Guaranteed Validity for Empirical Approaches to Adaptive Data Analysis

Ryan Rogers
Previously at University of Pennsylvania

Aaron Roth
University of Pennsylvania

Adam Smith
Boston University

Nathan Srebro
Toyota Technical Institute of Chicago

Om Thakkar
Boston University

Blake Woodworth
Toyota Technical Institute of Chicago

Abstract

We design a general framework for answering adaptive statistical queries that focuses on providing explicit confidence intervals along with point estimates. Prior work in this area has either focused on providing tight confidence intervals for specific analyses, or providing general worst-case bounds for point estimates. Unfortunately, as we observe, these worst-case bounds are loose in many settings — often not even beating simple baselines like sample splitting. Our main contribution is to design a framework for providing valid, instance-specific confidence intervals for point estimates that can be generated by heuristics. When paired with good heuristics, this method gives guarantees that are orders of magnitude better than the best worst-case bounds. We provide a Python library implementing our method.

1 Introduction

Many data analysis workflows are *adaptive*, i.e., they re-use data over the course of a sequence of analyses, where the choice of analysis at any given stage depends on the results from previous stages. Such adaptive re-use of data is an important source of *overfitting* in machine learning and *false discovery* in the empirical sciences [Gelman and Loken, 2014]. Adaptive workflows arise, for example, when exploratory data analysis is mixed with confirmatory data analysis, when hold-out sets are re-used to search through large hyper-parameter spaces or to perform feature selection, and when datasets are repeatedly re-used within a research community.

A simple solution to this problem—that we can view as a naïve benchmark—is to simply not re-use data. More precisely, one could use *sample splitting*: partitioning the dataset into k equal-sized pieces, and using a fresh piece of the dataset for each of k adaptive interactions with the data. This allows us to treat each analysis as nonadaptive, and allows many quantities of interest to be accurately estimated with their empirical estimate, and paired with tight confidence intervals that come from classical statistics. This seemingly naive approach is wasteful in its use of data, however: the sample size needed to conduct a series of k adaptive analyses grows linearly with k.

A line of recent work [Dwork et al., 2015c,a,b, Russo and Zou, 2016, Bassily et al., 2016, Rogers et al., 2016, Feldman and Steinke, 2017a,b, Xu and Raginsky, 2017, Zrnic and Hardt, 2019, Mania et al., 2019] aims to improve on this baseline by using mechanisms which provide "noisy" answers to queries rather than exact empirical answers. Methods coming from these works require that the sample size grow proportional to the *square root* of the number of adaptive analyses, dramatically beating the sample splitting baseline asymptotically. Unfortunately, the bounds proven in these papers—even when optimized—only beat the naïve baseline when both the dataset size n, and the number of adaptive rounds k, are very large; see Figure 1.

The failure of these worst-case bounds to beat simple baselines in practice — despite their attractive asymptotics — has been a major obstacle to the practical adoption of techniques from this literature. There are two difficulties with directly improving this style of bounds. The first is that we are limited by what we can prove: mathematical analyses can often be loose by constants that are significant in practice. The more fun-

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

damental difficulty is that these bounds are guaranteed to hold even against a *worst-case* data analyst, who is adversarially attempting to find queries which over-fit the sample: one would naturally expect that when applied to a real workload of queries, such worst-case bounds would be extremely pessimistic. We address both difficulties in this paper.

Contributions In this paper, we move the emphasis from algorithms that provide point estimates to algorithms that explicitly manipulate and output *confidence intervals* based on the queries and answers so far, providing the analyst with both an estimated value and a measure of its actual accuracy. At a technical level, we have two types of contributions:

First, we give optimized worst-case bounds that carefully combine techniques from different pieces of prior work—plotted in Figure 1. For certain mechanisms, our improved worst-case bounds are within small constant factors of optimal, in that we can come close to saturating their error bounds with a concrete, adversarial query strategy (Section 2). However, even these optimized bounds require extremely large sample sizes to improve over the naive sample splitting baseline, and their pessimism means they are often loose.

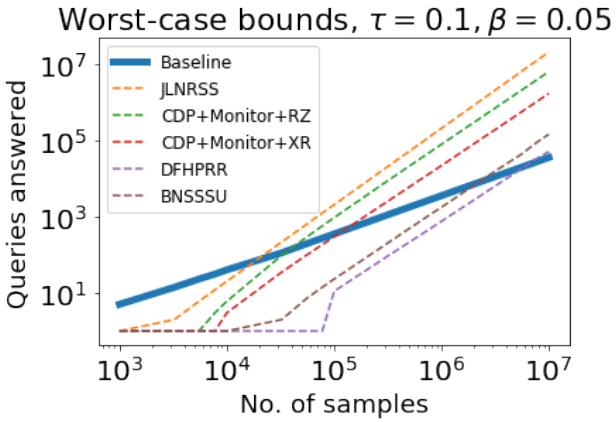

Figure 1: Comparison of various worst-case bounds for the Gaussian mechanism with the sample splitting baseline. 'DFHPRR' and 'BNSSSU' refer to bounds given in prior work (Dwork et al. [2015d], Bassily et al. [2016]). The other lines plot improved worst-case bounds derived in this paper, whereas 'JLNRSS' refers to bounds in subsequent work (Jung et al. [2019]). (See Section 2 for the full model and parameter descriptions.)

Our main result is the development of a simple framework called *Guess and Check*, that allows an analyst to pair *any method* for "guessing" point estimates and confidence interval widths for their adaptive queries, and then rigorously validate those guesses on an additional held-out dataset. So long as the analyst mostly guesses correctly, this procedure can continue indefinitely. The main benefit of this framework is that it allows the analyst to guess confidence intervals whose guarantees *exceed what is guaranteed by the worst-case theory*, and still enjoy rigorous validity in the event that they pass the "check". This makes it possible to take advantage of the non-worst-case nature of natural query strategies, and avoid the need to "pay for" constants that seem difficult to remove from worst-case bounds. Our empirical evaluation demonstrates that our approach can improve on worst-case bounds by orders of magnitude, and that it improves on the naive baseline even for modest sample sizes: see Figure 2, and Section 3 for details. We also provide a Python library containing an implementation of our Guess and Check framework.

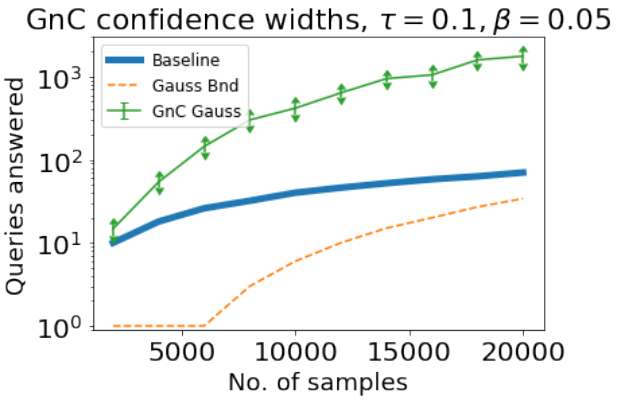

Figure 2: Performance of Guess and Check with the Gaussian mechanism providing the guesses ("GnC Gauss") for a plausible query strategy (see Section 3.1), compared with the best worst-case bounds for the Gaussian mechanism ("Gauss Bnd"), and the baseline.

Related Work Our "Guess and Check" (GnC) framework draws inspiration from the Thresholdout method of Dwork et al. [2015a], which uses a holdout set in a similar way. GnC has several key differences, which turn out to be crucial for practical performance. First, whereas the "guesses" in Thresholdout are simply the empirical query answers on a "training" portion of the dataset, we make use of other heuristic methods for generating guesses (including, in our experiments, Thresholdout itself) that empirically often seem to prevent overfitting to a substantially larger degree than their worst-case guarantees suggest. Second, we make confidence-intervals first-order objects: whereas the "guesses" supplied to Thresholdout are simply point estimates, the "guesses" supplied to GnC are point estimates along with confidence intervals. Finally, we use a more sophisticated analysis to track the number of bits leaked from the holdout, which lets us give tighter

confidence intervals and avoids the need to a priori set an upper bound on the number of times the holdout is used. Gossmann et al. [2018] use a version of Thresholdout to get worst-case accuracy guarantees for values of the AUC-ROC curve for adaptively obtained queries. However, apart from being limited to binary classification tasks and the dataset being used only to obtain AUC values, their bounds require "unrealistically large" dataset sizes. Our results are complementary to theirs; by using appropriate concentration inequalities, GnC could also be used to provide confidence intervals for AUC values. Their technique could be used to provide the "guesses" to GnC.

Our improved worst-case bounds combine a number of techniques from the existing literature: namely the information theoretic arguments of Russo and Zou [2016], Xu and Raginsky [2017] together with the "monitor" argument of Bassily et al. [2016], and a more refined accounting for the properties of specific mechanisms using *concentrated differential privacy* (Dwork and Rothblum [2016], Bun and Steinke [2016b]).

Feldman and Steinke [2017a,b] give worst-case bounds that improve with the variance of the asked queries. In Section 3.1, we show how GnC can be used to give tighter bounds when the empirical query variance is small.

Mania et al. [2019] give an improved union bound for queries that have high overlap, that can be used to improve bounds for adaptively validating similar models, in combination with description length bounds. Zrnic and Hardt [2019] take a different approach to going beyond worst-case bounds in adaptive data analysis, by proving bounds that apply to data analysts that may only be adaptive in a constrained way. A difficulty with this approach in practice is that it is limited to analysts whose properties can be inspected and verified — but provides a potential explanation why worst-case bounds are not observed to be tight in real settings. Our approach is responsive to the degree to which the analyst actually overfits, and so will also provide relatively tight confidence intervals if the analyst satisfies the assumptions of Zrnic and Hardt [2019].

In very recent work (subsequent to this paper), Jung et al. [2019] give a further tightening of the worst-case bounds, improving the dependence on the coverage probability β. Their bounds (shown in Figure 1) do not significantly affect the comparison with our GnC method since they yield only worst-case analysis.

1.1 Preliminaries

As in previous work, we assume that there is a dataset $X = (x_1, \cdots, x_n) \sim \mathcal{D}^n$ drawn i.i.d. from an unknown distribution $\mathcal{D}$ over a universe $\mathcal{X}$. This dataset is the input to a mechanism $\mathcal{M}$ that also receives a sequence of queries $\phi_1, \phi_2, ...$ from an analyst $\mathcal{A}$ and outputs, for each one, an answer. Each ϕ_i is a *statistical query*, defined by a bounded function $\phi_i : \mathcal{X} \to [0, 1]$. We denote the expectation of a statistical query ϕ over the data distribution by $\phi(\mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}}[\phi(x)]$, and the empirical average on a dataset by $\phi(X) = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)$.

The mechanism's goal is to give estimates of $\phi_i(\mathcal{D})$ for query ϕ_i on the unknown $\mathcal{D}$. Previous work looked at analysts that produce a single point estimate a_i, and measured error based on the distances $|a_i - \phi_i(\mathcal{D})|$. As mentioned above, we propose a shift in focus: we ask mechanisms to produce a confidence interval specified by a point estimate a_i and width τ_i. The answer (a_i, τ_i) is *correct for* ϕ_i *on* $\mathcal{D}$ if $\phi_i(\mathcal{D}) \in (a_i - \tau_i, a_i + \tau_i)$. (Note that the data play no role in the definition of correctness—we measure only population accuracy.)

An interaction between randomized algorithms $\mathcal{M}$ and $\mathcal{A}$ on dataset $X \in \mathcal{X}^n$ (denoted $\mathcal{M}(X) \rightleftharpoons \mathcal{A}$) consists of an unbounded number of query-answer rounds: at round i, $\mathcal{A}$ sends ϕ_i, and $\mathcal{M}(X)$ replies with (a_i, τ_i). $\mathcal{M}$ receives X as input. $\mathcal{A}$ receives no direct input, but may select queries *adaptively*, based on the answers in previous rounds. The interaction ends when either the mechanism or the analyst stops. We say that the mechanism provides simultaneous coverage if, with high probability, *all* its answers are correct:

Definition 1.1 (Simultaneous Coverage)**.** Given $\beta \in (0, 1)$, we say that $\mathcal{M}$ has *simultaneous coverage* $1 - \beta$ if, for all $n \in \mathbb{N}$, all distributions $\mathcal{D}$ on $\mathcal{X}$ and all randomized algorithms $\mathcal{A}$,

$$\Pr_{\substack{X\sim\mathcal{D}^n,\\ \{(\phi_i,a_i,\tau_i)\}_{i=1}^{k}\leftarrow(\mathcal{M}(X)\rightleftharpoons\mathcal{A})}}[\forall i \in [k] : \phi_i(\mathcal{D}) \in a_i \pm \tau_i] \geq 1-\beta$$

We denote by k the (possibly random) number of rounds in a given interaction.

Definition 1.2 (Accuracy)**.** We say $\mathcal{M}$ is (τ, β)-*accurate*, if $\mathcal{M}$ has simultaneous coverage $1 - \beta$ and its interval widths satisfy $\max_{i\in[k]} \tau_i \leq \tau$ with probability 1.

2 Confidence intervals from worst-case bounds

Our emphasis on explicit confidence intervals led us to derive worst-case bounds that are as tight as possible given the techniques in the literature. We discuss the Gaussian mechanism here, and defer the application to Thresholdout to the supplementary material.

The Gaussian mechanism is defined to be an algorithm that, given input dataset $X \sim \mathcal{D}^n$ and a query $\phi : \mathcal{X} \to [0, 1]$, reports an answer $a = \phi(X) + N\left(0, \frac{1}{2n^2\rho}\right)$, where $\rho > 0$ is a parameter. It has existing analyses

for simultaneous coverage (see Dwork et al. [2015d], Bassily et al. [2016]) — but these analyses involve large, sub-optimal constants. Here, we provide an improved worst-case analysis by carefully combining existing techniques. We use results from Bun and Steinke [2016a] to bound the mutual information of the output of the Gaussian mechanism with its input. We then apply an argument similar to that of Russo and Zou [2016] to bound the bias of the empirical average of a statistical query selected as a function of the perturbed outputs. Finally, we use Chebyshev's inequality, and the monitor argument from Bassily et al. [2016] to obtain high probability accuracy bound. Figure 1 shows the improvement in the number of queries that can be answered with the Gaussian mechanism with $(0.1, 0.05)$-accuracy. Our guarantee is stated below, with its proof deferred to the supplementary material.

Theorem 2.1. *Given input $X \sim \mathcal{D}^n$, confidence parameter β, and parameter ρ, the Gaussian mechanism is (τ, β)-accurate, where* $\tau = \sqrt{\frac{1}{2n\beta} \cdot \min_{\lambda \in [0,1)} \left(\frac{2\rho k n - \ln(1-\lambda)}{\lambda} \right)} + \frac{1}{2n} \sqrt{\frac{1}{\rho} \ln\left(\frac{4k}{\beta}\right)}$.

We now consider the extent to which our analyses are improvable for worst-case queries to the Gaussian and the Thresholdout mechanisms. To do this, we derive the worst query strategy in a particular restricted regime. We call it the "single-adaptive query strategy", and show that it maximizes the root mean squared error (RMSE) amongst all single query strategies under the assumption that each sample in the dataset is drawn u.a.r. from $\{-1, 1\}^{k+1}$, and the strategy is given knowledge of the empirical correlations of each of the first k features with the $(k+1)$st feature (which can be obtained e.g. with k non-adaptive queries asked prior to the adaptive query). We provide a pseudocode for the strategy and its analysis in the supplementary material. To make the bounds comparable, we translate our accuracy upper bounds for both the mechanisms to RMSE bounds (deferred to the supplementary material). Figure 3 shows the difference between our best upper bound and the realized RMSE (averaged over 100 executions) for the two mechanisms using $n = 5,000$ and various values of k. (For the Gaussian, we set ρ separately for each k, to minimize the upper bound.) On the top, we see that the two bounds for the Gaussian mechanism are within a factor of 2.5, even for $k = 50,000$ queries. Our bounds are thus reasonably tight in one important setting. For Thresholdout (bottom side), however, we see a large gap between the bounds which grows with k, even for our best query strategy[1]. This result points to the promise for empirically-based confidence intervals for complex mechanisms that are harder to analyze.

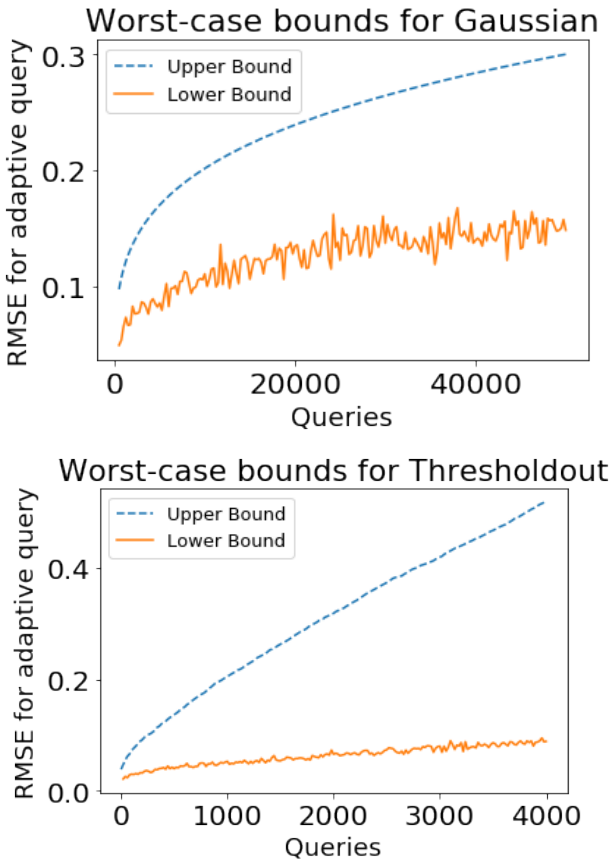

Figure 3: Worst-case upper (proven) and lower RMSE bounds (realized via the single-adaptive query strategy) with $n = 5,000$ for Gaussian (top) and Thresholdout (bottom).

3 The Guess and Check Framework

In light of the inadequacy of worst-case bounds, we here present our Guess and Check (GnC) framework which can go beyond the worst case. It takes as inputs *guesses* for both the point estimate of a query, and a confidence interval width. If GnC can validate a guess, it releases the guess. Otherwise, at the cost of widening the confidence intervals provided for future guesses, it provides the guessed confidence width along with a point estimate for the query using the holdout set such that the guessed width is valid for the estimate.

An instance of GnC, $\mathcal{M}$, takes as input a dataset X, desired confidence level $1 - \beta$, and a mechanism $\mathcal{M}_g$ which operates on inputs of size $n_g < n$. $\mathcal{M}$ randomly

[1] We tweak the adaptive query in the single-adaptive query strategy to result in maximum error for Thresholdout. We also tried "tracing" attack strategies (adapted from the fingerprinting lower bounds of Bun et al. [2014], Hardt and Ullman [2014], Steinke and Ullman [2015]) that contained multiple adaptive queries, but gave similar results.

splits X into two, giving one part X_g to $\mathcal{M}_g$, and reserving the rest as a holdout X_h. For each query ϕ_i, mechanism $\mathcal{M}_g$ uses X_g to make a "guess" $(a_{g,i}, \tau_i)$ to $\mathcal{M}$, for which $\mathcal{M}$ conducts a validity check. If the check succeeds, then $\mathcal{M}$ releases the guess as is, otherwise $\mathcal{M}$ uses the holdout X_h to provide a response containing a discretized answer that has τ_i as a valid confidence interval. This is closely related to Thresholdout. However, an important distinction is that the width of the target confidence interval, rather than just a point estimate, is provided as a guess. Moreover, the guesses themselves can be made by non-trivial algorithms.

Algorithm 1 Guess and Check

Require: Data $X \in \mathcal{X}^n$, confidence parameter β, analyst having mechanism $\mathcal{M}_g$ with inputs of size $n_g < n$

Randomly split X into a guess set X_g of size n_g to input into $\mathcal{M}_g$, and a holdout X_h of size $n_h = n - n_g$
$f \leftarrow 0$
$c_j \leftarrow \frac{6}{\pi^2(j+1)^2}$ for $j \geq 0$ //should just satisfy $\sum_{j\geq 0} c_j \leq 1$
for $i = 1$ to ∞ **do**
  //Compute # possible transcripts
  if $f > 0$ **then**
    $\nu_{i,f,\gamma_1^f} \leftarrow \binom{i-1}{f} \prod_{j\in[f]} \left(\frac{1}{\gamma_j}\right)$
  else
    $\nu_{i,f,\gamma_1^f} \leftarrow 1$
  $\beta_i \leftarrow (\beta \cdot c_{i-1} \cdot c_f)/\nu_{i,f,\gamma_1^f}$
  Receive query ϕ_i and guess $(a_{g,i}, \tau_i) \leftarrow \mathcal{M}_g(X_g, \phi_i)$ from analyst
  $a_{h,i} \leftarrow \phi_i(X_h)$ //holdout answer
  $\tau_h \leftarrow HoldoutTol(\beta_i, a_{g,i}, \tau_i, a_{h,i})$ //*HoldoutTol* returns a valid tolerance for $a_{h,i}$
  if $|a_{g,i} - a_{h,i}| \leq \tau_i - \tau_h$ **then**
    Output $(a_{g,i}, \tau_i)$ to analyst
  else
    $f \leftarrow f + 1$
    $\gamma_f \leftarrow \max_{[0,\tau_i)} \gamma$ s.t. $2e^{-2(\tau_i-\gamma)^2 n_h} \leq \beta_i$ //max. discretization parameter with validity
    if $\gamma_f > 0$ **then**
      Output $\left(\lfloor a_{h,i} \rfloor_{\gamma_f}, \tau_i\right)$ to analyst, where $\lfloor y \rfloor_\gamma$ denotes y discretized to multiples of γ
    else
      Output $\perp$ to analyst
      break //Terminate **for** loop

Depending on how long one expects/requires GnC to run, the input confidence parameter β can guide the minimum value of the holdout size n_h that will be required for GnC to be able to get a holdout width τ_h smaller than the desired confidence widths $\tau_i, \forall i \geq 1$. Note that this can be evaluated before starting GnC. Apart from that, we believe what is a good split will largely depend on the Guess mechanism. Hence, in general the split parameter should be treated as a hyperparameter for our GnC method. We provide pseudocode for GnC in Algorithm 1, and a block schematic of how a query is answered by GnC in Figure 4.

We provide coverage guarantees for GnC without any restrictions on the guess mechanism. To get the guarantee, we first show that for query ϕ_i, if function *HoldoutTol* returns a $(1-\beta_i)$-confidence interval τ_h for holdout answer $a_{h,i}$, and GnC's output is the guess $(a_{g,i}, \tau_i)$, then τ_i is a $(1-\beta_i)$-confidence interval for $a_{g,i}$. We can get a simple definition for *HoldoutTol* (formally stated in the supplementary material) via an application of Chernoff bound, but we provide a slightly sophisticated variant below that uses the guess and holdout answers to get better tolerances, especially under *low-variance* queries. We defer the proof of Lemma 3.1 to the supplementary material.

Lemma 3.1. *If the function HoldoutTol in GnC (Algorithm 1) is defined as*

$$HoldoutTol(\beta', a_g, \tau, a_h) = \begin{cases} \arg\min_{\tau' \in (0,\tau)} \left\{ \begin{array}{c} \left(\frac{1+\mu(e^\ell-1)}{e^{\ell(\mu+\tau')}}\right)^n \leq \frac{\beta}{2}, \\ \text{where } \ell \text{ solves} \\ \frac{\mu e^\ell}{1+\mu(e^\ell+1)} = \mu + \tau' \end{array} \right\} & \text{if } a_g > a_h, \\ \arg\min_{\tau' \in (0,\tau)} \left\{ \begin{array}{c} \left(\frac{1+\mu'(e^\ell-1)}{e^{\ell(\mu'+\tau')}}\right)^n \leq \frac{\beta}{2}, \\ \text{where } \ell \text{ solves} \\ \frac{\mu' e^\ell}{1+\mu'(e^\ell+1)} = \mu' + \tau' \end{array} \right\} & \text{o.w.} \end{cases}$$

where $\mu = a_g - \tau$ and $\mu' = 1 - a_g - \tau$, then for each query ϕ_i s.t. GnC's output is $(a_{g,i}, \tau_i)$, we have $\Pr\left(|a_{g,i} - \phi_i(\mathcal{D})| > \tau_i\right) \leq \beta_i$.

Next, if failure f occurs within GnC for query ϕ_i, by applying a Chernoff bound we get that γ_f is the maximum possible discretization parameter s.t. τ_i is a $(1-\beta_i)$-confidence interval for the discretized holdout answer $\lfloor a_{h,i} \rfloor_{\gamma_f}$. Finally, we get a simultaneous coverage guarantee for GnC by a union bound over the error probabilities of the validity over all possible transcripts between GnC and any analyst $\mathcal{A}$ with adaptive queries $\{\phi_1, \ldots, \phi_k\}$. The guarantee is stated below, with its proof deferred to the supplementary material.

Theorem 3.2. *The Guess and Check mechanism (Algorithm 1), with inputs dataset $X \sim \mathcal{D}^n$, confidence parameter β, and mechanism $\mathcal{M}_g$ that, using inputs of size $n_g < n$, provides responses ("guesses") of the form $(a_{g,i}, \tau_i)$ for query ϕ_i, has simultaneous coverage $1 - \beta$.*

3.1 Experimental evaluation

Now, we provide details of our empirical evaluation of the Guess and Check framework. In our experiments, we use two mechanisms, namely the Gaussian mechanism and Thresholdout, for providing guesses in GnC.

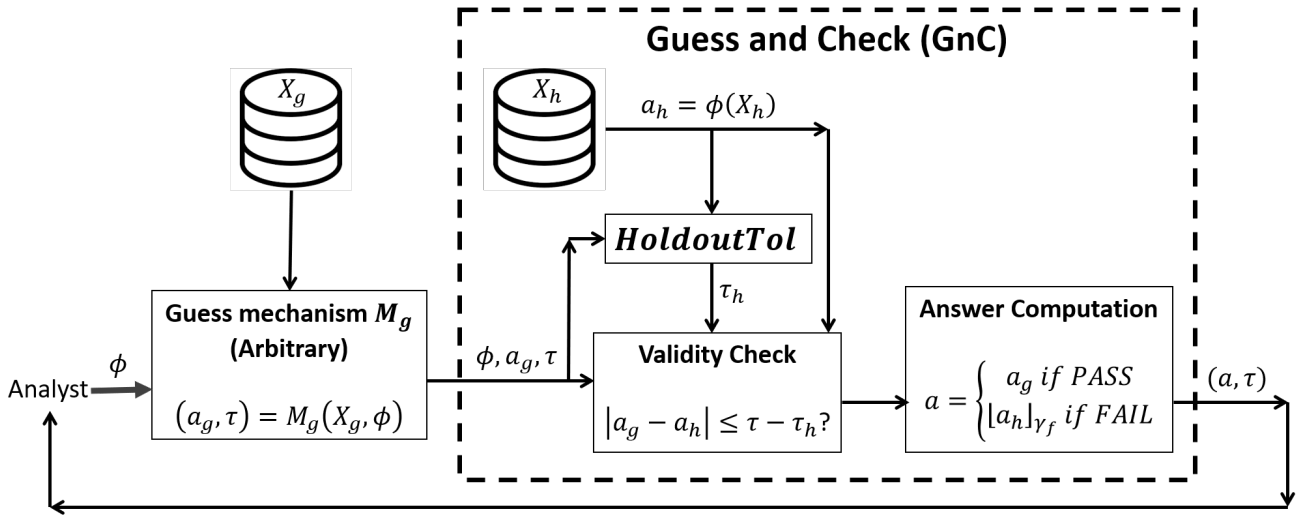

Figure 4: A schematic of how query ϕ is answered via our Guess and Check (GnC) framework. Dataset X_g is the guess set randomly partitioned by GnC. The dotted box represents computations that are previleged, and are not accessible to the analyst.

For brevity, we refer to the overall mechanism as GnC Gauss when the Gaussian is used to provide guesses, and GnC Thresh when Thresholdout is used.

Strategy for performance evaluation: Some mechanisms evaluated in our experiments provide worst-case bounds, whereas the performance of others is instance-dependent and relies on the amount of adaptivity present in the querying strategy. To highlight the advantages of the latter, we design a query strategy called the quadratic-adaptive query strategy. Briefly, it contains two types of queries: random non-adaptive queries in which each sample's contribution is generated i.i.d. from a Bernoulli distribution, and adaptive queries which are linear combinations of previous queries. The adaptive queries become more sparsely distributed with time; "hard" adaptive queries $\phi_i, i > 1$, are asked when i is a perfect square. They are computed in a similar manner as in the strategy used in Figure 3. We provide pseudocode for the strategy in the supplementary material.

Experimental Setup: We run the quadratic-adaptive strategy for up to $40,000$ queries. We tune the hyperparameters of each mechanism to optimize for this query strategy. We fix a confidence parameter β and set a target upper bound τ on the maximum allowable error we can tolerate, given our confidence bound. We evaluate each mechanism by the number of queries it can empirically answer with a confidence width of τ for our query strategy while providing a simultaneous coverage of $1-\beta$: i.e. the largest number of queries it can answer while providing (τ, β)-accuracy.

We plot the average and standard deviation of the number of queries k answered before it exceeds its target error bound in 20 independent runs over the sampled data and the mechanism's randomness. When we plot the actual realized error for any mechanism, we denote it by dotted lines, whereas the provably valid error bounds resulting from the confidence intervals produced by GnC are denoted by solid lines. Note that the empirical error denoted by dotted lines is not actually possible to know without access to the distribution, and is plotted just to visualize the tightness of the provable confidence intervals. We compare to two simple baselines: sample splitting, and answer discretization: the better of these two is plotted as the thick solid line. For comparison, the best worst-case bounds for the Gaussian mechanism (Theorem 2.1) are shown as dashed lines. Note that we improve by roughly two orders of magnitude compared to the tightest bounds for the Gaussian. We improve over the baseline at dataset sizes $n \geq 2,000$.

Boost in performance for low-variance queries: Since all the queries we construct take binary values on a sample $x \in \mathcal{X}$, the variance of query ϕ_i is given by $var(\phi_i) = \phi_i(\mathcal{D})(1-\phi_i(\mathcal{D}))$, as $\phi_i(\mathcal{D}) = \Pr(\phi_i(x) = 1)$. Now, $var(\phi_i)$ is maximized when $\phi_i(\mathcal{D}) = 0.5$. Hence, informally, we denote query ϕ_i as low-variance if either $\phi_i(\mathcal{D}) \ll 0.5$, or $\phi_i(\mathcal{D}) \gg 0.5$. We want to be able to adaptively provide tighter confidence intervals for low-variance queries (as, for e.g., the worst-case bounds of Feldman and Steinke [2017a,b] are able to). For instance, in Figure 5, we show that in the presence of

low-variance queries, using Lemma 3.1 for *HoldoutTol* (plot labelled "GnC Check:MGF") results in a significantly better performance for GnC Gauss as compared to using a Chernoff bound (plot labelled "GnC Check:Chern"). We fix $\tau, \beta = 0.05$, and set $\phi_i(\mathcal{D}) = 0.9$ for $i \geq 1$. We can see that as the dataset size grows, using Lemma 3.1 provides an improvement of almost 2 orders of magnitude in terms of the number of queries k answered. This is due to Lemma 3.1 providing tighter holdout tolerances τ_h for low-variance queries (with guesses close to 0 or 1), compared to those obtained via the Chernoff bound (agnostic to the query variance). Thus, we use Lemma 3.1 for *HoldoutTol* in all experiments with GnC below. The worst-case bounds for the Gaussian don't promise a coverage of $1 - \beta$ even for $k = 1$ in the considered parameter ranges. This is representative of a general phenomenon: switching to GnC-based bounds instead of worst-case bounds is often the difference between obtaining useful vs. vacuous guarantees.

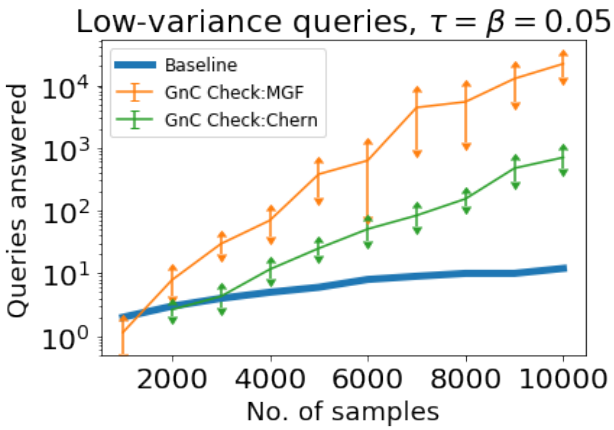

Figure 5: Gain in performance for GnC Gauss by using Lemma 3.1 for *HoldoutTol* ("GnC Check:MGF"), as compared to the simple variant obtained via Chernoff bound ("GnC Check:Chern").

Performance at high confidence levels: The bounds we prove for the Gaussian mechanism, which are the best known worst-case bounds for the considered sample size regime, have a substantially suboptimal dependence on the coverage parameter β : $\sqrt{1/\beta}$. On the other hand, sample splitting (and the bounds from Dwork et al. [2015d], Bassily et al. [2016] which are asymptotically optimal but vacuous at small sample sizes) have a much better dependence on β : $\ln(1/2\beta)$. Since the coverage bounds of GnC are strategy-dependent, the dependence of τ on β is not clear a priori. In Figure 6, we show the performance of GnC Gauss (labelled "GnC") when $\beta \in \{0.05, 0.005\}$. We see that reducing β by a factor of 10 has a negligible effect on GnC's performance. Note that this is the case even though the guesses are provided by the Gaussian, for which we do not have non-vacuous bounds with a mild dependence on β in the considered parameter range (see the worst-case bounds, plotted as "Bnd") — even though we might conjecture that such bounds exist. This gives an illustration of how GnC can correct deficiencies in our worst-case theory: conjectured improvements to the theory can be made rigorous with GnC's certified confidence intervals.

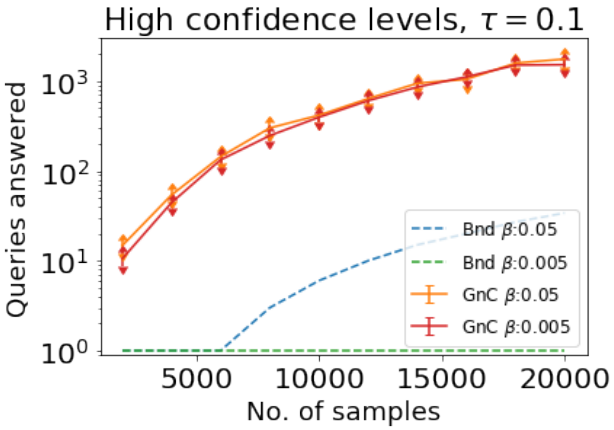

Figure 6: Performance of GnC Gauss ("GnC"), and the best Gaussian bounds ("Bnd"), for $\beta \in \{0.05, 0.005\}$.

GnC with different guess mechanisms: GnC is designed to be modular, enabling it to take advantage of arbitrarily complex mechanisms to make guesses. Here, we compare the performance of two such mechanisms for making guesses, namely the Gaussian mechanism, and Thresholdout. In Figure 7, we first plot the number of queries answered by the Gaussian ("Gauss Emp") and Thresholdout ("Thresh Emp") mechanisms, respectively, until the maximum empirical error of the query answers exceeds $\tau = 0.1$. It is evident that Thresholdout, which uses an internal holdout set to answer queries that likely overfit to its training set, provides better performance than the Gaussian mechanism. In fact, we see that for $n > 5000$, while Thresholdout is always able to answer $40,000$ queries (the maximum number of queries we tried in our experiments), the Gaussian mechanism isn't able to do so even for the largest dataset size we consider. Note that the "empirical" plots are generally un-knowable in practice, since we do not have access to the underlying distributions. But they serve as upper bounds for the best performance a mechanism can provide.

Next, we fix $\beta = 0.05$, and plot the performance of GnC Gauss and GnC Thresh. We see that even though GnC Thresh has noticeably higher variance, it provides performance that is close to two orders of magnitude larger than GnC Gauss when $n \geq 8000$. Moreover, for

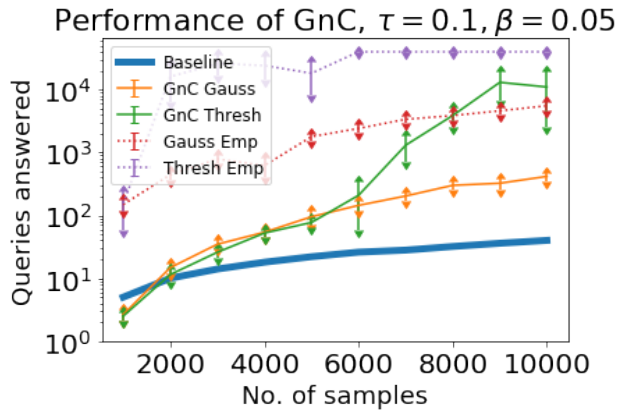

Figure 7: Performance of GnC with Gaussian ("GnC Gauss"), and Thresholdout ("GnC Thresh") mechanisms, together with their empirical error.

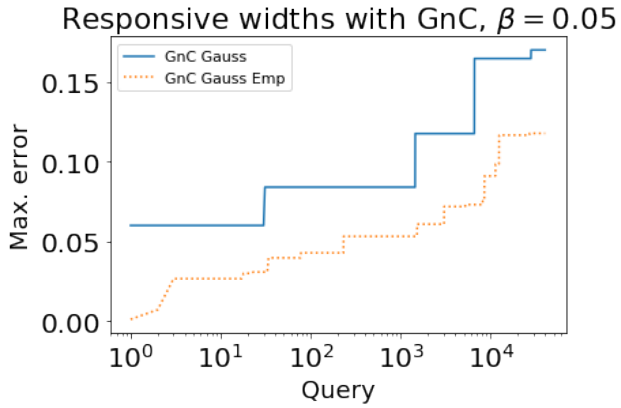

Figure 8: The accuracy of GnC with Gaussian guesses provides "responsive" confidence interval widths that closely track the empirical error incurred by the guesses of the Gaussian mechanism ("GnC Gauss Emp").

$n \geq 8000$, it is interesting to see GnC Thresh guarantees (τ, β)-accuracy for our strategy while consistently beating even the empirical performance of the Gaussian. We note that the best bounds for both the Gaussian and Thresholdout mechanisms alone (not used as part of GnC) do not provide any non-trivial guarantees in the considered parameter ranges.

Responsive widths that track the empirical error: The GnC framework is designed to certify guesses which represent both a point estimate and a desired confidence interval width for each query. Rather than having fixed confidence interval widths, this framework also provides the flexibility to incorporate guess mechanisms that provide increased interval widths as failures accumulate within GnC. This allows GnC to be able to re-use the holdout set in perpetuity, and answer an infinite number of queries (albeit with confidence widths that might grow to be vacuous).

In Figure 8, we fix $n = 30000, \beta = 0.05, \tau_1 = 0.06$, and plot the performance of GnC Gauss such that the guessed confidence width $\tau_{i+1} = \min(1.4\tau_i, 0.17)$ if the "check" for query ϕ_i results in a failure, otherwise $\tau_{i+1} = \tau_i$. For comparison, we also plot the actual maximum empirical error encountered by the answers provided by GnC ("GnC Gauss Emp"). It corresponds to the maximum empirical error of the answers of the Gaussian mechanism that is used as a guess mechanism within GnC, unless the check for a query results in a failure (which occurs 4 times in 40000 queries), in which case the error corresponds to the discretized answer on the holdout. We see that the statistically valid accuracy guaranteed by GnC is "responsive" to the empirical error of the realized answers produced by the GnC, and is almost always within a factor of 2 of the actual error.

Discussion: The runtime of our GnC system is dominated by the runtime of the mechanism providing the guesses. For each guess, the GnC system need only compute the empirical answer of the query on the holdout set, and a width (for example, from Lemma 3.1) that comes from a simple one-dimensional optimization. Thus, GnC with any particular Guess mechanism will have an execution time comparable to that of the Guess mechanism by itself. It is also important to note that GnC can be combined with any guess-generating mechanism, and it will inherit the worst-case generalization behavior of that mechanism. However, the GnC will typically provide much tighter confidence bounds (since the worst-case bounds are typically loose).

4 Conclusion

In this work, we focus on algorithms that provide explicit confidence intervals with *sound* coverage probabilities for adaptively posed statistical queries. We start by deriving tighter worst-case bounds for several mechanisms, and show that our improved bounds are within small constant factors of optimal for certain mechanisms. Our main contribution is the Guess and Check framework, that allows an analyst to use *any method* for "guessing" point estimates and confidence interval widths for their adaptive queries, and then rigorously validate those guesses on an additional holdout dataset. Our empirical evaluation demonstrates that GnC can improve on worst-case bounds by orders of magnitude, and that it improves on the naive baseline even for modest sample sizes. We also provide a Python library (Rogers et al. [2019]) implementing our GnC method.

Acknowledgements

The authors would like to thank Omer Tamuz for helpful comments regarding a conjecture that existed in a prior version of this work. A.R. acknowledges support in part by a grant from the Sloan Foundation, and NSF grants AF-1763314 and CNS-1253345. A.S. and O.T. were supported in part by a grant from the Sloan foundation, and NSF grants IIS-1832766 and AF-1763786. B.W. is supported by the NSF GRFP (award No. 1754881). This work was done in part while R.R., A.S., and O.T. were visiting the Simons Institute for the Theory of Computing.

References

Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1046–1059. ACM, 2016.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In Martin Hirt and Adam Smith, editors, *Theory of Cryptography*, pages 635–658, Berlin, Heidelberg, 2016a. Springer Berlin Heidelberg. ISBN 978-3-662-53641-4.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. *CoRR*, abs/1605.02065, 2016b. URL http://arxiv.org/abs/1605.02065.

Mark Bun, Jonathan Ullman, and Salil P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *STOC*, pages 1–10. ACM, May 31 – June 3 2014.

Cynthia Dwork and Guy N. Rothblum. Concentrated differential privacy. *CoRR*, abs/1603.01887, 2016.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toni Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems*, pages 2350–2358, 2015a.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248): 636–638, 2015b. doi: 10.1126/science.aaa9375. URL http://www.sciencemag.org/content/349/6248/636.abstract.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 117–126. ACM, 2015c.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, STOC '15, pages 117–126, New York, NY, USA, 2015d. ACM. ISBN 978-1-4503-3536-2. doi: 10.1145/2746539.2746580.

Vitaly Feldman and Thomas Steinke. Generalization for adaptively-chosen estimators via stable median. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, pages 728–757, 2017a. URL http://proceedings.mlr.press/v65/feldman17a.html.

Vitaly Feldman and Thomas Steinke. Calibrating noise to variance in adaptive data analysis. *CoRR*, abs/1712.07196, 2017b. URL http://arxiv.org/abs/1712.07196.

Andrew Gelman and Eric Loken. The statistical crisis in science. *American Scientist*, 102(6):460, 2014.

Alexej Gossmann, Aria Pezeshk, and Berkman Sahiner. Test data reuse for evaluation of adaptive machine learning algorithms: over-fitting to a fixed'test'dataset and a potential solution. In *Medical Imaging 2018: Image Perception, Observer Performance, and Technology Assessment*, volume 10577, page 105770K. International Society for Optics and Photonics, 2018.

Moritz Hardt and Jonathan Ullman. Preventing false discovery in interactive data analysis is hard. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 454–463. IEEE, 2014.

Christopher Jung, Katrina Ligett, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Moshe Shenfeld. A new analysis of differential privacy's generalization guarantees, 2019.

Horia Mania, John Miller, Ludwig Schmidt, Moritz Hardt, and Benjamin Recht. Model similarity mitigates test set overuse. *arXiv preprint arXiv:1905.12580*, 2019.

Ryan Rogers, Aaron Roth, Adam Smith, and Om Thakkar. Max-information, differential privacy, and post-selection hypothesis testing. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, 2016.

Ryan Rogers, Aaron Roth, Adam Smith, Nathan Srebro, Om Thakkar, and Blake Woodworth. Repository for empirical adaptive data analysis. https://github.com/omthkkr/empirical_adaptive_data_analysis, 2019.

Daniel Russo and James Zou. Controlling bias in adaptive data analysis using information theory. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS*, 2016.

Thomas Steinke and Jonathan Ullman. Interactive fingerprinting codes and the hardness of preventing false discovery. In *Proceedings of The 28th Conference on Learning Theory*, pages 1588–1628, 2015.

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In *NIPS 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 2521–2530, 2017.

Tijana Zrnic and Moritz Hardt. Natural analysts in adaptive data analysis. *arXiv preprint arXiv:1901.11143*, 2019.