Toward Better Generalization Bounds with Locally Elastic Stability

Zhun Deng [1] **Hangfeng He** [2] **Weijie J. Su** [3]

Abstract

Algorithmic stability is a key characteristic to ensure the generalization ability of a learning algorithm. Among different notions of stability, *uniform stability* is arguably the most popular one, which yields exponential generalization bounds. However, uniform stability only considers the worst-case loss change (or so-called sensitivity) by removing a single data point, which is distribution-independent and therefore undesirable. There are many cases that the worst-case sensitivity of the loss is much larger than the average sensitivity taken over the single data point that is removed, especially in some advanced models such as random feature models or neural networks. Many previous works try to mitigate the distribution independent issue by proposing weaker notions of stability, however, they either only yield polynomial bounds or the bounds derived do not vanish as sample size goes to infinity. Given that, we propose *locally elastic stability* as a weaker and distribution-dependent stability notion, which still yields exponential generalization bounds. We further demonstrate that locally elastic stability implies tighter generalization bounds than those derived based on uniform stability in many situations by revisiting the examples of bounded support vector machines, regularized least square regressions, and stochastic gradient descent.

1. Introduction

A central question in machine learning is how the performance of an algorithm on the training set carries over to unseen data. Continued efforts to address this question have given rise to numerous generalization error bounds on the gap between the population risk and empirical risk, using a variety of approaches from statistical learning theory (Vapnik, 1979; 2013; Bartlett & Mendelson, 2002; Bousquet & Elisseeff, 2002). Among these developments, algorithmic stability stands out as a general approach that allows one to relate certain specific properties of an algorithm to its generalization ability. Ever since the work of Devroye & Wagner (1979), where distribution-independent exponential generalization bounds for the concentration of the leave-one-out estimate are proposed, various results for different estimates are studied. Lugosi & Pawlak (1994) study the smooth estimates of the error for the deleted estimate developed in terms of a posterior distribution and Kearns & Ron (1999) propose *error stability*, which provides sanity-check bounds for more general classes of learning rules regarding the deleted estimate. For general learning rules, Bousquet & Elisseeff (2002) propose the notion of *uniform stability*, which extends Lugosi & Pawlak (1994)'s work and yields exponential generalization bounds. Loosely speaking, Bousquet & Elisseeff (2002) show that an algorithm would generalize well to new data if this algorithm is uniformly stable in the sense that its loss function is not sensitive to the deletion of a single data point. To date, uniform stability is perhaps the most popular stability notion.

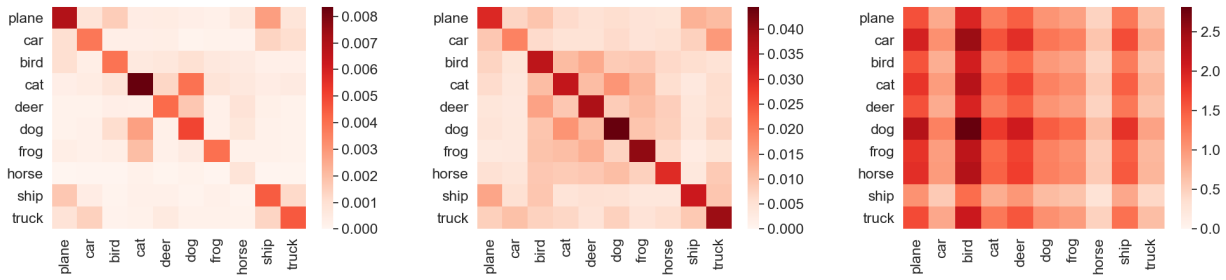
Despite many recent developments, most results on stability and generalization can be divided into two categories if not counting sanity-check bounds. The first category includes stability notions such as hypothesis stability, which only yield sub-optimal *polynomial* generalization bounds. The second category includes stability notions based on uniform stability and its variants, which yield optimal *exponential* generalization bounds. Nevertheless, the stability notions in the second category either stop short of providing distribution-dependent bounds or, worse, the bounds do *not* vanish even when the training sample size tends to infinity (Abou-Moustafa & Szepesvári, 2019). Recognizing these facts, in this paper, we aim to relax the uniform stability notion and propose a weaker and distribution-dependent stability notion, which yields exponential generalization bounds that are consistent in the sense that the bounds vanish to zero as the training sample size tends to infinity.

1.1. A Motivating Example

To further motivate our study, note that there are many cases where the worst-case sensitivity of the loss is much

[1]Harvard University [2]Department of Computer and Information Science, University of Pennsylvania [3]Wharton Statistics Department, University of Pennsylvania. Correspondence to: Zhun Deng <zhundeng@g.harvard.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.



(a) Sensitivity of neural networks. (b) Sensitivity of a random feature model. (c) Sensitivity of a linear model.

Figure 1. Class-level sensitivity approximated by influence functions for neural networks (based on a pre-trained 18-layer ResNet), a random feature model (based on a randomly initialized 18-layer ResNet), and a linear model on CIFAR-10. The vertical axis denotes the classes in the test data and the horizontal axis denotes the classes in the training data. The class-level sensitivity from class a in the training data to class b in the test data is defined as $C(c_a, c_b) = \frac{1}{|S_a| \times |\tilde{S}_b|} \sum_{z_i \in S_a} \sum_{z \in \tilde{S}_b} |l(\hat{\theta}, z) - l(\hat{\theta}^{-i}, z)|$, where S_a denotes the set of examples from class a in the training data and $\tilde{S}_b$ denotes set of examples from class b in the test data.

larger than the average sensitivity, especially in random feature models or neural networks. As a concrete example, from Figure 1, we can observe that the sensitivity of neural networks and random feature models depends highly on the label information. To be precise, consider training two models on the CIFAR-10 dataset (Krizhevsky, 2009) and another dataset obtained by removing one training example, say an image of a plane, from CIFAR-10, respectively. Figure 1 shows that the difference between the loss function values for the two models *depends* on the label of the test image that the loss function is evaluated at: the difference between the loss function values, or sensitivity for short, is significant if the test image is another plane, and the sensitivity is small if the test image is from a different class, such as car or cat. Concretely, the average plane-to-plane difference is about seven times the average plane-to-cat difference. The dependence on whether the two images belong to the same class results in a pronounced diagonal structure in Figure 1(a), which is consistent with the phenomenon of *local elasticity* in deep learning training (He & Su, 2020; Chen et al., 2020). In particular, this structural property of the loss function differences clearly demonstrates that uniform stability fails to capture how sensitive the loss function is in the *population* sense, which is considerably smaller than the worst-case sensitivity, for the neural networks and random feature models.

1.2. Our Contribution

As our first contribution, we introduce a new notion of algorithmic stability that is referred to as *locally elastic stability* to take into account the message conveyed by Figure 1. This new stability notion imposes a data-dependent bound on the sensitivity of the loss function, as opposed to a constant bound that uniform stability and many of its relaxations use.

The second contribution of this paper is to develop a generalization bound for any locally elastically stable algorithm. This new generalization bound is obtained by a fine-grained analysis of the empirical risk, where using McDiarmid's inequality as in Bousquet & Elisseeff (2002) no longer works. Specifically, we expect the empirical sum of the sensitivities by deleting different samples to be close to the expected sensitivity taken over the deleted sample. However, conditioning on that event, the dependency among input examples invalidate McDiarmid's inequality. To overcome this difficulty, we develop novel techniques that allow us to obtain a sharper analysis of some important quantities. Our results show that the generalization error is, loosely speaking, upper bounded by the expectation of the sensitivity function associated with locally elastic stability over the population of training examples. Assuming uniform stability, however, classical generalization bounds are mainly determined by the largest possible sensitivity over all pairs of training examples. We further demonstrate that our bounds are tighter than those derived based on uniform stability in many situations by revisiting the examples of bounded support vector machines (SVM), regularized least square regressions, and stochastic gradient descent (SGD). Although it requires further exploration on how to make the new bounds applicable to deep learning models in practice, the insights from this new stability notion shall shed light on the development of future approaches toward demystifying the generalization ability of modern neural networks.

1.3. Related Work

Ever since Kearns & Ron (1999) and Bousquet & Elisseeff (2002) proposed the notions of uniform stability and hypothesis stability, a copious line of works has been de-

voted to extending and elaborating on their frameworks. In Mukherjee et al. (2006), Shalev-Shwartz et al. (2010) and Kutin & Niyogi (2002), the authors show there exist cases where stability is the key necessary and sufficient condition for learnability but uniform convergence is not. On one hand, error stability is not strong enough to guarantee generalization (Kutin & Niyogi, 2012). On the other hand, hypothesis stability guarantees generalization but only provides polynomial tail bounds. Fortunately, uniform stability guarantees generalization and further provides exponential tail bounds. In Feldman & Vondrak (2018), the authors develop the generalization bound for the cases where uniform stability parameter is of order $\Omega(1/\sqrt{m})$, where m is the sample size. In subsequent work, Feldman & Vondrak (2019) prove a nearly tight high probability bound for any uniformly stable algorithm. In Bousquet et al. (2020), the authors provide sharper bounds than Feldman & Vondrak (2019) and also provide general lower bounds which can be applied to certain generalized concentration inequalities. There are also works seeking to relax uniform stability such as (Abou-Moustafa & Szepesvári, 2019), but their bound still has a small term that would not vanish even with an infinite sample size and a vanishing stability parameter.

In addition, researchers demonstrate that many popular optimization methods, such as SGD, satisfy algorithmic stability. In Hardt et al. (2015), the authors show that SGD satisfies uniform stability. Lei & Ying (2020) further relax the smoothness and convexity assumptions, and others instead discuss the nonconvex case for SGD in more detail (Kuzborskij & Lampert, 2018; Madden et al., 2020). Kuzborskij & Lampert (2018) recently propose another notion of data-dependent stability for SGD. Our work can be viewed as a relaxation of uniform stability and SGD will be shown to satisfy our new notion of algorithmic stability.

2. Locally Elastic Stability

We first collect some notations that are used throughout this paper, which mostly follows that of Bousquet & Elisseeff (2002). Denote by $S = \{z_1, z_2, \cdots, z_m\}$ the training set, where $z_i \in \mathcal{Z} \subseteq \mathbb{R}^d$ are i.i.d. draws from a distribution $\mathcal{D}$ on the space $\mathcal{Z}$. One instance of $\mathcal{Z}$ is $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are input space and label space respectively. For a function class $\mathcal{F}$, a learning algorithm $\mathscr{A} : \mathcal{Z}^m \to \mathcal{F}$ takes the training set S as input and outputs a function $\mathcal{A}_S \in \mathcal{F}$. For any m-sized training set S, let $S^{-i} = \{z_1, \cdots, z_{i-1}, z_{i+1}, \cdots, z_m\}$ be derived by removing the ith element from S and $S^i = \{z_1, \cdots, z_{i-1}, z_i', z_{i+1}, \cdots, z_m\}$ be derived by replacing the ith element from S with another example z_i'. For any input z, we consider a loss function $l(f, z)$. We are particularly interested in the loss $l(f, z)$ when the function $f = \mathcal{A}_S$.

Now, we formally introduce the notion of locally elastic stability below. Let $\beta_m(\cdot, \cdot)$ be a sequence of functions indexed by $m \geq 2$ that each maps any pair of $z, z' \in \mathcal{Z}$ to a positive value.

Definition 2.1 (Locally Elastic Stability)**.** *An algorithm $\mathscr{A}$ has locally elastic stability $\beta_m(\cdot, \cdot)$ with respect to the loss function l if, for all m, the inequality*

$$|l(\mathcal{A}_S, z) - l(\mathcal{A}_{S^{-i}}, z)| \leq \beta_m(z_i, z)$$

holds for all $S \in \mathcal{Z}^m, 1 \leq i \leq m$, and $z \in \mathcal{Z}$.

In words, the change in the loss function due to the removal of any z_i is bounded by a function depending on both z_i and the data point z where the loss is evaluated. In this respect, locally elastic stability is *data-dependent*. In general, $\beta_m(\cdot, \cdot)$ is not necessarily symmetric with respect to its two arguments. To further appreciate this definition, we compare it with uniform stability, which is perhaps one of the most popular algorithmic stability notions.

Definition 2.2 (Uniform Stability (Bousquet & Elisseeff, 2002))**.** *Let β_m^U be a sequence of scalars. An algorithm $\mathscr{A}$ has uniform stability β_m^U with respect to the loss function l if*

$$|l(\mathcal{A}_S, z) - l(\mathcal{A}_{S^{-i}}, z)| \leq \beta_m^U \tag{1}$$

holds for all $S \in \mathcal{Z}^m, 1 \leq i \leq m$, and $z \in \mathcal{Z}$.

First of all, by definition one can set $\beta_m^U = \sup_{z', z} \beta_m(z', z)$. Furthermore, a simple comparison between the two notions immediately reveals that locally elastic stability offers a finer-grained definition of the loss function sensitivity. The gain is significant particularly in the case where the worst possible value of $|l(\mathcal{A}_S, z) - l(\mathcal{A}_{S^{-i}}, z)|$ is much larger than its typical realizations.

2.1. Estimation Using Influence Functions

In the introduction part, we motivated the proposal of locally elastic stability by showing the class-level sensitivity for a random feature model and neural networks in Figure 1. In this subsection, we elaborate more on the experimental results and the corresponding approximation method. The examples we considered demonstrate small $\beta_m(z_i, z)$ for most z's in $\mathcal{Z}$ for any training example z_i,. The fact that $\beta_m(z_i, z)$ is small for most of z's is important to obtain a sharper generalization bound with locally elastic stability than the bound with uniform stability.

Specifically, consider a function f that is parameterized by θ and write $l(\theta, z)$ instead of $l(f, z)$ for the loss. Writing $f = f_\theta$, the algorithm $\mathscr{A}$ aims to output $f_{\hat{\theta}}$ where $\hat{\theta} = \arg\min_{\theta \in \Theta} \sum_{j=1}^m l(\theta, z_j)/m$ (we temporarily ignore the issue of uniqueness of minimizers here). Then, $\mathcal{A}_S$ defined previously is exactly $f_{\hat{\theta}}$. Denote $\hat{\theta}^{-i} =$

Models	$\sup_{z'\in S, z\in\mathcal{Z}} \beta_m(z', z)$	$\sup_{z'\in\mathcal{Z}} \mathbb{E}_z \beta_m(z', z)$	$ratio$
Neural networks	3.05	0.02	153
Random feature model	1.73	0.04	43

Table 1. Comparison between locally elastic stability and uniform stability for neural networks and the random feature model in Figure 1.

$\arg\min_{\theta\in\Theta} \sum_{j\neq i} l(\theta, z_j)/m$, we aim to quantitatively estimate $|l(\hat{\theta}, z) - l(\hat{\theta}^{-i}, z)|$ for all i's. However, quantifying the above quantity for all i's is computationally prohibitive in practice for neural networks and also a pain even for random feature model. In order to alleviate the computational issue, we adopt influence functions from Koh & Liang (2017) and consider the same simplified model as in Koh & Liang (2017): an N-layer neural network whose first $N-1$ layers are pre-trained. Given that model, when the loss function $l(\theta, z)$ is strictly convex in θ for all z, such as the continuously used cross entropy loss and squared loss with l_2 penalty on θ, we have the following approximation:

$$\begin{aligned}\beta_m(z_i, z) &:= |l(\hat{\theta}, z) - l(\hat{\theta}^{-i}, z)| \\ &\approx \frac{1}{m}\left|\nabla_\theta l(\hat{\theta}, z) H_{\hat{\theta}}^{-1} \nabla_\theta l(\hat{\theta}, z_i)\right|,\end{aligned} \tag{2}$$

where $H_{\hat{\theta}} = \sum_{j=1}^{m} \nabla^2 l(\hat{\theta}, z_j)/m$ is the Hessian. We remark that it is very common in transfer learning to pre-train the $N-1$ layers and it is different from the random feature model, where the first $N-1$ layers are chosen to be independent of data. We further consider training the full N-layer neural networks by analyzing the sensitivity of the loss step-wisely for SGD in the Appendix. In Figure 1, we demonstrate the class-level sensitivity approximated by influence functions for neural networks (based on a pre-trained 18-layer ResNet (He et al., 2016)) and a random feature model (based on a randomly initialized 18-layer Resnet) on CIFAR-10 (Krizhevsky, 2009).

The results indicate that for random feature models and neural networks with a training example z_i, if z is from the same class of z_i, then $\beta_m(z_i, z)$ is large; if z is from a different class $\beta_m(z_i, z)$ is small. Recognizing the long tail property of class frequencies for image datasets in practice, it would lead to small $\beta_m(z_i, z)$'s for most z's for any training example z_i.

We close this section by providing empirical evidence to justify our statement that "$\beta_m(z_i, z)$ for most z's is small for any training example z_i." Specifically, we compare $\sup_{z'\in S, z\in\mathcal{Z}} \beta_m(z', z)$ and $\sup_{z'\in\mathcal{Z}} \mathbb{E}_z \beta_m(z', z)$ for both neural networks and the random feature model, and the results are shown in Table 1.

It is worth noticing the dependence on whether the two images belong to the same class results in a pronounced diagonal structure in Figure 1(a) and 1(b), and in contrast, linear models do not exhibit such a strong dependence on the class of images, as evidenced by the absence of a diagonal structure in Figure 1(c). We believe the above phenomenon is one of the reasons that neural networks generalize well and our new proposed stability provides a new direction towards understanding the generalization behavior of neural networks.

2.2. Connection with Local Elasticity

Locally elastic stability has a profound connection with a phenomenon identified by He & Su (2020), where the authors consider the question: how does the update of weights of neural networks using induced gradient at an image (say a tiger) impact the prediction at another image? In response to this question, He & Su (2020) observe that the impact is significant if the two images have the same membership (e.g., the test image is another tiger) or share features (e.g., a cat), and the impact is small if the two images are not semantically related (e.g., a plane).[1] In contrast, this phenomenon is generally not present in kernel methods, and Chen et al. (2020) argue that this absence is in part responsible for the ineffectiveness of neural tangent kernels compared to real-world neural networks in terms of generalization. Related observations have been made in Chatterjee (2020) and Fort et al. (2019). This phenomenon, which He & Su (2020) refer to as local elasticity, would imply the characteristic of neural networks that we observe in Figure 1. Intuitively, from local elasticity we would expect that if we remove an image of a cat in the training set S, the loss after training on a test image of a plane would not be affected much compared with the loss obtained by training on the original training set (assuming the same randomness from sampling). Conversely, the final loss would be affected much if the test image is another tiger. Our Definition 2.1 formalizes the intuition of local elasticity by incorporating the membership dependence into the sensitivity of the loss function, hence is named as locally elastic stability.

The improvement brought by locally elastic stability is further enhanced by the *diversity* of real-life data. First, the number of classes in tasks resembling practical applications is often very large. For example, the ImageNet dataset contains more than 1000 classes (Deng et al., 2009). For most pairs z', z, their class memberships are different, leading to a relatively small value of $\beta_m(z', z)$ compared to the uniform upper bound adopted in uniform stability. Moreover, the long tail property of real-life images suggest that the class of cats, for example, consists of many cats with different appearances and non-vanishing frequencies (Zhu

[1] To be complete, this phenomenon does not appear in the initialized neural networks and become pronounced only after several epochs of training on the dataset.

et al., 2014) (further elaborations are included in Appendix **??**). Combining with the observations mentioned above, we would expect that for any fixed training example z', $\beta_m(z', z)$ would be small for most z sampled from the distribution $\mathcal{D}$. Therefore, the use of a uniform upper bound on the sensitivity is too pessimistic.

3. Generalization Bounds

In this section, we present our generalization bound for locally elastically stable algorithms and compare it to those implied by classical algorithmic stability notions.

Assumptions. We assume the space $\mathcal{Z}$ is bounded. In addition, from the approximation shown in (2), for many problems one has $|l(\hat{\theta}, z) - l(\hat{\theta}^{-i}, z)| = O(1/m)$. Moreover, Bousquet & Elisseeff (2002) show that β_m^U in uniform stability satisfies $\beta_m^U = O(1/m)$ for many problems including bounded SVM, k-local rules, and general regularization algorithms. This fact suggests that it is reasonable to expect that $\beta_m(z', z) = O_{z',z}(1)/m$ for locally elastic stability. More specifically, we have the following assumption.

Assumption 3.1. *For the function $\beta_m(\cdot, \cdot)$, for any $z, z' \in \mathcal{Z}$,*

$$\beta_m(z', z) = \frac{\beta(z', z)}{m}$$

for some function $\beta(\cdot, \cdot)$ that is independent of m. In addition, $\beta(\cdot, z)$ as a function of its first argument is L-Lipchitz continuous for all $z \in \mathcal{Z}$ and the loss function and there exists $M_\beta > 0$ such that $|\beta(\cdot, \cdot)| \leq M_\beta$.

In essence, $\beta_m(z', z) = \beta(z', z)/m$ is equivalent to assuming that $\sup_m m\beta_m(z', z)$ is finite for all z', z. The boundedness assumption of $\beta(\cdot, \cdot)$ holds if β is a continuous function in conjunction with the boundedness of $\mathcal{Z}$. In relating this assumption to uniform stability in Definition 2.2, we can take $\beta_m^U = M_\beta/m$.

Now, we are ready to state our main theorem. For convenience, write $\Delta(\mathcal{A}_S)$ as a shorthand for the defect $\mathbb{E}_z l(\mathcal{A}_S, z) - \sum_{j=1}^m l(\mathcal{A}_S, z_j)/m$, where the expectation $\mathbb{E}_z$ is over the randomness embodied in $z \sim \mathcal{D}$. In particular, $\mathbb{E}_z l(\mathcal{A}_S, z)$ depends on $\mathcal{A}_S$.

Theorem 3.1. *Let $\mathscr{A}$ be an algorithm that has locally elastic stability $\beta_m(\cdot, \cdot)$ with respect to the loss function l, which satisfies $0 \leq l \leq M_l$ for a constant M_l. Under Assumption 3.1, for any given η and any $0 < \delta < 1$ and , for sufficiently large m, with probability at least $1 - \delta$, we have*

$$\Delta(\mathcal{A}_S) \leq \frac{2\sup_{z' \in \mathcal{Z}} \mathbb{E}_z \beta(z', z)}{m} + 2\left(2 \sup_{z' \in \mathcal{Z}} \mathbb{E}_z \beta(z', z) + \eta + M_l\right) \sqrt{\frac{2\log(2/\delta)}{m}}.$$

We remark that (1). the parameter η in Theorem 3.1 is used to control the deviation $\sup_{z' \in \mathcal{Z}} \left| \sum_{j \neq k} \beta(z', z_j)/m - \mathbb{E}_z \beta(z', z) \right|$. As shown in our Lemma A.4 in the Appendix, we only need $\eta > 2M_\beta/m$. Thus, as stated in our theorem, *for any given $\eta > 0$*, as long as the sample size is large enough, i.e. $m > 2M_\beta/\eta$, all the claims involving η hold. In the subsequent discussions, for instance, in Section 4.1, we can set $\eta = \sup_{z' \in \mathcal{Z}} \mathbb{E}_z \beta(z', z)$ and Theorem 3.1 still holds. (2). Theorem 3.1 holds for m that is larger than a bound depending on $\delta, \eta, d, L, M_\beta$, and M_l, One coarse and sufficient condition provided in the Appendix is that m is large enough such that $\log(C'm)/m \leqslant \eta^2/(64M_\beta^2)$, $2M^2\log(2/\delta)/(\tilde{M}^2 m) \leqslant \eta^2/(128M_\beta^2)$, $2M/\tilde{M}\sqrt{2\log(2/\delta)/m} \leqslant \eta^2/(128M_\beta^2)$, and $m > 2M_\beta/\eta$ for constants C' (depnding on d), $\tilde{M}, M_\beta, \eta$, which can be achieved once we notice that $\lim_{m\to\infty} \log(C'm)/m \to 0$.

In this theorem, the bound on the defect $\Delta(\mathcal{A}_S)$ tends to 0 as $m \to \infty$. The factor $\sqrt{\log(2/\delta)}$ results from the fact that this locally elastic stability-based bound is an exponential bound. Notably, the bound depends on locally elastic stability through $\sup_{z' \in \mathcal{Z}} \mathbb{E}_z \beta(z', z)$, which is closely related to error stability (Kearns & Ron, 1999). See more discussion in Section 4.

Remark 3.1. *In passing, we make a brief remark on the novelty of the proof. An important step in our proof is to take advantage of the fact that $|\sum_{j \neq k} \beta(z', z_j)/m - \mathbb{E}_z \beta(z', z)|$ is small with high probability. Conditioning on this event, however, z_j's are no longer an i.i.d. sample from $\mathcal{D}$. The dependence among input examples would unfortunately invalidate McDiarmid's inequality, which is a key technique in proving generalization bounds for uniform stability. To overcome this difficulty, we develop new techniques to obtain a more careful analysis of some estimates. More details can be found in our Appendix.*

4. Comparisons with Other Notions of Algorithmic Stability

Having established the generalization bound for locally elastic stability, we compare our results with some classical notions (Bousquet & Elisseeff, 2002). As will be shown in this subsection, error stability is not sufficient to guarantee generalization, hypothesis stability only yields polynomial bounds, and uniform stability only considers the largest loss change on z in $\mathcal{Z}$ by removing z_i from S. In contrast, locally elastic stability not only provides exponential bounds as uniform stability but also takes into account the varying sensitivity of the loss. This fine-grained perspective can be used to improve the generalization bounds derived from uniform stability, when the average loss change by removing z_i from S over different z's in $\mathcal{Z}$ is much

smaller than the worst-case loss change.

4.1. Uniform Stability

Following Bousquet & Elisseeff (2002), for an algorithm $\mathscr{A}$ having uniform stability β_m^U (see Definition 2.2) with respect to the loss function l, if $0 \leq l(\cdot,\cdot) \leq M_l$, for any $\delta \in (0,1)$ and sample size m, with probability at least $1-\delta$,

$$\Delta(\mathcal{A}_S) \leq 2\beta_m^U + (4m\beta_m^U + M_l)\sqrt{\frac{\log(1/\delta)}{2m}}.$$

Notice that if an algorithm $\mathscr{A}$ satisfies locally elastic stability with $\beta_m(\cdot,\cdot)$, then it it has uniform stability with parameter $\beta_m^U := \sup_{z'\in S, z\in\mathcal{Z}} \beta(z',z)/m$. We can identify $\sup_{z'\in S, z\in\mathcal{Z}} \beta(z',z)$ with M_β in Assumption 3.1.

To get a better handle on the tightness of our new generalization bound, we revisit some classic examples in Bousquet & Elisseeff (2002) and demonstrate the superiority of using our bounds over using uniform stability bounds in certain cases. In order to have a clear presentation, let us briefly recap the assumptions and concepts used in Bousquet & Elisseeff (2002).

Assumption 4.1. *Any loss function l considered in this paragraph is associated with a cost function c_l, such that for a hypothesis f with respect to an example $z = (x, y)$, the loss function is defined as*

$$l(f, z) = c_l(f(x), y).$$

Definition 4.1. *A loss function l defined on $\mathcal{Y}^{\mathcal{X}} \times \mathcal{Y}$ is σ-admissible with respect to $\mathcal{Y}^{\mathcal{X}}$ if the associated cost function c_l is convex with respect to its first argument and the following condition holds: for any $y_1, y_2 \in \mathcal{Y}$ and any $y' \in \mathcal{Y}$*

$$|c_l(y_1, y') - c_l(y_2, y')| \leq \sigma\|y_1 - y_2\|_{\mathcal{Y}},$$

where $\|\cdot\|_{\mathcal{Y}}$ is the corresponding norm on $\mathcal{Y}$.

Reproducing kernel Hilbert space. A reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ is a Hilbert space of functions, in which point evaluation is a continuous linear functional and satisfies for any $h \in \mathcal{H}$, any $x \in \mathcal{X}$

$$h(x) = \langle h, K(x,\cdot)\rangle$$

where K is the corresponding kernel of $\mathcal{H}$. In particular, by Cauchy-Schwarz inequality, for any $h \in \mathcal{H}$, any $x \in \mathcal{X}$

$$|h(x)| \leq \|h\|_K \sqrt{K(x,x)},$$

where $\|\cdot\|_K$ is the norm induced by kernel K for the reproducing kernel Hilbert space $\mathcal{H}$. We denote $\sqrt{K(x,x)}$ as $\kappa(x)$. Notice that for the reproducing kernel Hilbert space, K must be a positive semi-definite kernel and $\kappa(x) \geq 0$.

In order to derive locally elastic stability bounds, we introduce the following lemma, which is a variant of Theorem 22 in Bousquet & Elisseeff (2002).

Lemma 4.1. *Let $\mathcal{H}$ be a reproducing kernel Hilbert space with kernel K, and for any $x \in \mathcal{X}$, $K(x,x) \leq \kappa^2 < \infty$. The loss function l is σ-admissible with repect to $\mathcal{H}$ and the learning algorithm is defined by*

$$\mathcal{A}_S = \arg\min_{h\in\mathcal{H}} \frac{1}{m}\sum_{j=1}^m l(h, z_j) + \lambda\|h\|_K^2,$$

where λ is a positive constant. Then, $\mathcal{A}_S$ has uniform stability β_m^U and locally elastic stability $\beta_m(z_i, z)$ such that

$$\beta_m^U \leq \frac{\sigma^2\kappa^2}{2\lambda m} \text{ and } \beta_m(z_i,z) \leq \frac{\sigma^2\kappa(x_i)\kappa(x)}{2\lambda m}.$$

Now, we are ready to investigate how the locally elastic bounds improve over the uniform stability bounds in the bounded SVM regression and regularized least square regression studied in Bousquet & Elisseeff (2002). We remark here, following the same settings in Bousquet & Elisseeff (2002), though the algorithms in the examples below are minimizing a regularized version of the loss, the generalization gap studied above is still $\Delta(\mathcal{A}_S) = \mathbb{E}_z l(\mathcal{A}_S, z) - \sum_{j=1}^m l(\mathcal{A}_S, z_j)/m$.

Example 4.1 (Stability of bounded SVM regression)**.** *Assume K is a bounded kernel, such that $K(x,x) \leq \kappa^2$ for all $x \in \mathcal{X}$, and $\mathcal{Y} = [0, B]$ for a real positive number B. Consider the loss function for $\tau > 0$,*

$$l(f,z) = |f(x) - y|_\tau = \begin{cases} 0, & \text{if } |f(x)-y| \leq \tau, \\ |f(x)-y| - \tau, & \text{otherwise.} \end{cases}$$

The learning algorithm is defined by

$$\mathcal{A}_S = \arg\min_{h\in\mathcal{H}} \frac{1}{m}\sum_{j=1}^m l(h, z_j) + \lambda\|h\|_K^2.$$

Noting $0 \leq l(f,z) \leq B$ and $\sigma = 1$ in our case [2] and using Lemma 4.1, we obtain the following bound via uniform stability:

$$\Delta(\mathcal{A}_S) \leq \frac{\kappa^2}{\lambda m} + \left(\frac{2\kappa^2}{\lambda} + B\right)\sqrt{\frac{\log(1/\delta)}{2m}}. \quad (\mathcal{B}_1)$$

In addition, we obtain the following bound via locally elastic stability by choosing $\eta = \kappa\mathbb{E}_x\kappa(x)/\lambda$

$$\Delta(\mathcal{A}_S) \leq \frac{\kappa\mathbb{E}_x\kappa(x)}{\lambda m} + \left(\frac{3\kappa\mathbb{E}_x\kappa(x)}{\lambda} + 2B\right)\sqrt{\frac{2\log(2/\delta)}{m}}. \quad (\mathcal{B}_2)$$

[2] There are several small typos in the original Example 1 and 3 in Bousquet & Elisseeff (2002) with respect to the range of $l(f,z)$ which we correct in our examples.

For simplicity, we consider K to be the bilinear kernel (similar analysis can be extended to other kernels such as polynomial kernels) $K(x,x') = \langle x, x'\rangle$ and all $x \in \mathcal{X}$'s norm are bounded by B'. Then, $\kappa = B'^2$ and $\mathbb{E}_x\kappa(x) = \mathbb{E}_x\|x\|^2$. Apparently, the first term on the RHS in $(\mathcal{B}_2)$ is smaller than the first term on the RHS in $(\mathcal{B}_1)$. So we focus on comparing the second terms for both inequalities. For $\delta < 0.5$, we have $\log(2/\delta) \leq 2\log(1/\delta)$. Applying the above inequality to $(\mathcal{B}_2)$, we can simplify the expressions, and if we further have

$$\left(\frac{2\kappa^2}{\lambda} + B\right) \geq 2\sqrt{2}\left(\frac{3\kappa\mathbb{E}_x\kappa(x)}{\lambda} + 2B\right) \tag{3}$$

the bound obtained in $(\mathcal{B}_2)$ is tighter than the one in $(\mathcal{B}_1)$. If the scale of $\kappa\mathbb{E}_x\kappa(x)/\lambda$ and B and are relatively small comparing with κ^2/λ, (3) apparently holds. Notice the first requirement regarding $\kappa\mathbb{E}_x\kappa(x)/\lambda$ being relatively small comparing with κ^2/λ is distribution-dependent and can be easily achieved if the distribution of $\|x\|$ is concentrated around zero. If we further have B'^2 is large enough comparing with $B\lambda$, the bound in $(\mathcal{B}_2)$ is tighter than the one in $(\mathcal{B}_1)$.

In particular, if x is a distribution such that

$$\mathcal{P}\left(\|x\| \leq \frac{B'}{6}\right) \geq \frac{23}{24},$$

*and $B'^2 \geq 8\sqrt{2}B\lambda$, $\delta < 0.5$, the bound obtained in $(\mathcal{B}_2)$ is tighter than the one in $(\mathcal{B}_1)$. Moreover, our locally elastic bound is **significantly tighter** than the one obtain via uniform stability, if $B'^2 \gg B\lambda$.*

Example 4.2 (Stability of regularized least square regression)**.** *Consider $\mathcal{Y} = [0, B]$ and denote $\mathcal{H}$ as the reproducing kernel Hilbert space induced by kernel K. The regularized least square regression algorithm is defined by*

$$\mathcal{A}_S = \arg\min_{h\in\mathcal{H}} \frac{1}{m}\sum_{j=1}^{m} l(h, z_j) + \lambda\|h\|_K^2,$$

where $l(f,z) = (f(x) - y)^2$. Then, with Lemma 4.1, we obtain the following bound via uniform stability

$$\Delta(\mathcal{A}_S) \leq \frac{4\kappa^2B^2}{\lambda m} + \left(\frac{8\kappa^2B^2}{\lambda} + B^2\right)\sqrt{\frac{\log(1/\delta)}{2m}}. \quad (\mathcal{B}_3)$$

Meanwhile, we can obtain the following bound via locally elastic stability

$$\Delta(\mathcal{A}_S) \leq \frac{4\kappa\mathbb{E}_x\kappa(x)B^2}{\lambda m} + \left(\frac{12\kappa\mathbb{E}_x\kappa(x)B^2}{\lambda} + 2B^2\right)\sqrt{\frac{2\log(2/\delta)}{m}}. \quad (\mathcal{B}_4)$$

Similarly, for simplicity, let us consider K to be the bilinear kernel $K(x,x') = \langle x, x'\rangle$ and all $x \in \mathcal{X}$'s norm are bounded by B'. $\kappa = B'^2$ and $\mathbb{E}_x\kappa(x) = \mathbb{E}_x\|x\|^2$. With the same spirit as in Example 4.1, if x is a distribution such that

$$\mathcal{P}\left(\|x\| \leq \frac{B'}{4}\right) \geq \frac{3}{4}$$

and suppose $B'^4 \geq \lambda$, $\delta < 0.5$, the bound obtained in $(\mathcal{B}_4)$ is tighter than the one in $(\mathcal{B}_3)$. Similar as Example 4.1, our locally elastic bound is significantly tighter than the one obtain via uniform stability, if $B'^4 \gg \lambda$.

4.2. Hypothesis Stability.

For a training set S with m examples, an algorithm $\mathscr{A}$ has hypothesis stability β_m^H with respect to the loss function l if

$$\mathbb{E}_{S,z}\left|l(\mathcal{A}_S, z) - l(\mathcal{A}_{S^{-i}}, z)\right| \leq \beta_m^H$$

holds for all $S \in \mathcal{Z}^m$ and $1 \leq i \leq m$, where β_m^H is a sequence of scalars . If $0 \leq l(\cdot,\cdot) \leq M_l$, for any $\delta \in (0,1)$ and sample size m, Bousquet & Elisseeff (2002) show that with probability at least $1-\delta$

$$\Delta(\mathcal{A}_S) \leq \sqrt{\frac{M_l^2 + 12M_lm\beta_m^H}{2m\delta}}.$$

For $\beta_m^H = O(1/m)$, hypothesis stability only provides a tail bound of order $O(1/\sqrt{m\delta})$ (polynomial tail bound) while locally elastic stability provides tail bounds of order $O(\sqrt{\log(1/\delta)/m})$ (exponential tail bound). In addition, for an algorithm $\mathscr{A}$ satisfying locally elastic stability $\beta_m(\cdot,\cdot)$, it by definition satisfies hypothesis stability with parameter $\mathbb{E}_{z',z}\beta_m(z',z)$.

4.3. Error Stability.

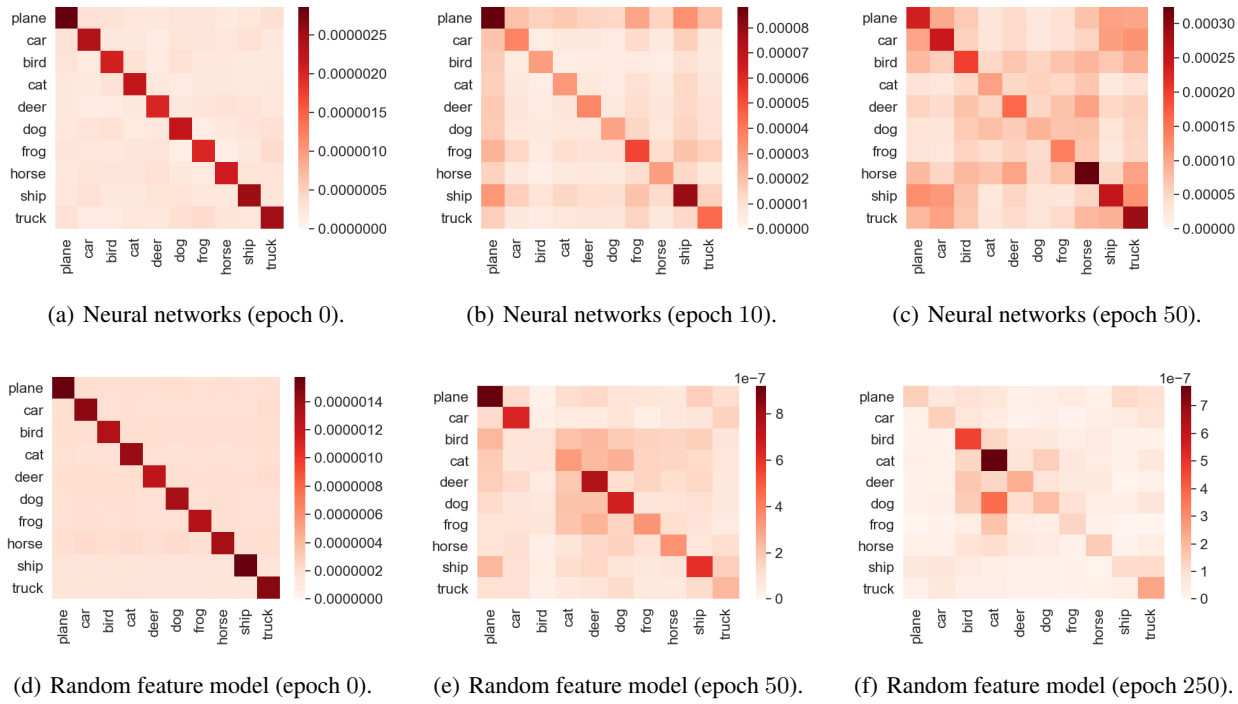
For a training set S with m examples, an algorithm $\mathscr{A}$ has error stability β_m^E with respect to the loss function l if

$$\left|\mathbb{E}_z[l(\mathcal{A}_S, z)] - \mathbb{E}_z[l(\mathcal{A}_{S^{-i}}, z)]\right| \leq \beta_m^E,$$

for all $S \in \mathcal{Z}^m$ and $1 \leq i \leq m$. Error stability is closely related to locally elastic stability in the sense that β_m^E can take the value of $\sup_{z'\in\mathcal{Z}}\mathbb{E}_z\beta_m(z',z)$. However, as pointed out by Kutin & Niyogi (2012), this notion is too weak to guarantee generalization in the sense that there exists an algorithm $\mathscr{A}$ where the error stability parameter goes to 0 as the sample size m tends to infinity but the generalization gap does not go to 0.

5. Locally Elastic Stability and Stochastic Gradient Descent

In Hardt et al. (2016), the authors demonstrate that SGD satisfies uniform stability under the standard Lipschitz and



(a) Neural networks (epoch 0). (b) Neural networks (epoch 10). (c) Neural networks (epoch 50).

(d) Random feature model (epoch 0). (e) Random feature model (epoch 50). (f) Random feature model (epoch 250).

Figure 2. Exact stepwise characterization of class-level sensitivity for neural networks and random feature models trained with different numbers of epochs by SGD on CIFAR-10. The class-level sensitivity for a stepwise update of SGD is $C'(c_a, c_b) = \frac{1}{|S_a| \cdot |\tilde{S}_b|} \sum_{z_i \in S_a} \sum_{z \in \tilde{S}_b} |l(\hat{\theta}_t - \eta \nabla_\theta l(\hat{\theta}_t, z_i), z) - l(\hat{\theta}_t, z)|$, where S_a denotes the set of examples with class a in the training data and $\tilde{S}_b$ denotes the set of examples with class b in the test data.

smoothness assumptions. As another concrete application of locally elastic stability, we revisit this problem and show that SGD also satisfies locally elastic stability under similar assumptions.

SGD algorithm consists of multiples steps of stochastic gradient updates $\hat{\theta}_{t+1} = \hat{\theta}_t - \eta_t \nabla_\theta l(\hat{\theta}_t, z_{i_t})$, where we allow the learning rate to change over time and η_t is the learning rate at time t, i_t is picked uniformly at random from $\{1, \cdots, m\}$. Throughout this subsection, we develop our results for a T-step SGD. For a randomized algorithm $\mathscr{A}$ like SGD, we can extend the definition of locally elastic stability just as Hardt et al. (2016) do for uniform stability (Definition 2.2). As shown in Figure 2, we further demonstrate the step-wise characterization of class-level sensitivity for neural networks (based on a pre-trained ResNet-18) and random feature models (based on a randomly initialized ResNet-18) trained for different numbers of epochs by SGD on CIFAR-10.

Definition 5.1. *A randomized algorithm $\mathscr{A}$ is $\beta_m(\cdot, \cdot)$-locally elastic stable if for all datasets $S \in \mathcal{Z}^n$, we have*

$$|\mathbb{E}_{\mathscr{A}}[l(\mathcal{A}_S, z)] - \mathbb{E}_{\mathscr{A}}[l(\mathcal{A}_{S^{-i}}, z)]| \leq \beta_m(z_i, z),$$

where the expectation is over the randomness embedded in the algorithm $\mathscr{A}$.

For SGD, the algorithm $\mathscr{A}$ outputs functions $\mathcal{A}_S$ and $\mathcal{A}_{S^{-i}}$ which are parameterized by $\hat{\theta}_T$ and $\hat{\theta}_T^{-i}$ and we further study whether there is a function $\beta_m(\cdot, \cdot)$ such that $|\mathbb{E}[l(\hat{\theta}_T, z)] - \mathbb{E}[l(\hat{\theta}_T^{-i}, z)]| \leq \beta_m(z_i, z)$, where the expectation is taken with respect to randomness coming from uniformly choosing the index at each iteration. Under similar settings as in Hardt et al. (2016), we develop estimates of the locally elastic stability parameters separately for convex, strongly convex, and non-convex cases. Due to the space constraint, we only show our results here for convex and non-convex cases and defer the treatment of the strongly convex case to Appendix.

Proposition 5.1 (Convex Optimization)**.** *Assume that the loss function $l(\cdot, z)$ is α-smooth and convex for all $z \in \mathcal{Z}$. In addition, $l(\cdot, z)$ is $L(z)$-Lipschitz and $L(z) < \infty$ for all $z \in \mathcal{Z}$: $|l(\theta, z) - l(\theta', z)| \leq L(z)\|\theta - \theta'\|$ for all θ, θ'. We further assume $L = \sup_{z \in \mathcal{Z}} L(z) < \infty$. Suppose that we run SGD with step sizes $\eta_t \leq 2/\alpha$ for T steps. Then,*

$$|\mathbb{E}[l(\hat{\theta}_T, z)] - \mathbb{E}[l(\hat{\theta}_T^{-i}, z)]| \leq \frac{(L + L(z_i))L(z)}{m} \sum_{t=1}^{T} \eta_t.$$

Proposition 5.2 (Non-convex Optimization)**.** *Assume that the loss function $l(\cdot, z)$ is non-negative and bounded for*

all $z \in \mathcal{Z}$. Without loss of generality, we assume $0 \leq l(\cdot, z) \leq 1$. In addition, we assume $l(\cdot, z)$ is α-smooth. We further assume $l(\cdot, z)$ is $L(z)$-Lipschitz and $L(z) < \infty$ for all $z \in \mathcal{Z}$ and $L = \sup_{z \in \mathcal{Z}} L(z) < \infty$. Suppose that we run SGD for T steps with monotonically non-increasing learning rate $\eta_t \leq c/t$ for some constant $c > 0$. Then,

$$|\mathbb{E}[l(\hat{\theta}_T, z)] - \mathbb{E}[l(\hat{\theta}_T^{-i}, z)]| \leq \gamma_m \phi_\alpha(m, T, z_i, z),$$

where $\phi_\alpha(m, T, z_i, z) = (c(L(z_i) + L)L(z)T^{\alpha c})^{\frac{1}{\alpha c+1}}$ and $\gamma_m = (1 + 1/(\alpha c))/(m-1)$.

From the propositions above, we see that SGD has locally elastic stability with parameter taking the form $\beta(\cdot, \cdot)/m$, where $\beta(\cdot, \cdot)$ is independent of m. This is consistent with our assumptions regarding the form of $\beta_m(\cdot, \cdot)$ in Section 3. We remark that unlike Hardt et al. (2016), our results use $\hat{\theta}_T^{-i}$ instead of $\hat{\theta}_T^i$ in order to be consistent with our definition in Definition 2.1, where $\hat{\theta}_T^i$ is the parameter obtained by training on S^i (replacing the ith element from S with another example instead of removing the ith element as in S^{-i}). This setting requires us to provide new techniques. Specifically, we construct new coupling sequences to obtain an upper bound on $|\mathbb{E}[l(\hat{\theta}_T, z)] - \mathbb{E}[l(\hat{\theta}_T^{-i}, z)]|$ (see more in the Appendix).

Comparison with results in Hardt et al. (2016) By using $L(z)$ instead of L, if $\mathbb{E}L(z) \ll L$, which holds for most common models in practice, we would expect to obtain a sharper generalization bound for SGD compared with the one derived using uniform stability in Hardt et al. (2016) according to the discussion in Section 4. Due to limited space, let us only compare Proposition 5.2 with Theorem 3.12 in Hardt et al. (2015) with a simple example as an illustration. With some abuse of notation, we still use $\Delta(\mathcal{A}_S)$ to denote $\mathbb{E}_z\mathbb{E}[l(\hat{\theta}_T, z)] - \sum_{i=1}^m \mathbb{E}[l(\hat{\theta}_T^{-i}, z)]/m$.

In Theorem 3.12 in Hardt et al. (2015), via uniform stability, under the assumptions in Proposition 5.2, one can obtain the following bound:

$$\Delta(\mathcal{A}_S) \leq 2\beta_m^U + (4m\beta_m^U + 1)\sqrt{\frac{\log(1/\delta)}{2m}}, \quad (\mathcal{B}_5)$$

where

$$\beta_m^U = \frac{1 + 1/(\alpha c)}{m-1} \left(2cL^2T^{\alpha c}\right)^{\frac{1}{\alpha c+1}}.$$

While via locally elastic stabilty, one can obtain

$$\Delta(\mathcal{A}_S) \leq \frac{2\sup_{z' \in \mathcal{Z}} \mathbb{E}_z \beta(z', z)}{m}$$
$$+2\left(2\sup_{z' \in \mathcal{Z}} \mathbb{E}_z \beta(z', z) + 1\right)\sqrt{\frac{2\log(2/\delta)}{m}}. \quad (\mathcal{B}_6)$$

where

$$\beta(z', z) = \frac{1 + 1/(\alpha c)}{m-1} \left(c(L(z') + L)\, L(z)T^{\alpha c}\right)^{\frac{1}{\alpha c+1}}.$$

Example 5.1. *Let us take $l(\theta, z) = z^2 e^{-\theta^2}$, where θ and z are all scalars, where $z \in [0, 1]$ and $\theta \in \mathbb{R}$. Apparently, the loss function is $\alpha = 2$-smooth with respect to z. Meanwhile,*

$$\frac{d}{d\theta} l(\theta, z) = -2z^2\theta e^{-\theta^2}.$$

The fact that $\theta e^{-\theta^2} \leq e^{-1/2}/\sqrt{2}$ leads to $L(z) = \sqrt{2}e^{-1/2}z^2$ and $L = \sqrt{2}e^{-1/2}$.

With the same spirit as in Example 4.1, if we choose learning rate $\eta \leq 1/t$, when $\delta < 0.5$, as long as $\sup_{z' \in \mathcal{Z}} \mathbb{E}_z \beta(z', z) < \sqrt{2}/8\beta_m^U$ and $\beta_m^U \geq 2\sqrt{2} - 1/2$, the bound obtained in $(\mathcal{B}_6)$ is tighter than the one in $(\mathcal{B}_5)$. It is easy to see that these conditions can be easily satisfied if T is large enough and $\mathbb{E}[(L(z))^{\frac{1}{3}}] < L^{\frac{1}{3}}$. Therefore, if we further have the condition that z lies in a small vicinity of 0 with high probability (for example, $\mathbb{P}\left(|z| \leq \frac{1}{2}\right) > \frac{2}{3}$), then the bound obtained in $(\mathcal{B}_6)$ would be tighter than the one in $(\mathcal{B}_5)$. In particular, if the training time T is long enough, the bound obtained in $(\mathcal{B}_6)$ would be significantly tighter than the one in $(\mathcal{B}_5)$.

6. Conclusion and Future Work

In this work, we introduce a new notion of algorithmic stability, which is a relaxation of uniform stability yes still gives rise to exponential generalization bounds. It also provides a promising direction to obtain useful theoretical bounds for demystifying the generalization ability of modern neural networks through the lens of local elasticity (He & Su, 2020). However, as shown in Theorem 3.1, we currently still require the sample size m to be large enough so that our theoretical results hold. Whether that requirement could be removed is worthy of further investigation. In addition, our bound is related to the constant M_l, which is typically very large in practice if we apply the bound to neural networks. Thus, an interesting question is to examine whether this constant could be improved or not.

Acknowledgements

We are grateful to Cynthia Dwork and Vitaly Feldman for inspiring discussions and constructive comments. This work was supported in part by NSF through CAREER DMS-1847415, CCF-1763665 and CCF-1934876, an Alfred Sloan Research Fellowship, the Wharton Dean's Research Fund, and Contract FA8750-19-2-0201 with the US Defense Advanced Research Projects Agency (DARPA).

References

Karim Abou-Moustafa and Csaba Szepesvári. An exponential efron-stein inequality for lq stable learning rules. *arXiv preprint arXiv:1903.05457*, 2019.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov): 463–482, 2002.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar): 499–526, 2002.

Olivier Bousquet, Yegor Klochkov, and Nikita Zhivotovskiy. Sharper bounds for uniformly stable algorithms. In *Conference on Learning Theory*, pp. 610–626, 2020.

Satrajit Chatterjee. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. *arXiv preprint arXiv:2002.10657*, 2020.

Shuxiao Chen, Hangfeng He, and Weijie J. Su. Label-aware neural tangent kernel: Toward better generalization and local elasticity. In *Advances in Neural Information Processing Systems*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Ieee, 2009.

Luc Devroye and Terry Wagner. Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Transactions on Information Theory*, 25(2):202–207, 1979.

Vitaly Feldman and Jan Vondrak. Generalization bounds for uniformly stable algorithms. In *Advances in Neural Information Processing Systems*, pp. 9747–9757, 2018.

Vitaly Feldman and Jan Vondrak. High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In *Conference on Learning Theory*, pp. 1270–1279, 2019.

Stanislav Fort, Paweł Krzysztof Nowak, Stanislaw Jastrzebski, and Srini Narayanan. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint arXiv:1901.09491*, 2019.

M Hardt, B Recht, and Y Singer. Train faster, generalize better: Stability of stochastic gradient descent. arxiv 2015. *arXiv preprint arXiv:1509.01240*, 2015.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1225–1234. PMLR, 2016.

Hangfeng He and Weijie J. Su. The local elasticity of neural networks. In *International Conference on Learning Representations*, 2020.

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

Michael Kearns and Dana Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural computation*, 11(6):1427–1453, 1999.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1885–1894, 2017.

A Krizhevsky. Learning multiple layers of features from tiny images. *Master?s thesis, Department of Computer Science, University of Toronto*, 2009.

Samuel Kutin and Partha Niyogi. Almost-everywhere algorithmic stability and generalization error. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pp. 275–282, 2002.

Samuel Kutin and Partha Niyogi. Almost-everywhere algorithmic stability and generalization error. *arXiv preprint arXiv:1301.0579*, 2012.

Ilja Kuzborskij and Christoph Lampert. Data-dependent stability of stochastic gradient descent. In *International Conference on Machine Learning*, pp. 2815–2824. PMLR, 2018.

Yunwen Lei and Yiming Ying. Fine-grained analysis of stability and generalization for stochastic gradient descent. *arXiv preprint arXiv:2006.08157*, 2020.

Gábor Lugosi and Miroslaw Pawlak. On the posterior-probability estimate of the error rate of nonparametric classification rules. *IEEE Transactions on Information Theory*, 40(2):475–481, 1994.

Liam Madden, Emiliano Dall'Anese, and Stephen Becker. High probability convergence and uniform stability bounds for nonconvex stochastic gradient descent. *arXiv preprint arXiv:2006.05610*, 2020.

Sayan Mukherjee, Partha Niyogi, Tomaso Poggio, and Ryan Rifkin. Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics*, 25(1-3):161–193, 2006.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 11:2635–2670, 2010.

V Vapnik. Estimation of dependences based on empirical data nauka, 1979.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. Capturing long-tail distributions of object subcategories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 915–922, 2014.