima. **(right)** β is reduced to 0.004, which allows us to increase τ and maintain the same level of privacy. One can notice that we are able to achieve lower empirical risk (train error is ≈ 0.06 for $\gamma = 3$), risk (≈ 0.19 on the test set), and thus a lower PAC-Bayes bound on the risk (≈ 0.32) compared to the best result for $\beta = 1$ setup. The largest value of γ tested is 50. In this case, the prior variance $(\tau\gamma)^{-1}$ is very small, and the step size is also very small (initial step size is set to base learning rate times the prior variance). This may explain why Entropy-SGLD finds slightly worse classifiers compared to smaller values of γ.

Chapter 5

Training Generative Adversarial Networks with Maximum Mean Discrepancy discriminator

In this chapter, we consider the problem of learning generative models from i.i.d. data with unknown distribution $\mathscr{P}$. We formulate the learning problem as one of finding a function G, called the *generator*, such that, given an input Z drawn from some fixed *noise* distribution $\mathscr{N}$, the distribution of the output $G(Z)$ is close to the data's distribution $\mathscr{P}$. Note that, given G and $\mathscr{N}$, we can easily generate new samples despite not having an explicit representation for the underlying density.

We are particularly interested in the case where the generator is a deep neural network whose parameters we must learn. Rather than being used to classify or predict, these networks transport input randomness to output randomness, thus inducing a distribution. The first direct instantiation of this idea is due to MacKay (1994), although MacKay draws connections even further back to the work of Saund (1989) and others on autoencoders, suggesting that generators can be understood as decoders. MacKay's proposal, called *density networks*, uses multi-layer perceptrons (MLP) as generators and learns the parameters by approximating Bayesian inference.

Since MacKay's proposal, there has been a great deal of progress on learning generative models, especially over high-dimensional spaces like images. Some of the most successful approaches have been based on restricted Boltzmann machines (Salakhutdinov and Hinton, 2009) and deep Boltzmann networks (Hinton and Salakhutdinov, 2006). A recent example is the Neural Autoregressive Density Estimator due to Uria, Murray, and Larochelle (2013). An indepth survey, however, is beyond the scope of this chaptera.

The work in this chapter builds on a proposal due to Goodfellow et al. (2014). Their *generative adversarial nets* framework takes an indirect approach to learning deep generative neural networks: a discriminator network is trained to recognize the difference between training data and generated samples, while the generator is trained to confuse the discriminator. The resulting two-player game is cast as a minimax optimization of a differentiable objective and solved greedily by iteratively performing gradient descent steps to improve the generator and then the discriminator.

Given the greedy nature of the algorithm, Goodfellow et al. (2014) give a careful prescription for balancing the training of the generator and the discriminator. In particular, two gradient steps on the discriminator's parameters are taken for every iteration of the generator's parameters. It is not clear at this point how sensitive this balance is as the data set and network vary. In this chapter, we describe an approximation to adversarial learning that replaces the adversary with a closed-form nonparametric two-sample test statistic based on the Maximum Mean Discrepancy (MMD), which we adopted from the kernel two sample test (Gretton et al., 2012). We call our proposal *MMD nets*.[1] We give bounds on the estimation error incurred by optimizing an empirical estimator rather than the true population MMD and give some illustrations on synthetic and real data.

5.1 Learning to sample as optimization

It is well known that, for any distribution $\mathscr{P}$ and any continuous distribution $\mathscr{N}$ on sufficiently regular spaces $\mathbb{X}$ and $\mathbb{W}$, respectively, there is a function $G : \mathbb{W} \to \mathbb{X}$, such that $G(W) \sim \mathscr{P}$ when $W \sim \mathscr{N}$. (See, e.g., (Kallenberg, 2006, Lem. 3.22).) In other words, we can transform an input from a fixed input distribution $\mathscr{N}$ through a deterministic function, producing an output whose distribution is $\mathscr{P}$. For a given family $\{G_\theta\}$ of functions $\mathbb{W} \to \mathbb{X}$, called *generators*, we can cast the problem of learning a generative model as an optimization

$$\arg\min_\theta \delta(\mathscr{P}, G_\theta(\mathscr{N})), \tag{5.1}$$

where δ is some measure of discrepancy and $G_\theta(\mathscr{N})$ is the distribution of $G_\theta(W)$ when $W \sim \mathscr{N}$. In practice, we only have i.i.d. samples $X_1, X_2, \ldots$ from $\mathscr{P}$, and so we optimize an empirical estimate of $\delta(\mathscr{P}, G_\theta(\mathscr{N}))$.

[1] In independent work reported in a recent preprint, Li, Swersky, and Zemel (Li, Swersky, and Zemel, 2015) also propose to use MMD as a training objective for generative neural networks. We leave a comparison to future work.

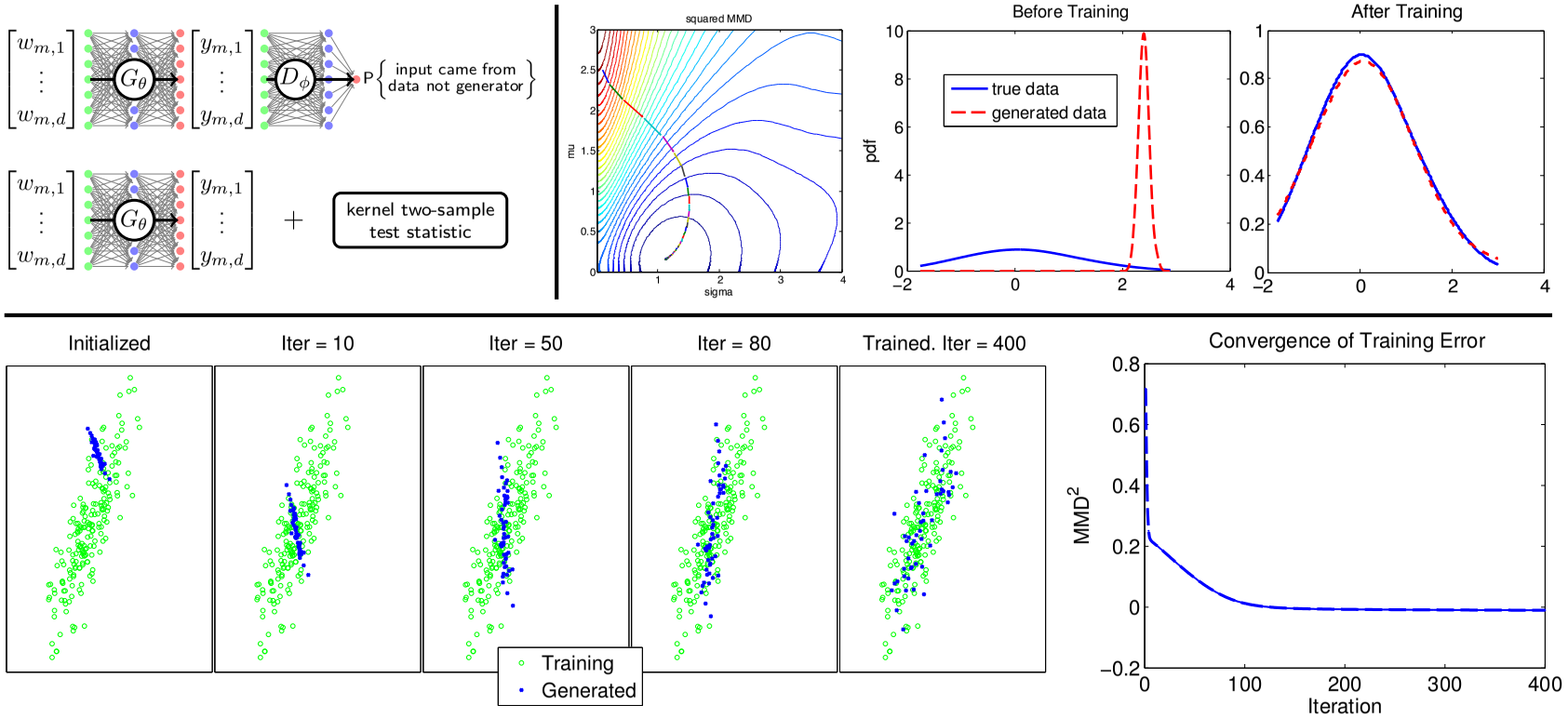


Fig. 5.1 **(top left)** Comparison of adversarial nets and MMD nets. **(top right)** Here we present a simple one-dimensional illustration of optimizing a generator via MMD. Both the training data and noise data are Gaussian distributed and we consider the class of generators given by $G_{(\mu,\sigma)}(w) = \mu + \sigma w$. The plot on the left shows the isocontours of the MMD-based cost function and the path taken by gradient descent. On right, we show the distribution of the generator before and after a number of training iterations, as compared with the data generating distribution. Here we did not resample the generated points and so we do not expect to be able to drive the MMD to zero and match the distribution exactly. **(bottom)** The same procedure is repeated here for a two-dimensional dataset. On the left, we see the gradual alignment of the Gaussian-distributed input data to the Gaussian-distributed output data as the parameters of the generator G_θ are optimized. The learning curve on the right shows the decrease in MMD obtained via gradient descent.

5.1.1 Adversarial Nets

Adversarial nets (Goodfellow et al., 2014) can be cast within this framework: Let $\{D_\phi\}$ be a family of functions $\mathbb{X} \to [0,1]$, called *discriminators*. We recover the adversarial nets objective with the discrepancy

$$\delta_{\mathrm{AN}}(\mathscr{P}, G_\theta(\mathscr{N})) = \max_\phi \mathbb{E}\big[\log D_\phi(X) + \log(1 - D_\phi(Y))\big], \tag{5.2}$$

where $X \sim \mathscr{P}$ and $Y \sim G_\theta(\mathscr{N})$. In this case, Eq. (5.1) becomes

$$\min_\theta \max_\phi V(G_\theta, D_\phi) \tag{5.3}$$

where

$$V(G_\theta, D_\phi) = \mathbb{E}\big[\log D_\phi(X) + \log(1 - D_\phi(G_\theta(W)))\big] \tag{5.4}$$

for $X \sim \mathscr{P}$ and $W \sim \mathscr{N}$. The output of the discriminator D_ϕ can be interpreted as the probability it assigns to its input being drawn from $\mathscr{P}$, and so $V(G_\theta, D_\phi)$ is the expected log loss incurred when classifying the origin of a point equally likely to have been drawn from $\mathscr{P}$ or $G_\theta(\mathscr{N})$. Therefore, optimizing ϕ maximizes the probability of distinguishing samples from $\mathscr{P}$ and $G_\theta(\mathscr{N})$. Assuming that the optimal discriminator exists for every θ, the optimal generator G is that whose output distribution is closest to $\mathscr{P}$, as measured by the Jensen–Shannon divergence, which is minimized when $G_\theta(\mathscr{N}) = \mathscr{P}$.

In (Goodfellow et al., 2014), the generators G_θ and discriminators D_ϕ are chosen to be multilayer perceptrons (MLP). In order to find a minimax solution, they propose taking alternating gradient steps along D_ϕ and G_θ. Note that the composition $D_\phi(G_\theta(\cdot))$ that appears in the value function is yet another (larger) MLP. This fact permits the use of the back-propagation algorithm to take gradient steps.

5.1.2 MMD as an adversary

In their paper introducing adversarial nets, Goodfellow et al. (2014) remark that a balance must be struck between optimizing the generator and optimizing the discriminator. In particular, the authors suggest k maximization steps for every one minimization step to ensure that D_ϕ is well synchronized with G_θ during training. A large value for k, however, can lead to overfitting. In their experiments, for every step taken along the gradient with

respect to G_θ, they take two gradient steps with respect to D_ϕ to bring D_ϕ closer to the desired optimum (Goodfellow, pers. comm.).

It is unclear how sensitive this balance is. Regardless, while adversarial networks deliver impressive sampling performance, the optimization takes approximately 7.5 hours to train on the MNIST dataset running on a GeForce GTX TITAN GPU from nVidia with 6GB RAM. Can we potentially speed up the process with a more tractable choice of adversary?

Our proposal is to replace the adversary with the kernel two-sample test introduced by Gretton et al. (2012). In particular, we replace the family of discriminators with a family $\mathscr{H}$ of test functions $\mathbb{X} \to \mathbb{R}$, closed under negation, and use the maximum mean discrepancy between $\mathscr{P}$ and $G_\theta(\mathscr{N})$ over $\mathscr{H}$, given by

$$\delta_{\mathrm{MMD}_{\mathscr{H}}}(\mathscr{P}, G_\theta(\mathscr{N})) = \sup_{f \in \mathscr{H}} \mathbb{E}[f(X)] - \mathbb{E}[f(Y)], \tag{5.5}$$

where $X \sim \mathscr{P}$ and $Y \sim G_\theta(\mathscr{N})$. See Fig. 5.1 for a comparison of the architectures of adversarial and MMD nets.

While Eq. (5.5) involves a maximization over a family of functions, Gretton et al. (2012) show that it can be solved in closed form when $\mathscr{H}$ is a reproducing kernel Hilbert space (RKHS).

More carefully, let $\mathscr{H}$ be a reproducing kernel Hilbert space (RKHS) of real-valued functions on Ω and let $\langle \cdot, \cdot \rangle_{\mathscr{H}}$ denote its inner product. By the reproducing property it follows that there exists a *reproducing kernel* $k \in \mathscr{H}$ such that every $f \in \mathscr{H}$ can be expressed as

$$f(x) = \langle f, k(\cdot, x) \rangle_{\mathscr{H}} = \sum \alpha_i k(x, x_i) \tag{5.6}$$

The functions *induced by a kernel* k are those functions in the closure of the span of the set $\{k(\cdot, x) : x \in \Omega\}$, which is necessarily an RKHS. Note, that for every positive definite kernel there is a unique RKHS $\mathscr{H}$ such that every function in $\mathscr{H}$ satisfies Eq. (5.6).

Assume that $\mathbb{X}$ is a nonempty compact metric space and $\mathscr{F}$ a class of functions $f : \mathbb{X} \to \mathbb{R}$. Let p and q be Borel probability measures on $\mathbb{X}$, and let X and Y be random variables with distribution p and q, respectively. The *maximum mean discrepancy* (MMD) between p and q is

$$\mathrm{MMD}(\mathscr{F}, p, q) = \sup_{f \in \mathscr{F}} \mathbb{E}[f(X)] - \mathbb{E}[f(Y)]$$

If $\mathscr{F}$ is chosen to be an RKHS $\mathscr{H}$, then

$$\mathrm{MMD}^2(\mathscr{F}, p, q) = \|\mu_p - \mu_q\|^2_{\mathscr{H}}$$

where $\mu_p \in \mathscr{H}$ is the *mean embedding* of p, given by

$$\mu_p = \int_{\mathbb{X}} k(x, \cdot)\, p(\mathrm{d}x) \in \mathscr{H}$$

and satisfying, for all $f \in \mathscr{H}$,

$$\mathbb{E}[f(X)] = \langle f, \mu_p \rangle_{\mathscr{H}}. \tag{5.7}$$

The properties of $\mathrm{MMD}(\mathscr{H}, \cdot, \cdot)$ depend on the underlying RKHS $\mathscr{H}$. For our purposes, it suffices to say that if we take $\mathbb{X}$ to be $\mathbb{R}^D$ and consider the RKHS $\mathscr{H}$ induced by Gaussian or Laplace kernels, then MMD is a metric, and so the minimum of our learning objective is achieved uniquely by $\mathscr{P}$, as desired. (For more details, see Sriperumbudur et al. (2008).)

In practice, we often do not have access to p or q. Instead, we are given independent i.i.d. data $X, X', X_1, \ldots, X_N$ and $Y, Y', Y_1, \ldots, Y_M$ fom p and q, respectively, and would like to estimate the MMD. Gretton et al. (2012) showed that

$$\mathrm{MMD}^2[\mathscr{H}, p, q] = \mathbb{E}[k(X, X') - 2k(X, Y) + k(Y, Y')]$$

and then proposed an unbiased estimator

$$\begin{aligned} \mathrm{MMD}^2_u[\mathscr{H}, X, Y] = &\frac{1}{N(N-1)} \sum_{n \neq n'} k(x_n, x_{n'}) \\ &+ \frac{1}{M(M-1)} \sum_{m \neq m'} k(y_m, y_{m'}) \\ &- \frac{2}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} k(x_n, y_m). \end{aligned} \tag{5.8}$$

5.2 MMD nets

With an unbiased estimator of the MMD objective in hand, we can now define our proposal, *MMD nets*: Fix a neural network G_θ, where θ represents the parameters of the network. Let $W = (w_1, \ldots, w_M)$ denote noise inputs drawn from $\mathscr{N}$, let $Y_\theta = (y_1, \ldots, y_m)$ with $y_j = G_\theta(w_j)$ denote the noise inputs transformed by the network G_θ, and let $X = (x_1, .., x_N)$ denote the

Algorithm 3 Stochastic gradient descent for MMD nets.

Initialize M, θ, α, k
Randomly divide training set X into N_{mini} mini batches
for $i \leftarrow 1,$number-of-iterations **do**
  Regenerate noise inputs $\{w_i\}_{i=1,\ldots,M}$ every r iterations
  for $n_{\text{mini}} \leftarrow 1, N_{\text{mini}}$ **do**
    for $m \leftarrow 1, M$ **do**
      $y_m \leftarrow G_\theta(w_m)$
    compute the n'th minibatch's gradient $\nabla C^{(n)}$
    update learning rate α (e.g., RMSPROP)
    $\theta \leftarrow \theta - \alpha \nabla C_n$

training data in $\mathbb{R}^D$. Given a positive definite kernel k on $\mathbb{R}^D$, we minimize $C(Y_\theta, X)$ as a function of θ, where

$$C(Y_\theta, X) = \frac{1}{M(M-1)} \sum_{m \neq m'} k(y_m, y_{m'}) - \frac{2}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} k(y_m, x_n).$$

Note that $C(Y_\theta, X)$ is composed of only those parts of the unbiased estimator (Eq. (5.8)) that depend on θ.

In practice, the minimization is solved by gradient descent, possibly on subsets of the data. More carefully, the chain rule gives us

$$\nabla C(Y_\theta, X) = \frac{1}{N} \sum_{n=1}^{N} \sum_{m=1}^{M} \frac{\partial C_n(Y_\theta, X_n)}{\partial y_m} \frac{\partial G_\theta(w_m)}{\partial \theta},$$

where

$$C_n(Y_\theta, X_n) = \frac{1}{M(M-1)} \sum_{m \neq m'} k(y_m, y_{m'}) - \frac{2}{M} \sum_{m=1}^{M} k(y_m, x_n).$$

Each derivative $\frac{\partial C_n(Y_\theta, X_n)}{\partial y_m}$ is easily computed for standard kernels like the RBF kernel. Our gradient $\nabla C(Y_\theta, X_n)$ depends on the partial derivatives of the generator with respect to its parameters, which we can compute using back propagation.

5.3 MMD generalization bounds

MMD nets operate by minimizing an empirical estimate of the MMD. This estimate is subject to Monte Carlo error and so the network weights (parameters) $\hat{\theta}$ that are found to minimize the empirical MMD may do a poor job at minimizing the exact population MMD. We show that, for sufficiently large data sets, this estimation error is bounded, despite the space of parameters θ being continuous and high dimensional.

Let Θ denote the space of possible parameters for the generator G_θ, let $\mathscr{N}$ be the distribution on $\mathscr{W}$ for the noisy inputs, and let $p_\theta = G_\theta(\mathscr{N})$ be the distribution of $G_\theta(W)$ when $W \sim \mathscr{N}$ for $\theta \in \Theta$. Let $\hat{\theta}$ be the value optimizing the unbiased empirical MMD estimate, i.e.,

$$\mathrm{MMD}_u^2(\mathscr{H}, X, Y_{\hat{\theta}}) = \inf_\theta \mathrm{MMD}_u^2(\mathscr{H}, X, Y_\theta), \tag{5.9}$$

and let θ^* be the value optimizing the population MMD, i.e.,

$$\mathrm{MMD}^2(\mathscr{H}, p_{\mathrm{data}}, p_{\theta^*}) = \inf_\theta \mathrm{MMD}^2(\mathscr{H}, p_{\mathrm{data}}, p_\theta).$$

We are interested in bounding the difference

$$\mathrm{MMD}^2(\mathscr{H}, p_{\mathrm{data}}, p_{\hat{\theta}}) - \mathrm{MMD}^2(\mathscr{H}, p_{\mathrm{data}}, p_{\theta^*}).$$

To that end, for a measured space $\mathscr{X}$, write $L_\infty(\mathscr{X})$ for the space of essentially bounded functions on $\mathscr{X}$ and write $B(L_\infty(\mathscr{X}))$ for the unit ball under the sup norm, i.e.,

$$B(L_\infty(\mathscr{X})) = \{f\colon \mathscr{X} \to \mathbb{R} : (\forall x \in \mathscr{X}) f(x) \in [-1,1]\}. \tag{5.10}$$

The bounds we obtain will depend on a notion of complexity captured by the fat-shattering dimension:

Definition 5.3.1 (Fat-shattering (Mendelson, 2003))**.** Let $\mathscr{X}_N = \{x_1, \ldots, x_N\} \subset \mathscr{X}$ and $\mathscr{F} \subset B(L_\infty(\mathscr{X}))$. For every $\varepsilon > 0$, $\mathscr{X}_N$ is said to be *ε-shattered by $\mathscr{F}$* if there is some function $h\colon \mathscr{X} \to \mathbb{R}$, such that for every $I \subset \{1, \ldots, N\}$ there is some $f_I \in \mathscr{F}$ for which

$$f_I(x_n) \geq h(x_n) + \varepsilon \text{ if } n \in I, \tag{5.11}$$

$$f_I(x_n) \leq h(x_n) - \varepsilon \text{ if } n \notin I. \tag{5.12}$$

For every ε, the *fat-shattering dimension of* $\mathscr{F}$, written $\mathrm{fat}_\varepsilon(\mathscr{F})$, is defined as

$$\mathrm{fat}_\varepsilon(\mathscr{F}) = \sup\{|\mathscr{X}_N| : \mathscr{X}_N \subset \mathscr{X}, \mathscr{X}_N \text{ is } \varepsilon\text{-shattered by } \mathscr{F}\}. \tag{5.13}$$

Consider the class

$$\mathscr{G}_{k+}^{\mathbb{X}} = \{g = k(x, G_\theta(\cdot)) : x \in \mathbb{X}, \theta \in \Theta\}$$

of functions from $\mathscr{W}$ to $\mathbb{R}$ that are compositions of some generator and the kernel with some fixed input, and the (sub)class

$$\mathscr{G}_{k+} = \{g = k(G_\theta(w), G_\theta(\cdot)) : w \in \mathscr{W}, \theta \in \Theta\}.$$

We then have the following bound on the estimation error:

Theorem 5.3.2 (estimation error)**.** *Assume the kernel is bounded by one and that there exists $\gamma_1, \gamma_2 > 1$ and $p_1, p_2 \in \mathbb{N}$ such that, for all $\varepsilon > 0$, it holds that $fat_\varepsilon(\mathscr{G}_{k+}) \leq \gamma_1 \varepsilon^{-p_1}$ and $fat_\varepsilon(\mathscr{G}_{k+}^X) \leq \gamma_2 \varepsilon^{-p_2}$. Then with probability at least $1-\delta$,*

$$\mathrm{MMD}^2(\mathscr{H}, p_{\text{data}}, p_{\hat{\theta}}) < \mathrm{MMD}^2(\mathscr{H}, p_{\text{data}}, p_{\theta^*}) + \varepsilon,$$

with

$$\varepsilon = r(p_1, \gamma_1, M) + r(p_2, \gamma_2, M-1) + 12M^{-\frac{1}{2}}\sqrt{\log\frac{2}{\delta}},$$

where the rate $r(p, \gamma, M)$ is

$$r(p, \gamma, M) = C_p\sqrt{\gamma}\begin{cases} M^{-\frac{1}{2}} & \text{if } p < 2, \\ M^{-\frac{1}{2}}\log^{\frac{3}{2}}(M) & \text{if } p = 2, \\ M^{-\frac{1}{p}} & \text{if } p > 2, \end{cases}$$

for constants C_{p_1} and C_{p_2} depending on p_1 and p_2 alone.

The proof appears in Section 5.4. We can obtain simpler, but slightly more restrictive, hypotheses if we bound the fat-shattering dimension of the class of generators $\{G_\theta : \theta \in \Theta\}$ alone: Take the observation space $\mathbb{X}$ to be a bounded subset of a finite-dimensional Euclidean space and the kernel to be Lipschitz continuous and translation invariant. For the RBF kernel, the Lipschitz constant is proportional to the inverse of the length-scale: the resulting bound loosens as the length scale shrinks.

5.4 Proofs

We begin with some preliminaries and known results.

Theorem 5.4.1 (McDiarmid's Inequality (Mendelson, 2003)). *Let $f : \mathscr{X}_1 \times \cdots \times \mathscr{X}_N \to \mathbb{R}$ and assume there exists $c_1, \ldots, c_N \geq 0$ such that, for all $k \in \{1, \ldots, N\}$, we have*

$$\sup_{x_1,\ldots,x_k,x_k',\ldots,x_N} |f(x_1,\ldots,x_k,\ldots,x_N) \tag{5.14}$$

$$- f(x_1,\ldots,x_k',\ldots,x_N)| \leq c_k. \tag{5.15}$$

Then, for all $\varepsilon > 0$ and independent random variables $\xi_1, \ldots, \xi_n$ in $\mathscr{X}$,

$$\mathbb{P}\{f(\xi_1,\ldots,\xi_N) - \mathbb{E}(f(\xi_1,\ldots,\xi_N)) \geq \varepsilon)\} \qquad < \exp\left(\frac{-2\varepsilon^2}{\sum_{n=1}^N c_n^2}\right). \tag{5.16}$$

Theorem 5.4.2 ((Mendelson, 2003, Thm. 2.35)). *Let $\mathscr{F} \subset B(L_\infty(\mathscr{X}))$. Assume there exists $\gamma > 1$, such that for all $\varepsilon > 0$, $fat_\varepsilon(\mathscr{F}) \leq \gamma\varepsilon^{-p}$ for some $p \in \mathbb{N}$. Then there exists constants C_p depending on p only, such that $\mathscr{R}_N(\mathscr{F}) \leq C_p\Psi(p,N,\gamma)$ where*

$$\Psi(p,N,\gamma) = \gamma^{\frac{1}{2}} \begin{cases} 1 & \text{if } 0 < p < 2 \\ \log^{\frac{3}{2}} N & \text{if } p = 2 \\ N^{\frac{1}{2}-\frac{1}{p}} & \text{if } p > 2. \end{cases}$$

Theorem 5.4.3 ((Gretton et al., 2012)). *Assume $0 \leq k(x_i,x_j) \leq K$, $M = N$. Then*

$$\mathbb{P}\left[|\mathrm{MMD}_u^2(\mathscr{H},X,Y_\theta) - \mathrm{MMD}^2(\mathscr{H},p_{\mathrm{data}},p_\theta)| > \varepsilon\right] \leq \delta_\varepsilon$$

where

$$\delta_\varepsilon = 2\exp\left(-\frac{\varepsilon^2 M}{16K^2}\right).$$

The case where Θ is a finite set is elementary:

Theorem 5.4.4 (estimation error for finite parameter set). *Let p_θ be the distribution of $G_\theta(W)$, with θ taking values in some finite set $\Theta = \{\theta_1, ..., \theta_T\}, T < \infty$. Then, with probability at least $1 - (T+1)\delta_\varepsilon$, where δ_ε is defined as in Theorem 5.4.3, we have*

$$\mathrm{MMD}^2(\mathscr{H},p_{\mathrm{data}},p_{\hat{\theta}}) < \mathrm{MMD}^2(\mathscr{H},p_{\mathrm{data}},p_{\theta^*}) + 2\varepsilon.$$

Proof. Let $\mathscr{E}(\theta) = \mathrm{MMD}_u^2(\mathscr{H}, X, Y_\theta)$ and let $\mathscr{T}(\theta) = \mathrm{MMD}^2(\mathscr{H}, p_{\mathrm{data}}, p_\theta)$.

Note, that the upper bound stated in Theorem 5.4.3 holds for the parameter value θ^*, i.e.,

$$\mathbb{P}\left[|\mathscr{E}(\theta^*) - \mathscr{T}(\theta^*)| > \varepsilon\right] \leq \delta_\varepsilon. \tag{5.17}$$

Because $\hat{\theta}$ depends on the training data X and generator data Y, we use a uniform bound that holds over all θ. Specifically,

$$\begin{aligned} \mathbb{P}\left[|\mathscr{E}(\hat{\theta}) - \mathscr{T}(\hat{\theta})| > \varepsilon\right] &\leq \mathbb{P}\left[\sup_\theta |\mathscr{E}(\theta) - \mathscr{T}(\theta)| > \varepsilon\right] \\ &\leq \sum_{t=1}^{T} \mathbb{P}\left[|\mathscr{E}(\hat{\theta}) - \mathscr{T}(\hat{\theta})| > \varepsilon\right] \leq T\delta_\varepsilon. \end{aligned} \tag{5.18}$$

This yields that with probability at least $1 - T\delta_\varepsilon$,

$$\begin{aligned} 2\varepsilon &\geq |\mathscr{E}(\hat{\theta}) - \mathscr{T}(\hat{\theta})| + |\mathscr{E}(\theta^*) - \mathscr{T}(\theta^*)| \\ &\geq |\mathscr{E}(\theta^*) - \mathscr{E}(\hat{\theta}) + \mathscr{T}(\hat{\theta}) - \mathscr{T}(\theta^*)|. \end{aligned} \tag{5.19}$$

Since θ^* was chosen to minimize $\mathscr{T}(\theta)$, we know that $\mathscr{T}(\hat{\theta}) \geq \mathscr{T}(\theta^*)$. Similarly, by Eq. (5.9), $\mathscr{E}(\theta^*) \geq \mathscr{E}(\hat{\theta})$. Therefore it follows that

$$\begin{aligned} 2\varepsilon &\geq \mathscr{T}(\hat{\theta}) - \mathscr{T}(\theta^*) \\ &= \mathrm{MMD}^2(\mathscr{H}, p_{\mathrm{data}}, p_{\theta^*}) - \mathrm{MMD}^2(\mathscr{H}, p_{\mathrm{data}}, p_{\hat{\theta}}) \end{aligned}$$

proving the theorem. □

The following result follows immediately from Theorem 5.4.4.

Corollary 5.4.5. *With the definitions from Theorem 5.4.4, with probability at least* $1 - \delta$*,*

$$\mathrm{MMD}^2(\mathscr{H}, p_{\mathrm{data}}, p_{\hat{\theta}}) < \mathrm{MMD}^2(\mathscr{H}, p_{\mathrm{data}}, p_{\theta^*}) + 2\varepsilon_\delta, \tag{5.20}$$

where

$$\varepsilon_\delta = 8K\sqrt{\frac{1}{M}\log\left[2(T+1)\delta\right]}. \tag{5.21}$$

In order to prove the general result, we begin with some technical lemmas. The development here owes much to Gretton et al. (2012).

Lemma 5.4.6. *Let* $\mathscr{F} = \{f : \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}\}$ *and*

$$\mathscr{F}_+ = \{h = f(y,\cdot) : f \in \mathscr{F}, y \in \mathscr{Y}\} \cap B(L_\infty(\mathscr{Y})). \tag{5.22}$$

Let $\{Y_n\}_{n=1}^N$ *be* μ*-distributed independent random variables in* $\mathscr{Y}$*. Assume for some* $\gamma > 1$ *and some* $p \in \mathbb{N}$*, we have* $fat_\varepsilon(\mathscr{F}_+) \leq \gamma\varepsilon^{-p}$*, for all* $\varepsilon > 0$*. For* $y_n \in \mathscr{Y} \quad \forall n = 1,\ldots,N$*, define* $\rho(y_1,\ldots,y_N)$ *to be*

$$\sup_{f\in\mathscr{F}} \left| \mathbb{E}\left(f(Y,Y')\right) - \frac{1}{N(N-1)} \sum_{n\neq n'} f(y_n, y_{n'}) \right|. \tag{5.23}$$

Then there exists a constant C *that depends on* p*, such that*

$$\mathbb{E}\left(\rho(Y_1,\ldots,Y_N)\right) \leq \frac{C}{\sqrt{N-1}} \Psi(\gamma, N-1, p). \tag{5.24}$$

Proof. Let us introduce $\{\zeta_n\}_{n=1}^N$, where ζ_n and $Y_{n'}$ have the same distribution and are independent for all $n,n' \in \{1,\ldots,N\}$. Then the following is true:

$$\mathbb{E}(f(Y,Y')) = \mathbb{E}\Big(\frac{1}{N(N-1)} \sum_{n,n':n\neq n'} f(\zeta_n, \zeta_{n'})\Big) \tag{5.25}$$

Using Jensen's inequality and the independence of Y, Y' and $Y_n, Y_{n'}$, we have

$$\begin{aligned}
&\mathbb{E}\left(\rho(Y_1,\ldots,Y_N)\right) \\
&= \mathbb{E}\Bigg(\sup_{f\in\mathscr{F}} \Bigg| \mathbb{E}(f(Y,Y')) \\
&\qquad\qquad - \frac{1}{N(N-1)} \sum_{n\neq n'} f(Y_m, Y_{m'}) \Bigg| \Bigg) \\
&\leq \mathbb{E}\Bigg(\sup_{f\in\mathscr{F}} \Bigg| \frac{1}{N(N-1)} \sum_{n\neq n'} f(\zeta_n, \zeta_n') \\
&\qquad\qquad - \frac{1}{N(N-1)} \sum_{n\neq n'} f(Y_n, Y_{n'}) \Bigg| \Bigg).
\end{aligned} \tag{5.26}$$

Introducing conditional expectations allows us to rewrite the equation with the sum over n outside the expectations. I.e., Eq. (5.26) equals

$$\begin{aligned}&\frac{1}{N}\sum_n \mathbb{E}\mathbb{E}^{(Y_n,\zeta_n)}\Big(\sup_{f\in\mathscr{F}}\Big|\frac{1}{N-1}\sum_{n\neq n'}\Phi(\zeta_n,\zeta_{n'},Y_n,Y_{n'})\Big|\Big)\\ &\quad=\mathbb{E}\mathbb{E}^{(Y,\zeta)}\Big(\sup_{f\in\mathscr{F}}\Big|\frac{1}{N-1}\sum_{n=1}^{N-1}\sigma_n\Phi(\zeta,\zeta_n,Y,Y_n)\Big|\Big),\end{aligned}\tag{5.27}$$

where $\Phi(x,x',y,y') = f(x,x') - f(y,y')$. The second equality follows by symmetry of random variables $\{\zeta_n\}_{n=1}^{N-1}$. Note that we also added Rademacher random variables $\{\sigma_n\}_{n=1}^{N-1}$ before each term in the sum since $(f(\zeta_n,\zeta_{n'}) - f(Y_n,Y_{n'}))$ has the same distribution as $-(f(\zeta_n,\zeta_{n'}) - f(Y_n,Y_{n'}))$ for all n,n' and therefore the σ's do not affect the expectation of the sum.

Note that ζ_m and Y_m are identically distributed. Thus the triangle inequality implies that Eq. (5.27) is less than or equal to

$$\frac{2}{N-1}\mathbb{E}\left(\mathbb{E}^{(Y)}\left(\sup_{f\in\mathscr{F}}\Big|\sum_{n=1}^{N-1}\sigma_n f(Y,Y_n)\Big|\right)\right)\tag{5.28}$$

$$\leq \frac{2}{\sqrt{N-1}}\mathscr{R}_{N-1}(\mathscr{F}_+),\tag{5.29}$$

where $\mathscr{R}_{N-1}(\mathscr{F}_+)$ is the Rademacher's complexity of $\mathscr{F}_+$. Then by Theorem 5.4.2, we have

$$\mathbb{E}\left(\rho(Y_1,\ldots,Y_N)\right) \leq \frac{C}{\sqrt{N-1}}\Psi(\gamma,N-1,p).$$

□

Lemma 5.4.7. *Let $\mathscr{F} = \{f : \mathscr{X}\times\mathscr{Y}\to\mathbb{R}\}$ and $\mathscr{F}_+ = \{f : x\times\mathscr{Y}\to\mathbb{R}, x\in\mathscr{X}\}$ and assume $\mathscr{F}_+ \subset B(L_\infty(\mathscr{Y}))$. Let $\{X_n\}_{n=1}^N$ and $\{Y_m\}_{n=1}^M$ be ν- and μ-distributed independent random variables in $\mathscr{X}$ and $\mathscr{Y}$, respectively. Assume for some $\gamma > 1$, such that for all $\varepsilon > 0$, $fat_\varepsilon(\mathscr{F}_+) \leq \gamma\varepsilon^{-p}$, for some $p\in\mathbb{N}$. For all $x_n\in\mathscr{X}$, $n\leq N$, and all $y_m\in\mathscr{Y}$, $m\leq M$, define*

$$\rho(x_1,\ldots,x_N,y_1,\ldots,y_M) = \tag{5.30}$$

$$\sup_{f\in\mathscr{F}}\Big|\mathbb{E}(f(X,Y) - \frac{1}{NM}\sum_{n,m}f(x_n,y_m)\Big|.\tag{5.31}$$

Then there exists C that depends on p, such that

$$\mathbb{E}\left(\rho(X_1,\ldots,X_N,Y_1,\ldots,Y_M)\right) \leq \frac{C}{\sqrt{M}}\Psi(\gamma,M,p).\tag{5.32}$$

Proof. The proof is very similar to that of Lemma 5.4.6. □

Proof of Theorem 5.3.2. The proof follows the same steps as the proof of Theorem 5.4.4 apart from a stronger uniform bound stated in Eq. (5.18). I.e., we need to show:

$$\mathbb{P}\left[\sup_{\theta\in\Theta}|\mathscr{E}(\theta)-\mathscr{T}(\theta)|\geq\varepsilon\right]\leq\delta.$$

Expanding MMD as defined by Eq. (5.8), and substituting $Y=G_\theta(W)$, yields

$$\begin{aligned}
&\sup_{\theta\in\Theta}|\mathscr{E}(\theta)-\mathscr{T}(\theta)|\\
&=\sup_{\theta\in\Theta}\Big|\mathbb{E}(k(X,X'))\\
&\qquad-\frac{1}{N(N-1)}\sum_{n'\neq n}k(X_n,X_{n'})\\
&\qquad+\mathbb{E}(k(G_\theta(W),G_\theta(W')))\\
&\qquad-\frac{1}{M(M-1)}\sum_{m\neq m'}k(G_\theta(W_m),G_\theta(W_{m'}))\\
&\qquad-2\mathbb{E}(k(X,G_\theta(W)))\\
&\qquad+\frac{2}{MN}\sum_{m,n}k(X_n,G_\theta(W_m))\Big|.
\end{aligned}\tag{5.33}$$

For all $n\in\{1,\ldots,N\}$, $k(X_n,X_{n'})$ does not depend on θ and therefore the first two terms of the equation above can be taken out of the supremum. Also, note that since $|k(\cdot,\cdot)|\leq K$, we have

$$\Big|\zeta(x_1,\ldots,x_n,\ldots,x_N)-\zeta(x_1,\ldots,x'_n,\ldots,x_N)\Big|\leq\frac{2K}{N},\tag{5.34}$$

where

$$\zeta(x_1,\ldots,x_N)=\frac{1}{N(N-1)}\sum_{n,n':n'\neq n}k(x_n,x_{n'}),\tag{5.35}$$

and ζ is an unbiased estimate of $\mathbb{E}(k(X,X'))$. Then from McDiarmid's inequality on ζ, we have

$$\begin{aligned}&\mathbb{P}\left(\left|\mathbb{E}(k(X,X')) - \frac{1}{N(N-1)}\sum_{n'\neq n} k(X_n,X_{n'})\right| \geq \varepsilon\right)\\ &\qquad \leq \exp\left(-\frac{\varepsilon^2}{2K^2}N\right).\end{aligned} \tag{5.36}$$

Therefore Eq. (5.33) is bounded by the sum of the bound on Eq. (5.36) and the following:

$$\begin{aligned}\sup_{\theta\in\Theta}\Big|&\mathbb{E}(k(G_\theta(W),G_\theta(W')))\\ &- \frac{1}{M(M-1)}\sum_{m\neq m'} k(G_\theta(W_m),G_\theta(W_{m'}))\\ &- 2\mathbb{E}(k(X,G_\theta(W)))\\ &+ \frac{2}{MN}\sum_{m,n} k(X_n,G_\theta(W_m))\Big|.\end{aligned} \tag{5.37}$$

Thus the next step is to find the bound for the supremum above.

Define

$$f(W_1,\ldots,W_M;p_{\text{noise}}) = f(\underline{W}_M) \tag{5.38}$$

$$= \sup_{\theta\in\Theta}\Big|\mathbb{E}(k(G_\theta(W),G_\theta(W'))) \tag{5.39}$$

$$- \frac{1}{M(M-1)}\sum_{m\neq m'} k(G_\theta(W_m),G_\theta(W_{m'}))\Big| \tag{5.40}$$

and

$$h(X_1,\ldots,X_N,W_1,\ldots,W_M;p_{\text{data}},p_{\text{noise}}) \tag{5.41}$$

$$= h(\underline{X}_N,\underline{W}_M) \tag{5.42}$$

$$= \sup_{\theta\in\Theta}\Big|\frac{1}{MN}\sum_{m,n} k(X_n,G_\theta(W_m)) - \mathbb{E}(k(X,G_\theta(W)))\Big|. \tag{5.43}$$

Then by triangle inequality, the supremum in Eq. (5.37) is bounded by

$$f(\underline{W}_M) + 2h(\underline{X}_N,\underline{W}_M).$$

We will first find the upper bound on $f(\underline{W}_M)$, i.e., for every $\varepsilon > 0$, we will show that there exists δ_f, such that

$$\mathbb{P}\left(f(\underline{W}_M) > \varepsilon\right) \leq \delta_f \tag{5.44}$$

For each $m \in \{1, \ldots, M\}$,

$$\Big| f(W_1, \ldots, W_m, \ldots, W_M) \\ - f(W_1, \ldots, W'_m, \ldots, W_M) \Big| \leq \frac{2K}{M}$$

since the kernel is bounded by K, and therefore $k(G_\theta(W_m), G_\theta(W_{m'}))$ is bounded by K for all m. The conditions of Theorem 5.4.1 are satisfied and thus we can use McDiarmid's Inequality on f:

$$\mathbb{P}\left(f(\underline{W}_M) - \mathbb{E}(f(\underline{W}_M)) \geq \varepsilon\right) \leq \exp\left(-\frac{\varepsilon^2 M}{2K^2}\right).$$

Define

$$\mathscr{G}_k = \{k(G_\theta(\cdot), G_\theta(\cdot)) : \theta \in \Theta\}$$

To show Eq. (5.44), we need to bound the expectation of f. We can apply Lemma 5.4.6 on the function classes $\mathscr{G}_k$ and $\mathscr{G}_{k+}$. The resulting bound is

$$\mathbb{E}(f(\underline{W}_M)) \leq \varepsilon_{p1} = \frac{C_f}{\sqrt{M-1}} \Psi(\gamma_1, M-1, p_1), \tag{5.45}$$

where p_1 and γ_1 are parameters associated with fat shattering dimension of $\mathscr{G}_{k+}$ as stated in the assumptions of the theorem, and C_f is a constant depending on p_1.

Now we can write down the bound on f:

$$\mathbb{P}\left(f(\underline{W}_M) \geq \varepsilon_{p_1} + \varepsilon\right) \leq \exp\left(-\frac{\varepsilon^2 M}{2K^2}\right) = \delta_f. \tag{5.46}$$

Similarly, $h(\underline{X}_N, \underline{W}_M)$ has bounded differences:

$$\Big| h(X_1, \ldots, X_n, \ldots, X_N, W_1, \ldots, W_M) \\ - h(X_1, \ldots, X_{n'}, \ldots, X_N, W_1, \ldots, W_M) \Big| \leq \frac{2K}{N}$$

and

$$\Big|h(X_1,\ldots,X_N,W_1,\ldots,W_m,\ldots,W_M) - h(X_1,\ldots,X_N,W_1,\ldots,W_{m'},\ldots,W_M)\Big| \leq \frac{2K}{M}.$$

McDiarmid's inequality then implies

$$\mathbb{P}\left(h(\underline{X}_N,\underline{W}_M) - \mathbb{E}(h(\underline{X}_N,\underline{W}_M) \geq \varepsilon\right) \leq \exp\left(-\frac{\varepsilon^2}{2K^2}\frac{NM}{N+M}\right). \tag{5.47}$$

We can bound expectation of $h(\underline{X}_N,\underline{W}_M)$ using Lemma 5.4.7 applied on $\mathscr{G}_k^{\mathbb{X}}$ and $\mathscr{G}_{k+}^{\mathbb{X}}$, where

$$\mathscr{G}_k^{\mathbb{X}} = \{k(\cdot, G_\theta(\cdot)) : \theta \in \Theta\}.$$

Then

$$\mathbb{E}(h(\underline{X}_N,\underline{W}_M)) \leq \varepsilon_{p_2} = \frac{C_h}{\sqrt{M}}\Psi(\gamma_2, M, p_2). \tag{5.48}$$

for some constant C_h that depends on $p_@$. The final bound on h is then

$$\mathbb{P}\left(h(\underline{X}_N,\underline{W}_M) \geq \varepsilon_{p_2} + \varepsilon\right) \leq \exp\left(-\frac{\varepsilon^2}{2K^2}\frac{NM}{N+M}\right) = \delta_h.$$

Summing up the bounds from Eq. (5.46) and Eq. (5.47), it follows that

$$\mathbb{P}\left(f(\underline{W}_M) + 2h(\underline{X}_N,\underline{W}_M) \geq \varepsilon_{p1} + 2\varepsilon_{p_2} + 3\varepsilon\right) \leq \max(\delta_f, \delta_h) = \delta_h.$$

Using the bound in Eq. (5.36), we have obtain the uniform bound we were looking for:

$$\mathbb{P}\left[\sup_{\theta\in\Theta}|\mathscr{E}(\theta) - \mathscr{T}(\theta)| > \varepsilon_{p_1} + 2\varepsilon_{p_2} + 4\varepsilon\right] \leq \delta_h,$$

which by Eq. (5.18) yields

$$\mathbb{P}\left[|\mathscr{E}(\hat{\theta}) - \mathscr{T}(\hat{\theta})| > \varepsilon_{p_1} + 2\varepsilon_{p_2} + 4\varepsilon\right] \leq \delta_h.$$

Since it was assumed that $K = 1$ and $N = M$, we get $\delta_h = \exp\left(-\varepsilon^2 M/4\right)$.

To finish, we proceed as in the proof of Theorem 5.4.4. We can rearrange some of the terms to get a different form of Eq. (5.17):

$$\mathbb{P}\left[|\mathscr{E}(\theta^*) - \mathscr{T}(\theta^*)| > 2\varepsilon\right] \leq 2\exp\left(-\frac{\varepsilon^2 M}{4}\right) = 2\delta_h. \tag{5.49}$$

All of the above implies that for any $\varepsilon > 0$, there exists δ, such that

$$\mathbb{P}\big(\mathrm{MMD}^2(\mathscr{H}, p_{\text{data}}, p_{\hat{\theta}}) - \mathrm{MMD}^2(\mathscr{H}, p_{\text{data}}, p_{\theta^*}) \geq \varepsilon\big) \leq \delta, \tag{5.50}$$

where

$$\varepsilon = \varepsilon_{p_1} + 2\varepsilon_{p_2} + \frac{12}{\sqrt{M}}\sqrt{\log\frac{2}{\delta}}.$$

We can rewrite ε as:

$$\varepsilon = r(p_1, \gamma_1, M) + r(p_2, \gamma_2, M-1) + 12M^{-\frac{1}{2}}\sqrt{\log\frac{2}{\delta}},$$

The rate $r(p, \gamma, N)$ is given by Eq. (5.45) and Eq. (5.48):

$$r(p, \gamma, M) = C_p\sqrt{\gamma}\begin{cases} M^{-\frac{1}{2}} & \text{if } p < 2, \\ M^{-\frac{1}{2}}\log^{\frac{3}{2}}(M) & \text{if } p = 2, \\ M^{-\frac{1}{p}} & \text{if } p > 2, \end{cases}$$

where C_{p_1} and C_{p_2} depend on p_1 and p_2 alone. □

We close by noting that the approximation error is zero in the nonparametric limit.

Theorem 5.4.8 (Gretton et al. (2012))**.** *Let F be the unit ball in a universal RKHS $\mathscr{H}$, defined on the compact metric space $\mathbb{X}$, with associated continuous kernel $k(\cdot,\cdot)$. Then* $\mathrm{MMD}[\mathscr{H}, p, q] = 0$ *if and only if* $p = q$.

Corollary 5.4.9 (approximation error)**.** *Assume p_{data} is in the family $\{p_\theta\}$ and that $\mathscr{H}$ is an RKHS induced by a characteristic kernel. Then*

$$\inf_\theta \mathrm{MMD}(\mathscr{H}, p_{\text{data}}, p_\theta) = 0$$

and the infimum is achieved at θ satisfying $p_\theta = p_{\text{data}}$.

Proof. By Theorem 5.4.8, it follows that $\mathrm{MMD}^2(\mathscr{H},\cdot,\cdot)$ is a metric. The result is then immediate. □

5.5 Empirical evaluation

In this section, we demonstrate the approach on an illustrative synthetic example as well as the standard MNIST digits and Toronto Face Dataset (TFD) benchmarks. We show that MMD-based optimization of the generator rapidly delivers a generator that produces recognizable samples, but these samples are inferior to those produced by adversarial networks, both visually and as measured by an estimate of the mean log density on a held-out test set.

5.5.1 Gaussian data, kernel, and generator

Under an RBF kernel and Gaussian generator with parameters $\theta = \{\mu, \sigma\}$, it is straightforward to find the gradient of $C(Y_\theta, X)$ by applying the chain rule. Using fixed random standard normal numbers $\{w_1, ..., w_M\}$, we have $y_m = \mu + \sigma w_m$ for $m \in \{1, .., M\}$. The result of these illustrative synthetic experiments can be found in Fig. 5.1. The dataset consisted of $N = 200$ samples from a Gaussian $\{\mu, \sigma\} = \{0, 0\}$ and $M = 50$ noise input samples were generated also from a Gaussian with $\{\mu, \sigma\} = \{0, 0\}$ with a fixed random seed. The algorithm was initialized at values $\{\mu, \sigma\} = \{2.5, 0.1\}$. We fixed the learning rate to 0.5 and ran gradient descent steps for $K = 250$ iterations.

5.5.2 MNIST digits

We evaluated MMD nets on MNIST digits (Lecun et al., 1998). The generator was chosen to be a fully connected, 3 hidden layer neural network with sigmoidal activation functions. Following Gretton et al. (2012), we used a radial basis function (RBF) kernel, but also evaluated the rational quadratic (RQ) kernel (Rasmussen and Williams, 2005) and Laplacian kernel, but found that the RBF performed best in the parameter ranges we evaluated. We used Bayesian optimization (WHETLab) to set the bandwidth of the RBF and the number of neurons in each layer on initial test runs of 50,000 iterations. However, one can get a similar-quality generator simply using the median heuristic (Gretton et al., 2012) to set the kernel bandwidth. The learning rate was adjusting during optimization by RMSPROP (Tieleman and Hinton, 2012).

Fig. 5.2 presents the digits learned after 1,000,000 iterations. (Doubling the number of iterations produced similar images.) We performed minibatch stochastic gradient descent, resampling the generated digits every 300 iterations, with minibatches of 500 training and

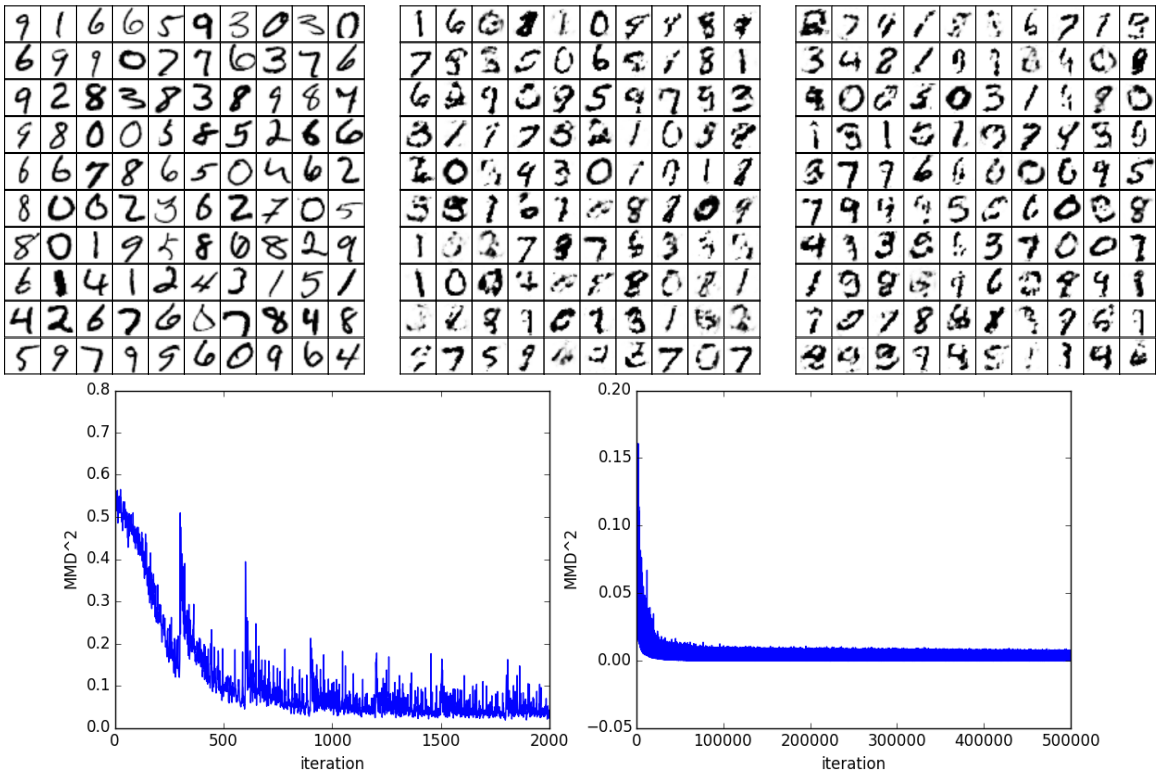


Fig. 5.2 **(top-left)** MNIST digits from the training set. **(top-right)** Newly generated digits produced after 1,000,000 iterations (approximately 5 hours). Despite the remaining artifacts, the resulting kernel-density estimate of the test data was state of the art at the time when MMD nets were introduced. **(top-center)** Newly generated digits after 300 further iterations optimizing the associated empirical MMD. **(bottom-left)** MMD learning curves for first 2000 iterations. **(bottom-right)** MMD learning curves from 2000 to 500,000 iterations. Note the difference in y-axis scale. No appreciable change is seen in later iterations.

generated points. It is clear that the digits produced have many artifacts not appearing in the MNIST data set. Indeed, the mean log density of held-out test data was estimated to be only $\mathbf{113} \pm \mathbf{2}$, as compared with the reported $\mathbf{225} \pm \mathbf{2}$ achieved by adversarial nets. On the other hand, most of the gain is achieved by MMD nets in the first 100-200k iterations, and so perhaps MMD nets could be used to initialize a network further optimized by other means.

5.5.3 Toronto Face dataset

We also evaluated MMD nets on the Toronto face dataset (TFD) Susskind, Anderson, and Hinton, 2010. We used a 3-hidden-layer sigmoidal MLP with similar architecture (1000, 600, and 1000 units) and RBF kernel for the cost function with the same hyper parameter. We used 500 training and generated points per batch. The generated points were resampled every 500 iterations. The network was optimized for 500,000 iterations. Samples from the

Fig. 5.3 **(left)** TFD. **(right)** Faces generated by network trained for 500,000 iterations. **(center)** After an additional 500 iterations.

resulting network are plotted in Fig. 5.3. Again, the samples produced by MMD nets are clearly distinguishable from the training samples and this is reflected in a much lower mean log density than adversarial nets.

5.6 Discussion

MMD offers a closed-form surrogate for the discriminator in the adversarial nets framework. After using Bayesian optimization for the parameters, we found that the network produced samples that were visually similar, but far from indistinguishable from those used to train the network. On one hand, adversarial nets handedly outperformed MMD nets in terms of mean log density. On the other, MMD nets achieve most of their gain quickly and so it seems promising to combine MMD nets with another technique, perhaps using MMD nets to initialize a more costly procedure.

5.7 Recent Follow-up work

A number of authors have extended our work on MMD nets (e.g., Lee et al. (2017), Li et al. (2017), Sutherland et al. (2016), and Bińkowski et al. (2018)).

In the work described in this chapter, MMD replaces the discriminator in GANs learning setup. An alternative use of the MMD statistic is proposed by Sutherland et al. (2016): the authors suggest to use MMD to evaluate the samples obtained from the generator. This novel quadratic MMD test offers an alternative to kernel density estimation. The new test allows one to visually and numerically assess the quality and failures of the generator distribution. Sutherland et al. (2016) also use a lower variance estimate of the MMD statistic and propose how one can compute it efficiently.

Li et al. (2017) propose to learn the kernel using adversarial learning techniques, and call the resulting learning scheme MMD GANs. The authors perform a number of empirical studies in order to compare the performance of MMD GANs to MMD nets and other standard GAN approaches. Their findings indicate, that MMD GANs achieve state of the art results on standard benchmark datasets.

Another recent paper by Bińkowski et al. (2018) contains a theoretical explanation of how GAN training can lead to biased gradient estimates for the generator network parameters in multiple discriminator learning settings. The bias arises due to the discriminator being optimized based on samples. This is true for MMD GANs as well as Wasserstein GANs (Arjovsky, Chintala, and Bottou, 2017). The authors prove that in the case of a fixed discriminator, the gradients on the generator parameters are unbiased.

Arjovsky, Chintala, and Bottou (2017) also describe a method for choosing the kernel for the MMD critic and suggest how one can adapt training strategies from the Wasserstein GANs. The paper contains empirical studies demonstrating that MMD GANs can achieve comparable performance as Wasserstein GANs with a much smaller critic network, making training more efficient and practical. In addition, the authors also propose a new metric for evaluating GAN convergence.

Conclusion

In this thesis we studied generalization bounds for existing learning algorithms using PAC-Bayes theory and nonconvex optimization. We also propose modifications to several existing learning algorithms in order to control overfitting. In addition, we design a new learning algorithm for unsupervised learning and analyze its generalization properties.

We produced the first nonvacuous train-set bound on the risk for a modern neural network via direct PAC-Bayes risk bound optimization. One drawback of our approach is that it is computationally intensive and is difficult to apply to much larger networks. Despite our bounds being the best known, the bounds we obtain are still much higher than error estimates on heldout data. In order to explain generalization, we need tighter bounds. Our current method relies on an isotropic Gaussian PAC-Bayesian posterior. As a result, the posterior does not capture the dependencies between neural network weights. Replacing the isotropic Gaussian posterior with a non-isotropic Gaussian posterior distribution on the parameters may produce a smaller KL divergence and tighter bound. However, networks used in practice have millions of parameters and so we cannot hope to optimize a full covariance matrix scalably. Variational inference methods that use factored representations could potentially be used to overcome this problem.

We introduced a PAC-Bayes bound that allows one to use a data-dependent but differentially private prior. Differential privacy is a very strong notion, and so we show that weak convergence towards a private prior suffices for generalization. It would be interesting to explore the necessity of this condition in order to obtain PAC-Bayes generalization bounds with data-dependent priors.

This leads to a related problem. PAC-Bayes generalization bounds allow us to bound the performance of a stochastic classifier. Therefore, we may understand the expected generalization properties of stochastic learning algorithms by studying their distributions. While experimentally evaluating the differentially private PAC-Bayes bound, we centered our attention on SGLD. Under certain assumptions, SGLD is known to converge weakly to a Gibbs distribution, which minimize PAC-Bayes bounds. However, recent works suggests that SGD may be usefully thought of as a stochastic learning algorithm. This is yet to be fully

understood. Thus a fundamental problem is: can we precisely formalize the distributions of SGD and SGLD? This knowledge may lead to improved understanding of the generalization properties of these learning algorithms. It may also lead to the construction of better approximate samplers to Gibbs distributions with trackable generalization error.

Poor prior choices lead to loose PAC-Bayes bounds. Local prior bounds have a big potential, since they eliminate the need to choose a prior distribution. We empirically evaluated local prior bounds using SGLD. We experimentally highlight that current local prior bounds need to account for the worst-case data distribution and thus fail to capture the dynamics of the generalization error for easy data as the temperature parameter in a Gibbs distribution increases beyond the point where the Gibbs distribution overfits on random labels. Further research is needed in order to obtain useful PAC-Bayes bounds based on local priors. One approach is to seek tighter data-dependent upper bounds on the KL divergence between the posterior and the local prior. Such an estimate should be sensitive to the data distribution. For the purposes of understanding generalization, we could potentially make use of held-out data.

References

V Vapnik and A Chervonenkis (1974). "About structural risk minimization principle". *Automation Remote Control* 8, p. 9.

Jorma Rissanen (June 1983). "A Universal Prior for Integers and Estimation by Minimum Description Length". *Ann. Statist.* 11.2, pp. 416–431. DOI: 10.1214/aos/1176346150.

Andrew Blake and Andrew Zisserman (1987). *Visual Reconstruction*. Cambridge, MA, USA: MIT Press.

Eric Saund (1989). "Dimensionality-Reduction Using Connectionist Networks." *IEEE Trans. Pattern Anal. Mach. Intell.* 11.3, pp. 304–314.

Geoffrey E. Hinton and Drew van Camp (1993). "Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights". *Proceedings of the Sixth Annual Conference on Computational Learning Theory*. COLT '93. Santa Cruz, California, USA: ACM, pp. 5–13. DOI: 10.1145/168304.168306.

David J.C. MacKay (1994). "Bayesian Neural Networks and Density Networks". *Nuclear Instruments and Methods in Physics Research, A*, pp. 73–80.

Paul W Goldberg and Mark R Jerrum (1995). "Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers". *Machine Learning* 18.2-3, pp. 131–148.

Pascal Koiran and Eduardo D Sontag (1996). "Neural networks with quadratic VC dimension". *Advances in neural information processing systems*, pp. 197–203.

Peter L Bartlett (1997). "For valid generalization the size of the weights is more important than the size of the network". *Advances in Neural Information Processing Systems*, pp. 134–140.

Sepp Hochreiter and Jürgen Schmidhuber (Jan. 1997). "Flat Minima". *Neural Comput.* 9.1, pp. 1–42. DOI: 10.1162/neco.1997.9.1.1.

John Shawe-Taylor and Robert C Williamson (1997). "A PAC analysis of a Bayesian estimator". *Proceedings of the tenth annual conference on Computational learning theory*. ACM, pp. 2–9.

Peter L Bartlett (1998). "The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network". *IEEE Transactions on Information Theory* 44.2, pp. 525–536.

Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998). "Gradient-based learning applied to document recognition". *Proceedings of the IEEE*, pp. 2278–2324.

Kenji Yamanishi (1998). "A decision-theoretic extension of stochastic complexity and its applications to learning". *IEEE Transactions on Information Theory* 44.4, pp. 1424–1439.

Peter L Bartlett, Vitaly Maiorov, and Ron Meir (1999). "Almost linear VC dimension bounds for piecewise polynomial networks". *Advances in Neural Information Processing Systems*, pp. 190–196.

David A. McAllester (1999a). "PAC-Bayesian Model Averaging". *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*. COLT '99. Santa Cruz, California, USA: ACM, pp. 164–170. DOI: 10.1145/307400.307435.

— (1999b). "Some PAC-Bayesian Theorems". *Machine Learning* 37.3, pp. 355–363. DOI: 10.1023/A:1007618624809.

Kenji Yamanishi (1999). "Extended stochastic complexity and minimax relative loss analysis". *International Conference on Algorithmic Learning Theory*. Springer, pp. 26–38.

John Langford and Matthias Seeger (2001). *Bounds for Averaging Classifiers*. Tech. rep. CMU-CS-01-102. Carnegie Mellon University.

Peter L Bartlett and Shahar Mendelson (2002). "Rademacher and Gaussian complexities: Risk bounds and structural results". *Journal of Machine Learning Research* 3.Nov, pp. 463–482.

Olivier Bousquet and André Elisseeff (2002). "Stability and generalization". *Journal of Machine Learning Research* 2.Mar, pp. 499–526.

Vladimir Koltchinskii and Dmitry Panchenko (2002). "Empirical margin distributions and bounding the generalization error of combined classifiers". *Ann. Statist.* 30.1, pp. 1–50.

John Langford (2002). "Quantitatively tight sample complexity bounds". PhD thesis. Carnegie Mellon University.

John Langford and Rich Caruana (2002a). "(Not) Bounding the True Error". *Advances in Neural Information Processing Systems 14*. Ed. by T. G. Dietterich, S. Becker, and Z. Ghahramani. MIT Press, pp. 809–816.

— (2002b). "(Not) Bounding the True Error". *Advances in Neural Information Processing Systems 14*. Ed. by T. G. Dietterich, S. Becker, and Z. Ghahramani. MIT Press, pp. 809–816.

Peter L. Bartlett and Shahar Mendelson (Mar. 2003). "Rademacher and Gaussian Complexities: Risk Bounds and Structural Results". *J. Mach. Learn. Res.* 3, pp. 463–482.

David A McAllester (2003). "PAC-Bayesian stochastic model selection". *Machine Learning* 51.1, pp. 5–21.

Shahar Mendelson (2003). "A Few Notes on Statistical Learning Theory". English. *Advanced Lectures on Machine Learning*. Ed. by Shahar Mendelson and AlexanderJ. Smola. Vol. 2600. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 1–40.

Andreas Maurer (2004). *A note on the PAC-Bayesian theorem*. arXiv: 041109 [cs.LG].

Peter Grünwald (2005). "A tutorial introduction to the minimum description length principle".

Carl Edward Rasmussen and Christopher K. I. Williams (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Cynthia Dwork (2006). "Differential Privacy". *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10–14, 2006, Proceedings, Part II*. Ed. by Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–12. DOI: 10.1007/11787006_1.

G E Hinton and R R Salakhutdinov (July 2006). "Reducing the dimensionality of data with neural networks". *Science* 313.5786, pp. 504–507.

Olav Kallenberg (2006). *Foundations of modern probability*. Springer Science & Business Media.

Tong Zhang (2006a). "From ε-entropy to KL-entropy: Analysis of minimum information complexity density estimation". *The Annals of Statistics* 34.5, pp. 2180–2210. DOI: 10.1214/009053606000000704.

— (2006b). "Information-theoretic upper and lower bounds for statistical estimation". *IEEE Transactions on Information Theory* 52.4, pp. 1307–1321.

Olivier Catoni (2007). "PAC-Bayesian supervised classification: the thermodynamics of statistical learning". *arXiv preprint arXiv:0712.0248*.

Peter D Grünwald (2007). *The minimum description length principle*. MIT press.

Frank McSherry and Kunal Talwar (2007). "Mechanism Design via Differential Privacy". *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '07. Washington, DC, USA: IEEE Computer Society, pp. 94–103. DOI: 10.1109/FOCS.2007.41.

Cynthia Dwork (2008a). "Differential privacy: A survey of results". *International Conference on Theory and Applications of Models of Computation*. Springer, pp. 1–19.

— (2008b). "Differential privacy: A survey of results". *International Conference on Theory and Applications of Models of Computation*. Springer, pp. 1–19.

Wenxin Jiang and Martin A Tanner (2008). "Gibbs posterior for variable selection in high-dimensional classification and data mining". *The Annals of Statistics*, pp. 2207–2231.

Bharath K. Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Gert Lanckriet, and Bernhard Schölkopf (2008). "Injective Hilbert Space Embeddings of Probability Measures". *Conf. Comp. Learn. Theory, (COLT)*.

Alex Krizhevsky (2009). *Learning Multiple Layers of Features from Tiny Images*. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Ruslan Salakhutdinov and Geoffrey E. Hinton (2009). "Deep Boltzmann Machines". *Journal of Machine Learning Research - Proceedings Track* 5, pp. 448–455.

Yann LeCun, Corinna Cortes, and Christopher J. C. Burges (2010). *MNIST handwritten digit database*. http://yann.lecun.com/exdb/mnist/.

J. M. Susskind, A. K. Anderson, and G. E. Hinton (2010). *The Toronto face database*. Tech. rep.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate (2011). "Differentially private empirical risk minimization". *Journal of Machine Learning Research* 12.Mar, pp. 1069–1109.

Max Welling and Yee W Teh (2011). "Bayesian learning via stochastic gradient Langevin dynamics". *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 681–688.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola (Mar. 2012). "A Kernel Two-sample Test". *J. Mach. Learn. Res.* 13, pp. 723–773.

Peter Grünwald (2012). "The Safe Bayesian-Learning the Learning Rate via the Mixability Gap." *ALT*. Springer, pp. 169–183.

Daniel Kifer, Adam Smith, and Abhradeep Thakurta (2012). "Private convex empirical risk minimization and high-dimensional regression". *Journal of Machine Learning Research* 1.41, pp. 1–40.

Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun (2012). "PAC-Bayes bounds with data dependent priors". *Journal of Machine Learning Research* 13.Dec, pp. 3507–3531.

T. Tieleman and G. Hinton (2012). *Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude*. COURSERA: Neural Networks for Machine Learning.

Guy Lever, François Laviolette, and John Shawe-Taylor (2013). "Tighter PAC-Bayes bounds through distribution-dependent priors". *Theoretical Computer Science* 473, pp. 4–28. DOI: 10.1016/j.tcs.2012.10.013.

David A. McAllester (2013). "A PAC-Bayesian Tutorial with A Dropout Bound". *CoRR* abs/1307.2118.

Darakhshan J Mir (2013). "Differential privacy: an exploration of the privacy-utility landscape". PhD thesis. Rutgers University.

Benigno Uria, Iain Murray, and Hugo Larochelle (2013). "A Deep and Tractable Density Estimator." *CoRR* abs/1310.1757.

Raef Bassily, Adam Smith, and Abhradeep Thakurta (2014). "Differentially private empirical risk minimization: Efficient algorithms and tight error bounds". *arXiv preprint arXiv:1405.7085*.

Christos Dimitrakakis, Blaine Nelson, Aikaterini Mitrokotsa, and Benjamin IP Rubinstein (2014). "Robust and private Bayesian inference". *International Conference on Algorithmic Learning Theory*. Springer, pp. 291–305.

Cynthia Dwork, Aaron Roth, et al. (2014a). "The algorithmic foundations of differential privacy". *Foundations and Trends® in Theoretical Computer Science* 9.3–4, pp. 211–407.

— (2014b). "The algorithmic foundations of differential privacy". *Foundations and Trends in Theoretical Computer Science* 9.3–4, pp. 211–407.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair†, Aaron Courville, and Yoshua Bengio (2014). "Generative Adversarial Nets".

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro (2014). *In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning*. Workshop track poster at ICLR 2015. arXiv: 1412.6614v4 [cs.LG].

Shai Shalev-Shwartz and Shai Ben-David (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research* 15.1, pp. 1929–1958.

Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org.

Carlo Baldassi, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina (2015). "Subdominant Dense Clusters Allow for Simple Learning and High Computational Performance in Neural Networks with Discrete Synapses". *Phys. Rev. Lett.* 115 (12), p. 128101. DOI: 10.1103/PhysRevLett.115.128101.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toni Pitassi, Omer Reingold, and Aaron Roth (2015a). "Generalization in adaptive data analysis and holdout reuse". *Advances in Neural Information Processing Systems*, pp. 2350–2358.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth (2015b). "Preserving statistical validity in adaptive data analysis". *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*. ACM, pp. 117–126.

Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani (2015). "Training Generative Neural Networks via Maximum Mean Discrepancy Optimization". *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*. UAI'15. Amsterdam, Netherlands: AUAI Press, pp. 258–267.

Moritz Hardt, Benjamin Recht, and Yoram Singer (2015). "Train faster, generalize better: Stability of stochastic gradient descent". *CoRR* abs/1509.01240.

Diederik P Kingma, Tim Salimans, and Max Welling (2015). "Variational Dropout and the Local Reparameterization Trick". *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., pp. 2575–2583.

Yujia Li, Kevin Swersky, and Richard Zemel (2015). "Generative Moment Matching Networks". http://arxiv.org/abs/1502.02761v1.

Behnam Neyshabur, Ruslan Salakhutdinov, and Nathan Srebro (2015). "Path-SGD: Path-Normalized Optimization in Deep Neural Networks". *Advances in Neural Information Processing Systems*. arXiv: 1506.02617v1 [cs.LG].

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro (2015). "Norm-based capacity control in neural networks". *Proceedings of the Eighth Annual Conference on Learning Theory*. COLT 2016, pp. 1376–1401. arXiv: 1503.00036v2 [cs.LG].

Yu-Xiang Wang, Stephen E. Fienberg, and Alexander J. Smola (2015). "Privacy for Free: Posterior Sampling and Stochastic Gradient Monte Carlo". *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, pp. 2493–2502.

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang (2016). "Deep Learning with Differential Privacy". *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. Vienna, Austria: ACM, pp. 308–318. DOI: 10.1145/2976749.2978318.

Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina (2016). "Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes". *Proceedings of the National Academy of Sciences* 113.48, E7655–E7662. DOI: 10.1073/pnas.1608103113. eprint: http://www.pnas.org/content/113/48/E7655.full.pdf.

Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman (2016). "Algorithmic stability for adaptive data analysis". *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, pp. 1046–1059.

Luc Bégin, Pascal Germain, François Laviolette, and Jean-Francis Roy (2016). "PAC-Bayesian bounds based on the Rényi divergence". *Artificial Intelligence and Statistics*, pp. 435–444.

Pier Giovanni Bissiri, CC Holmes, and Stephen G Walker (2016). "A general framework for updating belief distributions". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 78.5, pp. 1103–1130.

Pascal Germain, Francis Bach, Alexandre Lacoste, and Simon Lacoste-Julien (2016). "PAC-Bayesian Theory Meets Bayesian Inference". *Advances in Neural Information Processing Systems*, pp. 1884–1892.

Peter D Grünwald and Nishant A Mehta (2016). "Fast Rates for General Unbounded Loss Functions: from ERM to Generalized Bayes". arXiv: 1605.00252.

Peter D. Grünwald and Nishant A. Mehta (2016). "Fast Rates with Unbounded Losses". *CoRR* abs/1605.00252. arXiv: 1605.00252.

Elad Hazan, Kfir Yehuda Levy, and Shai Shalev-Shwartz (2016). "On Graduated Optimization for Stochastic Non-Convex Problems". *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 1833–1841.

Kentaro Minami, Hitomi Arai, Issei Sato, and Hiroshi Nakagawa (2016). "Differential Privacy without Sensitivity". *Advances in Neural Information Processing Systems 29*. Ed. by

D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., pp. 956–964.

Dougal J Sutherland, Hsiao-Yu Tung, Heiko Strathmann, Soumyajit De, Aaditya Ramdas, Alex Smola, and Arthur Gretton (2016). "Generative models and model criticism via optimized maximum mean discrepancy". *arXiv preprint arXiv:1611.04488*.

Yee Whye Teh, Alexandre H Thiery, and Sebastian J Vollmer (2016). "Consistency and fluctuations for stochastic gradient Langevin dynamics". *Journal of Machine Learning Research* 17, pp. 1–33.

Alessandro Achille and Stefano Soatto (2017). "On the Emergence of Invariance and Disentangling in Deep Representations". *CoRR* abs/1706.01350. arXiv: 1706.01350.

Martin Arjovsky, Soumith Chintala, and Léon Bottou (2017). "Wasserstein gan". *arXiv preprint arXiv:1701.07875*.

Peter Bartlett, Dylan J. Foster, and Matus Telgarsky (2017a). "Spectrally-normalized margin bounds for neural networks". *CoRR* abs/1706.08498. arXiv: 1706.08498.

Peter L. Bartlett (2017). "The impact of the nonlinearity on the VC-dimension of a deep network". Preprint.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky (2017b). "Spectrally-normalized margin bounds for neural networks". *Advances in Neural Information Processing Systems*, pp. 6241–6250.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe (2017). "Variational inference: A review for statisticians". *Journal of the American Statistical Association* just-accepted.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina (2017). "Entropy-SGD: Biasing Gradient Descent Into Wide Valleys". *International Conference on Learning Representations (ICLR)*. arXiv: 1611.01838v4 [cs.LG].

Gintare Karolina Dziugaite and Daniel M. Roy (2017). "Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data". *Proceedings of the 33rd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. arXiv: 1703.11008.

Peter D Grünwald and Nishant A Mehta (2017). "A Tight Excess Risk Bound via a Unified PAC-Bayesian-Rademacher-Shtarkov-MDL Complexity". *arXiv preprint arXiv:1710.07732*.

Suriya Gunasekar, Blake Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro (2017). *Implicit Regularization in Matrix Factorization*. arXiv: 1705.09280v1 [cs.LG].

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang (2017). "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima". *International Conference on Learning Representations (ICLR)*. arXiv: 1609.04836v2 [cs.LG].

Holden Lee, Rong Ge, Andrej Risteski, Tengyu Ma, and Sanjeev Arora (2017). "On the ability of neural nets to express distributions". *arXiv preprint arXiv:1702.07028*.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos (2017). "MMD GAN: Towards deeper understanding of moment matching network". *Advances in Neural Information Processing Systems*, pp. 2200–2210.

Behnam Neyshabur (2017). "Implicit regularization in deep learning". *arXiv preprint arXiv:1709.01953*.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro (2017a). *A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks*. arXiv: 1707.09564.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro (2017b). "Exploring generalization in deep learning". *Advances in Neural Information Processing Systems*, pp. 5949–5958.

Luca Oneto, Sandro Ridella, and Davide Anguita (2017). "Differential privacy and generalization: Sharper bounds with applications". *Pattern Recognition Letters* 89, pp. 31–38. DOI: 10.1016/j.patrec.2017.02.006.

Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky (2017). "Non-convex learning via Stochastic Gradient Langevin Dynamics: a nonasymptotic analysis". *Proc. Conference on Learning Theory (COLT)*. arXiv: 1702.03849.

Ravid Shwartz-Ziv and Naftali Tishby (2017). "Opening the black box of deep neural networks via information". *arXiv preprint arXiv:1703.00810*.

Niklas Thiemann, Christian Igel, Olivier Wintenberger, and Yevgeny Seldin (2017). "A Strongly Quasiconvex PAC-Bayesian Bound". *International Conference on Algorithmic Learning Theory*, pp. 466–492.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals (2017). "Understanding deep learning requires rethinking generalization". *International Conference on Representation Learning (ICLR)*. arXiv: 1611.03530v2 [cs.LG].

Alessandro Achille and Stefano Soatto (2018). "Information dropout: Learning optimal representations through noisy computation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. First appeared as https://arxiv.org/abs/1611.01353.

Pierre Alquier and Benjamin Guedj (May 2018). "Simpler PAC-Bayesian Bounds for Hostile Data". *Mach. Learn.* 107.5, pp. 887–902. DOI: 10.1007/s10994-017-5690-0.

Mikołaj Bińkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton (2018). "Demystifying MMD GANs". *arXiv preprint arXiv:1801.01401*.

Gintare Karolina Dziugaite and Daniel M. Roy (2018a). "Data-dependent PAC-Bayes priors via differential privacy". *Advances in Neural Information Processing Systems (NIPS)*. Vol. 29. Cambridge, MA: MIT Press. arXiv: 1802.09583.

— (2018b). "Entropy-SGD optimizes the prior of a PAC-Bayes bound: Generalization properties of Entropy-SGD and data-dependent priors". *Proceedings of the 35th International Conference on Machine Learning (ICML)*. arXiv: 1712.09376.

Samuel L Smith and Quoc V Le (2018). "A Bayesian perspective on generalization and stochastic gradient descent". *Proc. of ICLR*.