Black box tests for algorithmic stability

Byol Kim and Rina Foygel Barber

Kiho Park
parkkiho@uchicago.edu

1 Introduction

Consider an algorithm $\mathcal{A}$ that maps data sets to fitted regression functions:

$$\mathcal{A} : (X_1, Y_1), \cdots, (X_n, Y_n) \mapsto \widehat{\mu} : \mathbb{R}^d \to \mathbb{R}$$

Guarantees bounding the predictive error of the fitted model $\widehat{\mu}$ can still be obtained by testing its predictive accuracy on additional labeled data. We may not want to sacrifice sample size in order to obtain this independent holdout set. It is possible to give guarantees on predictive error by assuming a condition of stability on the algorithm $\mathcal{A}$. The stability condition essentially requires that slight perturbations of the training data set $((X_i, Y_i))_{i=1,\cdots,n}$ can alter the fitted model $\widehat{\mu}$ only slightly.

Unfortunately, verifying the stability of a given regression algorithm is itself challenging. We can only examine an algorithm's stability properties through empirical testing. Our aim is to examine the question of whether it is possible to infer the stability of an algorithm through black box testing that tries to determine the stability by observing the empirical behavior of the algorithm on various data sets.

2 Background and framework

2.1 Algorithmic stability

An algorithm $\mathcal{A}$ can be written as

$$A : \left(\bigcup_{n \geq 0} (\mathbb{R}^d \times \mathbb{R})^n \right) \times [0,1] \to \mathcal{M}_d$$

where $\mathcal{M}_d = \{\text{Measurable functions } \mu : \mathbb{R}^d \to \mathbb{R}\}$. Given n points $(x_1, y_1), \cdots, (x_n, y_n))$, the fitted regression function $\widehat{\mu}$ obtained from $\mathcal{A}$ is

$$\widehat{\mu} = \mathcal{A}[(x_1, y_1), \cdots, (x_n, y_n); \xi], \quad \xi \sim \text{Uniform}[0,1]$$

Here, the role of the last argument ξ is to encode the random behavior of $\mathcal{A}$, like the random seed. Throughout this paper, we assume that $\mathcal{A}$ is symmetric (same distribution by ξ). Also, we focus on the i.i.d. setting, i.e., $(X_i, Y_i) \overset{i.i.d.}{\sim} P$.

Definition 1. *Let $\mathcal{A}$ be a symmetric algorithm. Let $\epsilon \geq 0$ and $\delta \in [0,1)$. We say that $\mathcal{A}$ is (ϵ, δ)-stable with respect to training data sets of size n from data distribution P (or $(\mathcal{A}, P, n)$ is (ϵ, δ)-stable) if*

$$\mathbb{P}\{|\widehat{\mu}_n(X_{n+1}) - \widehat{\mu}_{n-1}(X_{n+1})| > \epsilon\} \leq \delta$$

where

$$\widehat{\mu}_n = \mathcal{A}[(X_1, Y_1), \cdots, (X_n, Y_n); \xi], \quad \widehat{\mu}_{n-1} = \mathcal{A}[(X_1, Y_1), \cdots, (X_{n-1}, Y_{n-1}); \xi']$$

where $\xi, \xi' \overset{i.i.d.}{\sim} \text{Uniform}[0,1]$.

2.1.1 Other notions of algorithmic stability

Various notions of algorithmic stability has been proposed, all describing different ways an algorithm may exhibit "continuity". Bousquet and Elisseeff (2002) shows that certain forms of algorithm stability implies good generalization behavior. Many recent works have also proposed extensions and modifications of existing notions of stability in the context of characterizing sufficient and necessary conditions for learning problems.

2.2 Testing for stability

Given a fixed $\epsilon \geq 0$ and $\delta \in [0, 1)$ and a desired confidence level $1 - \alpha$, we would like to test whether $(\mathcal{A}, P, n)$ is (ϵ, δ)-stable. Let $\mathcal{D}_\ell$ and $\mathcal{D}_u$ denote the available labeled and unlabeled data. We write $\widehat{T}_{\epsilon,\delta} = \widehat{T}_{\epsilon,\delta}(\mathcal{A}, \mathcal{D}_\ell, \mathcal{D}_u) \in \{0, 1\}$ for the data-dependent outcome of such a hypothesis test, where $\widehat{T}_{\epsilon,\delta} = 1$ indicates that we believe stability holds. Define the "ground truth" as

$$T^*_{\epsilon,\delta} = T^*_{\epsilon,\delta}(\mathcal{A}, P, n) = \mathbb{1}\{(\mathcal{A}, P, n) \text{ is } (\epsilon, \delta)\text{-stable}\}$$

Given this target, we seek a test $\widehat{T}_{\epsilon,\delta}$ that satisfies the following notion of assumption-free validity:

$$\text{For all } (\mathcal{A}, P, n), \text{ if } T^*_{\epsilon,\delta}(\mathcal{A}, P, n) = 0 \text{ then } \mathbb{P}\left\{\widehat{T}_{\epsilon,\delta} = 1\right\} \leq \alpha \tag{1}$$

2.3 Black box tests of stability

Let $\mathfrak{D}_\ell = \cup_{k\geq 0}(\mathbb{R}^d \times \mathbb{R})^k$ denote the space of labeled data sets of any size, and let $\mathfrak{D}_u = \cup_{k\geq 0}(\mathbb{R}^d)^k$ denote the space of unlabeled data sets of any size.

Definition 2 (Black box test)**.** *Consider any test $\widehat{T}$ that takes as input an algorithm $\mathcal{A} : \mathfrak{D}_\ell \to \mathcal{M}_d$, a labeled data set $\mathcal{D}_\ell \in \mathfrak{D}_\ell$, and an unlabeled data set $\mathcal{D}_u \in \mathfrak{D}_u$, and returns a (possibly randomized) binary output $\widehat{T}(\mathcal{A}, \mathcal{D}_\ell, \mathcal{D}_u) \in \{0, 1\}$. Then we say that $\widehat{T}$ is a black box test if for some measurable functions $f^{(1)}, f^{(2)}, \cdots$ and T, it can be defined in the following way:*

1. *At the initial stage $r = 1$,*
   (a) *Generate a labeled data set $\mathcal{D}^{(1)}_\ell \in \mathfrak{D}_\ell$ and an unlabeled data set $\mathcal{D}^{(1)}_u \in \mathfrak{D}_u$ as a function of the input data sets:*
   $$\left(\mathcal{D}^{(1)}_\ell, \mathcal{D}^{(1)}_u\right) = f^{(1)}[\mathcal{D}_\ell, \mathcal{D}_u, \zeta^{(1)}]$$
   where $\zeta^{(1)} \sim \text{Uniform}[0, 1]$.
   (b) *Fit a model and evaluate it using the generated data $\left(\mathcal{D}^{(1)}_\ell, \mathcal{D}^{(1)}_u\right)$:*
   $$\widehat{\mu}^{(1)} = \mathcal{A}\left[\mathcal{D}^{(1)}_\ell; \xi^{(1)}\right], \quad \widehat{\mathcal{Y}}^{(1)} = \widehat{\mu}^{(1)}\left[\mathcal{D}^{(1)}_u\right]$$
   where $\xi^{(1)} \sim \text{Uniform}[0, 1]$ and the operation of $\widehat{\mu}^{(1)}$ and $\mathcal{D}^{(1)}_u$ is understood in the pointwise sense.
2. *At each stage $r = 2, 3, \cdots$,*
   (a) *Generate a labeled data set $\mathcal{D}^{(r)}_\ell \in \mathfrak{D}_\ell$ and an unlabeled data set $\mathcal{D}^{(r)}_u \in \mathfrak{D}_u$ as a function of all the data sets and predictions observed so far:*
   $$\left(\mathcal{D}^{(r)}_\ell, \mathcal{D}^{(r)}_u\right) = f^{(r)}\left[\mathcal{D}_\ell, \mathcal{D}_u, \left(\mathcal{D}^{(s)}_\ell\right)^{r-1}_{s=1}, \left(\mathcal{D}^{(s)}_u\right)^{r-1}_{s=1}, \left(\widehat{\mathcal{Y}}^{(s)}\right)^{r-1}_{s=1}, \left(\zeta^{(s)}\right)^{r-1}_{s=1}, \left(\xi^{(s)}\right)^{r-1}_{s=1}, \zeta^{(r)}\right]$$
   where $\zeta^{(r)} \sim \text{Uniform}[0, 1]$.

(b) Fit a model and evaluate it using the generated data $\left(\mathcal{D}_\ell^{(r)}, \mathcal{D}_u^{(r)}\right)$*:*

$$\widehat{\mu}^{(r)} = \mathcal{A}\left[\mathcal{D}_\ell^{(r)}; \xi^{(r)}\right], \quad \widehat{\mathcal{Y}}^{(r)} = \widehat{\mu}^{(r)}\left[\mathcal{D}_u^{(r)}\right]$$

where $\xi^{(r)} \sim \text{Uniform}[0,1]$ *and the operation of* $\widehat{\mu}^{(r)}$ *and* $\mathcal{D}_u^{(r)}$ *is understood in the pointwise sense.*

3. *Compute the output* $\widehat{T} \in \{0,1\}$ *as a function of the sequence of generated data sets and observed predictions:*

$$\widehat{T} = T\left[\mathcal{D}_\ell, \mathcal{D}_u, \left(\mathcal{D}_\ell^{(r)}\right)_{r\geq 1}, \left(\mathcal{D}_u^{(r)}\right)_{r\geq 1}, \left(\widehat{\mathcal{Y}}^{(r)}\right)_{r\geq 1}, \left(\zeta^{(r)}\right)_{r\geq 1}, \left(\xi^{(r)}\right)_{r\geq 1}, \zeta\right]$$

where $\zeta \sim \text{Uniform}[0,1]$

2.3.1 Examples of black box tests

We write $\mathcal{D}_\ell = ((X_1, Y_1), \cdots, (X_{N_\ell}, Y_{N_\ell}))$ to denote the available labeled data and $\mathcal{D}_u = (X_{N_\ell+1}, \cdots, X_{N_\ell+N_u})$ for the available unlabeled data.

Example 1 (A simple Binomial test)**.** *Define*

$$\kappa = \kappa(n, N_\ell, N_u) = \min\left\{\frac{N_\ell}{n}, \frac{N_\ell + N_u}{n+1}\right\}$$

Then $\lfloor\kappa\rfloor$ *is the number of copies of independent data sets that can be constructed from* $\mathcal{D}_\ell$ *and* $\mathcal{D}_u$*.*

1. *For* $k = 1, \cdots, \lfloor\kappa\rfloor$*, construct the* k*-th data set with* $(X_{(k-1)n+1}, Y_{(k-1)n+1}), \cdots, (X_{kn}, Y_{kn})$ *and* $X_{\lfloor\kappa\rfloor n+k}$*.*
2. *For each* $k = 1, \cdots, \lfloor\kappa\rfloor$*, train models*

$$\widehat{\mu}_n^{(k)} = \mathcal{A}\left[(X_{(k-1)n+1}, Y_{(k-1)n+1}), \cdots, (X_{kn}, Y_{kn}); \xi^{(2k-1)}\right]$$

$$\widehat{\mu}_{n-1}^{(k)} = \mathcal{A}\left[(X_{(k-1)n+1}, Y_{(k-1)n+1}), \cdots, (X_{kn-1}, Y_{kn-1}); \xi^{(2k)}\right]$$

with $\xi^{(2k-1)}, \xi^{(2k)} \overset{i.i.d.}{\sim} \text{Uniform}[0,1]$*, and compute the difference in predictions*

$$\Delta^{(k)} = \left|\widehat{\mu}_n^{(k)}(X_{\lfloor\kappa\rfloor n+k}) - \widehat{\mu}_{n-1}^{(k)}(X_{\lfloor\kappa\rfloor n+k})\right|$$

3. *Compute how many of the* $\Delta^{(k)}$*'s exceed* ϵ*,* $B = \sum_{k=1}^{\lfloor\kappa\rfloor} \mathbb{1}\{\Delta^{(k)} > \epsilon\}$*, and compare this test statistic against the* Binomial$(\lfloor\kappa\rfloor, \delta)$ *distribution, returning* $\widehat{T}_{\epsilon,\delta} = 1$ *if* B *is sufficiently small or* $\widehat{T}_{\epsilon,\delta} = 0$ *otherwise.*

Example 2 (Bootstrapped sample method)**.**

1. *For* $r = 1, \cdots, R$*, construct a bootstrapped training sample* $(X_1^{(r)}, Y_1^{(r)}), \cdots, (X_n^{(r)}, Y_n^{(r)})$ *by sampling with replacement from* $\mathcal{D}_\ell$*. Sample a test point* $X_{n+1}^{(r)}$ *from* $\mathcal{D}_u$*.*
2. *As for Example 1, train models* $\widehat{\mu}_n^{(r)}$ *and* $\widehat{\mu}_{n-1}^{(r)}$*, and compute the difference in predictions* $\Delta^{(r)} = \left|\widehat{\mu}_n^{(r)}\left(X_{n+1}^{(r)}\right) - \widehat{\mu}_{n-1}^{(r)}\left(X_{n+1}^{(r)}\right)\right|$*. Then, compute how many of the* $\Delta^{(r)}$*'s exceed* ϵ*,* $\sum_{r=1}^{R} \mathbb{1}\{\Delta^{(r)} > \epsilon\}$ *and return* $\widehat{T}_{\epsilon,\delta} = 1$ *if this value is sufficiently small or* $\widehat{T}_{\epsilon,\delta} = 0$ *otherwise.*

Example 3 (Simulated sample method)**.**

1. *Use* $\mathcal{D}_\ell$ *and* $\mathcal{D}_u$ *to obtain an estimate* $\widehat{P}$ *of* P *via any method, then for* $r = 1, \cdots, R$*, draw a simulated data training set* $(X_1^{(r)}, Y_1^{(r)}), \cdots, (X_n^{(r)}, Y_1^{(r)}) \overset{i.i.d.}{\sim} \widehat{P}$ *and a simulated test point* $X_{n+1}^{(r)} \sim \widehat{P}_X$*.*
2. *Proceed as in Examples 1 and 2.*

2.3.2 Inadmissible tests

Tests that require uncountably infinitely many executions of a black box are not admissible under Definition 2. An example is a test $\widehat{T}$ that relies on evaluating $\sup_{y\in\mathbb{R}}\{\mathcal{A}[(X_1,Y_1),\cdots,(X_{n-1},Y_{n-1}),(X_n,y);\xi](X_{n+1})\}$.

2.3.3 Black box algorithms and black box functions

Definition 2 treats both the algorithm $\mathcal{A}$ itself as a black box, but also the resulting fitted functions $\widehat{\mu}$. That is, this definition allows for a setting where even the fitted functions themselves may be so complex that they can be studied only empirically.

3 Bounds for black box tests

In this section, we study the simple Binomial test defined in Example 1, and analyze its performance as a black-box test that achieves valid inference regardless of the algorithm $\mathcal{A}$ and distribution P. We then prove that the performance of this simple procedure is in fact essentially optimal among all black box tests.

3.1 A simple Binomial test

Since the $\Delta^{(k)}$'s are constructed on independent subsets of the data, we have $B\sim\text{Binomial}(\lfloor\kappa\rfloor,\delta^*_\epsilon)$, where $\delta^*_\epsilon=\mathbb{P}\{|\widehat{\mu}_n(X_{n+1})-\widehat{\mu}_{n-1}(X_{n+1})|>\epsilon\}$ is the unknown true probability. Therefore, we simply need to perform a one-tailed Binomial test on B to test whether $\delta^*_\epsilon\le\delta$ in which case $T^*_{\epsilon,\delta}=1$, or $\delta^*_\epsilon>\delta$ in which case $T^*_{\epsilon,\delta}=0$. Define $k^*_{\kappa,\delta}\in\{0,\cdots,\lfloor\kappa\rfloor-1\}$ and $a^*_{\kappa,\delta}\in(0,1]$ as the unique values satisfying

$$\mathbb{P}\{\text{Bin}(\lfloor\kappa\rfloor,\delta)<k\}+a^*_{\kappa,\delta}\cdot\mathbb{P}\{\text{Bin}(\lfloor\kappa\rfloor,\delta)=k\}=\alpha$$

and then define

$$\widehat{T}_{\epsilon,\delta}=\begin{cases}1 & \text{if } B<k^*_{\kappa,\delta}\\ \mathbb{1}\{\zeta\le a^*_{\kappa,\delta}\} & \text{if } B=k^*_{\kappa,\delta}\\ 0 & \text{if } B>k^*_{\kappa,\delta}\end{cases}$$

where $\zeta\sim\text{Unif}[0,1]$.

Clearly, this test is a black box test in the sense of Definition 22. We now verify the assumption-free validity property to ensure bounded error for detecting an algorithm is not stable, and calculate this test's power to detect when an algorithm is stable.

Theorem 1. *The black box test $\widehat{T}_{\epsilon,\delta}$ satisfies the assumption-free validity at level $1-\alpha$, i.e., for any $(\mathcal{A},P,n)$ that is not (ϵ,δ)-stable, it hold that $\mathbb{P}\left\{\widehat{T}_{\epsilon,\delta}=1\right\}\le\alpha$. Moreover, for any $(\mathcal{A},P,n)$ that is (ϵ,δ)-stable, if either $\delta^*_\epsilon=0$ or $\delta\le1-\alpha^{1/\lfloor\kappa\rfloor}$, then the power of the test is given by*

$$\mathbb{P}\left\{\widehat{T}_{\epsilon,\delta}=1\right\}=\left\{\alpha\cdot\left(\frac{1-\delta^*_\epsilon}{1-\delta}\right)^{\lfloor\kappa\rfloor}\right\}\wedge1$$

3.2 A bound on power for all black box tests

The simple black box test above appears to be a very naive and inefficient proposal. However, our next result establishes that, despite its simplicity, the simple Binomial test is optimal in some regimes, in the sense that no universally valid black box test can improve its power.

Theorem 2. *Fix any parameters $\epsilon \geq 0$ and $\delta \in [0,1)$, any Type I error level $\alpha \in (0,1)$, and any integers $n \geq 2$ and $N_\ell, N_u \geq 0$. Let $\widehat{T}_{\epsilon,\delta}$ be any black box test as in Definition 2, which satisfies the assumption-free validity at level $1-\alpha$. Then, for any $(\mathcal{A}, P, n)$ that is (ϵ,δ)-stable, the power of $\widehat{T}_{\epsilon,\delta}$ is bounded as*

$$\mathbb{P}\left\{\widehat{T}_{\epsilon,\delta} = 1\right\} \geq \left\{\alpha \cdot \left(\frac{1-\delta^*_\epsilon}{1-\delta}\right)^\kappa\right\} \wedge 1$$

where $\kappa = \kappa(n, N_\ell, N_u)$ is defined as above.

3.2.1 Matching upper and lower bounds

To understand, the implications of Theorem 2, let us compare the upper bound on power to the power achieved by our very simple test. In particular, in the case where κ is an integer, we see that the simple Binomial test is in fact optimal over all black box tests in the regime $\delta^*_\epsilon = 0$ or $\delta \leq 1 - \alpha^{1/\lfloor\kappa\rfloor}$.

3.2.2 Power and sample size

Is Theorem 2 a pessimistic result? The answer depends on how one wishes to use a stability test. First, we might simply be interested in characterizing the algorithm, by studying its behavior and its properties on different data distribution. In this setting, it is plausible that we might be interested in learning about the algorithm's stability at a sample size $n \ll N_\ell + N_u$. Alternatively, we may need to verify that stability hold in order to check assumptions for other procedures. For example, methods based on cross-validation can be used to construct predictive intervals around $\widehat{\mu}_n(X)$, but their coverage guarantees typically require algorithmic stability. In this case, we would probably want to test the algorithm's stability at $n = N_\ell$, then it would not be substantially better than random.

4 Alternative targets for inference

We define the associated targets as

$$\delta^*_\epsilon = \inf\{\delta \geq 0 : T^*_{\epsilon,\delta} = 1\} = \mathbb{P}\{|\widehat{\mu}_n(X_{n+1}) - \widehat{\mu}_{n-1}(X_{n+1})| > \epsilon\}$$

$$\epsilon^*_\delta = \inf\{\epsilon \geq 0 : T^*_{\epsilon,\delta} = 1\} = (1-\delta)\text{-quantile of } |\widehat{\mu}_n(X_{n+1}) - \widehat{\mu}_{n-1}(X_{n+1})|$$

By definition of (ϵ,δ)-stability, we see that for any (ϵ,δ), it hold that

$$T^*_{\epsilon,\delta} = 1 \iff \delta \geq \delta^*_\epsilon \iff \epsilon \geq \epsilon^*_\delta$$

Next, we define notions of assumption-free validity for inference on each of the new targets. For the problem of estimating δ^*_ϵ given a fixed $\epsilon \geq 0$, we seek an estimator $\widehat{\delta}_\epsilon$ such that

$$\mathbb{P}\left\{\widehat{\delta}_\epsilon \geq \delta^*_\epsilon\right\} \geq 1-\alpha \text{ for all } (\mathcal{A}, P, n) \tag{2}$$

Similarly, for the problem of estimating ϵ^*_δ given a fixed $\delta \in [0,1)$, we seek an estimator $\widehat{\epsilon}_\delta$ such that

$$\mathbb{P}\left\{\widehat{\epsilon}_\delta \geq \epsilon^*_\delta\right\} \geq 1-\alpha \text{ for all } (\mathcal{A}, P, n) \tag{3}$$

Proposition 1. *The following equivalence holds for inference procedures $\widehat{T}_{\epsilon,\delta}, \widehat{\delta}_\epsilon$, and $\widehat{\epsilon}_\delta$:*

(a) *For any fixed* $\epsilon \geq 0$, *suppose there is an estimator* $\widehat{\delta}_\epsilon$ *satisfying (2). Then for any* $\delta \in [0,1)$, *the test* $\widehat{T}_{\epsilon,\delta} = \mathbb{1}\{\widehat{\delta}_\epsilon \leq \delta\}$ *satisfies (1).*

(b) *For any fixed* $\delta \in [0,1)$, *suppose there is an estimator* $\widehat{\epsilon}_\delta$ *satisfying (3). Then for any* $\epsilon \geq 0$, *the test* $\widehat{T}_{\epsilon,\delta} = \mathbb{1}\{\widehat{\delta}_\epsilon \leq \delta\}$ *satisfies (1).*

(c) *Suppose* $\{\widehat{T}_{\epsilon,\delta} : \epsilon \geq 0, \delta \in [0,1]\}$ *is a family of tests such that each* $\widehat{T}_{\epsilon,\delta}$ *satisfies (1). Then, for any fixed* $\epsilon \geq 0$, *the estimator* $\widehat{\delta}_\epsilon = \inf\{\delta : \widehat{T}_{\epsilon,\delta'} = 1 \ \forall \ \delta' \geq \delta\}$ *satisfies (2).*

(d) *Similarly, under the same assumptions as in (c), for any fixed* $\delta \in [0,1)$, *the estimator* $\widehat{\epsilon}_\delta = \inf\{\epsilon : \widehat{T}_{\epsilon',\delta} = 1 \ \forall \ \epsilon' \geq \epsilon\}$ *satisfies (3).*

Proposition 1 implies that the problems of assumption-free inference for any of these three targets are essentially equivalent. Therefore, our conclusions apply to the other two inference problems as well to bound the power of any black box procedure.

5 Discussion

5.1 The role of uncountability

Our proof of Theorem 2 relies on finding a point x_* in the feature space $\mathbb{R}^d$, and a point y_* in the response space $\mathbb{R}$, with zero probability of being observed at any point throughout the course of the black box testing procedure. In particular, establishing the existence of such a point relies on the uncountability of $\mathbb{R}^d$ or of $\mathbb{R}$. However, in some settings either X or Y may a discrete random variable that is known to take values in a countable space $\mathbb{X} \subsetneq \mathbb{R}^d$ or $\mathbb{Y} \subsetneq \mathbb{R}$. For this more restricted definition of validity, the proof technique can no longer be carried out. Note that this setting is different than simply assuming that X or Y is discrete but not assuming a known support.

5.2 Black box algorithms and black box functions, revisited

There may be some settings where the algorithm $\mathcal{A}$ is sufficiently complex that it can only be thought of as a "black box" in any analysis, but a fitted function $\widehat{\mu}$ produced by the algorithm is simple enough to analyze theoretically. In this setting, we can use similar arguments as in Theorem 2 to establish an analogous bound on power with $\kappa = N_\ell/n$.

5.3 A different target: data-conditional stability

If the reason that we want to test stability is for verifying that predictive inference methods can be applied to $\mathcal{A}$, testing whether $(\mathcal{A}, P, n)$ is (ϵ, δ)-stable may be too strong of a goal. For example, for the jackknife's predictive coverage, we only need to test whether $\mathcal{A}$ is stable with respect to a random draw of the single last data point and we can condition on the training set. This notion of "data-conditional stability" appears to be a weaker property than the stability property of Definition 1, and therefore it may be possible to test this conditional property with higher power than the bounds established in Theorem 2.

References

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research,* 2:499 - 526, 2002.