DOI:10.1145/3051088

Guilt-Free Data Reuse

By Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth

Abstract

Existing approaches to ensuring the validity of inferences drawn from data assume a fixed procedure to be performed, selected before the data are examined. Yet the practice of data analysis is an intrinsically interactive and adaptive process: new analyses and hypotheses are proposed after seeing the results of previous ones, parameters are tuned on the basis of obtained results, and datasets are shared and reused.

In this work, we initiate a principled study of how to guarantee the validity of statistical inference in adaptive data analysis. We demonstrate new approaches for addressing the challenges of adaptivity that are based on techniques developed in privacy-preserving data analysis.

As an application of our techniques we give a simple and practical method for reusing a holdout (or testing) set to validate the accuracy of hypotheses produced adaptively by a learning algorithm operating on a training set.

1. INTRODUCTION: THE PROBLEM AND WHY IT IS IMPORTANT

From discovering new particles and clinical studies to election results prediction and credit score evaluation, scientific research, and industrial applications rely heavily on statistical data analysis. The goal of statistical data analysis is to enable an analyst to discover the properties of a process or phenomenon by analyzing data samples generated by the process. Fortunately, data samples reflect many properties of the process that generated them: if smoking increases the risk of lung cancer, then we should expect to see a correlation between smoking and lung cancer in samples of medical records. However, data will also exhibit idiosyncrasies that result from the randomness in the process of data sampling and do not say anything about the process that generated them—these idiosyncrasies will disappear if we resample new data from the process. Teasing out the true properties of the process from these idiosyncrasies is notoriously hard and error-prone task. Problems stemming from such errors can be very costly and have contributed to a wider concern about the reproducibility of research findings, most notably in medical research.[18]

Statisticians have long established a number of ways to measure the confidence in a result of analysis, most famously *p*-values and confidence intervals. These concepts allow the analyst to express the probability (over random sampling) that the outcome of an analysis is just an idiosyncrasy that does not hold for the true distribution of the data. Accordingly, the results can be declared statistically significant when this probability is sufficiently small. The guarantees that a confidence interval or *p*-value provide have a critical caveat; however, they apply only if the analysis procedure was chosen without examining the data to which the procedure is applied.

A simple and well-recognized misuse of this guarantee happens when an analyst performs multiple analyses but reports only the most favorable result (e.g., having the lowest *p*-value). It is known by many names including the multiple comparisons problem, multiple testing, *p*-hacking, and data dredging. As a result of such cherry picking, the reported analysis depends on the data, its stated *p*-value is incorrect and conclusions often invalid. A number of techniques, most notably false discovery rate control,[3] have been developed to address multiple comparisons when the set of analyses to be performed is known before the data are gathered. At the same time the practice of data analysis goes well beyond picking the best outcome from a fixed collection of analyses. Data exploration inspires hypothesis generation; results from one test determine which analyses are performed next; one study on a large corpus determines the next study on the same corpus. In short, data analysis in practice is inherently an *adaptive* process.

While very useful, reusing data in adaptive analysis can greatly increase the risk of spurious discoveries. Adaptive choices in analysis can lead to an exponential growth in the number of procedures that would have been performed had the analyst received different data samples. In other words adapting the analysis to data results in an implicit and potentially very large multiple comparisons problem aptly referred to as the "garden of forking paths" by Gelman and Loken.[14]

Although not usually understood in these terms, "Freedman's paradox" is an elegant demonstration of the powerful effect of adaptive analysis on the validity of conclusions.[13] In Freedman's simulation an equation is fitted, variables with an insignificant *t*-statistic are dropped and the equation is refit to this new—adaptively chosen—subset of variables, with famously misleading results: when the relationship between the dependent and explanatory variables is nonexistent, the procedure overfits, and erroneously declaring significant relationships. An excellent and interactive demonstration of variable selection on the results of linear regression analysis can be found in the online article of Aschwanden.[1]

While our previous discussion was concerned primarily with applications of statistics; adaptive data analysis presents a similar challenge in machine learning. An

This overview covers materials from two papers by the authors: "Preserving Statistical Validity in Adaptive Data Analysis," which appeared in *ACM Symposium on Theory of Computing (STOC) 2015*, and "Generalization in Adaptive Data Analysis and Holdout Reuse," which appeared in *Conference on Neural Information Processing Systems (NIPS) 2015*.

important goal in machine learning is to obtain a predictive model that generalizes well, that is, a model whose accuracy on the data is representative of its accuracy on future data generated by the same process. Indeed, a large body of theoretical and empirical research was developed for ensuring generalization in a variety of settings. In theoretical work, it is commonly assumed that the learning algorithm operates on a freshly sampled dataset. In practice, instead, a dataset is split randomly into two (or sometimes more) parts: the training set and the testing, or holdout, set. The training set is used for learning a predictor and the holdout set is used to estimate the true accuracy of the predictor. Because the predictor is independent of the holdout dataset, such an estimate is a valid estimate of the true prediction accuracy.

However, in practice the holdout dataset is rarely used only once. One prominent example in which a holdout set is often adaptively reused is hyperparameter tuning. Similarly, the holdout set in a machine learning competition, such as the famous ImageNet competition, is typically reused many times adaptively. Other examples include using the holdout set for variable selection, generation of base learners (in aggregation techniques, such as boosting and bagging), checking a stopping condition, and analyst-in-the-loop decisions. Such reuse is known to lead to overfitting to the holdout set (e.g., Refs.[7, 22]).

The literature recognizes the risks and proposes solutions in a number of special cases of adaptive data analysis. Most of them address a single round of adaptivity such as variable selection followed by regression on selected variables or model selection followed by testing and are optimized for specific inference procedures (see Chapter 7 in Ref.[17] for an overview). Yet, to our knowledge, there is no prior work giving a general methodology for addressing the risks of adaptive data reuse over many rounds of adaptivity and without restricting the type of procedures that are performed. We describe such a methodology, based on techniques from privacy-preserving data analysis, together with a concrete application we call the *reusable holdout*.

2. OUR APPROACH AND RESULTS[a]

Let us establish some simple terminology. We represent a data point as an element of some universe $\mathcal{X}$ and a dataset consists of n data points. A data generating process gives rise to a probability distribution over datasets. We will focus on the most commonly studied setting in which each point of the dataset is drawn randomly and independently from some unknown distribution $\mathcal{P}$ over $\mathcal{X}$. For example, the dataset may contain the health information and habits of n individuals, and the analyst is trying to learn about medical conditions affecting the population from which the individuals were drawn randomly.

We view adaptive analysis as a process in which an analyst wishes to ask a sequence of *queries* on a given dataset. Here a query could refer to an execution of some statistical procedure, a learning algorithm, preprocessing step, or any other inspection of the data. Crucially, after asking the first t queries, the analyst can use the results of those queries to pick the query performed at step $t + 1$. While our approach can be applied to a very general definition of queries, for simplicity we first focus on queries that estimate the mean of a function ϕ: $\mathcal{X} \to [0, 1]$ on the unknown distribution $\mathcal{P}$ or $\mathcal{P}[\phi] = \mathbf{E}_{x\sim\mathcal{P}}[\phi(x)]$. The estimate is required to be correct up to some additive error τ usually referred to as *tolerance* with high probability. Such queries allow the analyst to learn a variety of basic statistics of the population, for example, the fraction of the population over six feet all. More generally, they allow the analyst to estimate the true means and moments of individual attributes, correlations between attributes and the accuracy of any predictive model. Such queries are referred to as *statistical* in the context of the well-studied statistical query model[19] and have also been studied in statistics as *linear statistical functionals*. It is known that many standard data analyses can be performed using access to statistical queries instead of direct access to data (see Refs.[4, 19] for examples).

Even in this relatively simple setting the question of how many adaptively chosen statistical queries can be correctly answered using n samples drawn i.i.d. from $\mathcal{P}$ has not been previously examined. The conservative approach of using fresh samples for each adaptively chosen query requires n to scale linearly with the number of queries m. We observe that such a bad dependence is inherent in the standard approach of estimating expectations by the exact empirical average on the samples. This is directly implied by "Freedman's paradox"[13] and we describe an additional simple example in Ref.[9] This situation is in stark contrast to the nonadaptive case in which $n = O\left(\frac{\log m}{\tau^2}\right)$ samples suffice to answer m queries with a tolerance τ using empirical averages.

We demonstrate that the problem can be addressed using techniques developed in the context of *differential privacy*, a definition of privacy tailored to privacy-preserving data analysis. Roughly speaking, differential privacy ensures that the probability of observing any outcome from an analysis is "essentially unchanged" by modifying any single dataset element (the probability distribution is over the randomness introduced by the algorithm).

The central insight of the differentially private data analysis is that it is possible to learn statistical properties of a dataset, whereas controlling the amount of information revealed about any dataset element. Our approach is based on the same view of the adaptive data reuse problem: the analyst can be prevented from overfitting to the data if the amount of information about the data revealed to the analyst is limited. To ensure that information leakage is limited, the algorithm needs to control the access of the analyst to the data. We show that this view can be made formal by introducing the notion of maximum information between two random variables. This notion allows us to bound the factor by which uncertainty about the dataset is reduced given the output of the algorithm on this dataset. We describe it in more detail in Section 3.1.

[a] Additional averaging over k different partitions is used in k-fold cross-validation.

Our main technical result is a broad *transfer theorem* showing that any analysis that is carried out in a differentially private manner must lead to a conclusion that generalizes to the underlying distribution. This theorem allows us to draw on a rich body of results in differential privacy and to obtain corresponding results for our problem of guaranteeing generalization in adaptive data analysis. We describe this general theorem in detail in Section 3.

A direct corollary of our theorem is that, remarkably, it is possible to answer nearly *exponentially many* adaptively chosen statistical queries (in the size of the data set n). Equivalently, this reduces the *sample complexity* of answering m queries from *linear* in the number of queries to *polylogarithmic*, nearly matching the dependence that is necessary for nonadaptively chosen queries.

THEOREM 1. *There exists an algorithm that given a dataset of size at least $n \geq n_0$, can answer any m adaptively chosen statistical queries so that with high probability, each answer is correct up to tolerance τ, where*

$$n_0 = O\left(\frac{(\log m)^{3/2}\sqrt{\log|\mathcal{X}|}}{\tau^{7/2}}\right).$$

In this bound $\log|\mathcal{X}|$ should be viewed as roughly the *dimension* of the domain. For example, if the underlying domain is $\mathcal{X} = \{0, 1\}^d$, the set of all possible vectors of d-boolean attributes, then $\log|\mathcal{X}| = d$.

Unfortunately, this algorithm for answering queries is not computationally efficient (it has running time linear in the size of the data universe $|\mathcal{X}|$, which is *exponential* in the dimension of the data). Still, we show that it is possible to quadratically improve on the naïve empirical-mean-based approach by using a simple and practical algorithm that perturbs the answer to each query with independent noise.

THEOREM 2. *There exists a computationally efficient algorithm for answering m adaptively chosen statistical queries, such that with high probability, the answers are correct up to tolerance τ, given a data set of size at least $n \geq n_0$ for:*

$$n_0 = O\left(\frac{\sqrt{m}(\log m)^{3/2}}{\tau^{5/2}}\right).$$

A natural question raised by our results is whether there is an efficient algorithm that can answer an exponential number of adaptively chosen queries. This question was addressed in Refs.[16, 25] who show that under standard cryptographic assumptions no algorithm can improve on the upper bound achieved by our simple algorithm: any algorithm that can answer more than $\approx n^2$ adaptively chosen statistical queries must have running time exponential in $\log|\mathcal{X}|$.

This lower bound implies that practical algorithms that can answer an exponential number of arbitrarily and adaptively chosen queries are unlikely to exist. Yet we show that there is an alternative way to apply our techniques to answer an exponentially large number of queries efficiently. In this application, the analyst splits the dataset into a training set and a holdout set. The analyst can then perform any analysis on the training dataset, but can only access the holdout set via queries to our *reusable holdout* algorithm. The reusable holdout algorithm allows the analyst to validate her models and statistics against the holdout set. More formally, the analyst can pick any function $\phi : \mathcal{X} \to [0, 1]$. If the empirical mean of ϕ evaluated on the training set is close to the true expectation $\mathcal{P}[\phi]$, in other words ϕ does not overfit to the training set, then the reusable holdout confirms that there is no overfitting (but provides no additional information). Otherwise, the algorithm returns an estimate of $\mathcal{P}[\phi]$ that answers the statistical query for ϕ.

We describe a specific instantiation of reusable holdout referred to as Thresholdout. The number of queries that Thresholdout can answer is exponential in the size of the holdout set n as long as the number of times that the analyst overfits (to the training set) is at most quadratic in n. The analysis of Thresholdout is based on known techniques in differential privacy and our transfer theorem. In Section 4, we describe Thresholdout and its guarantees in detail. We then illustrate the properties of Thresholdout using a simple classification algorithm on synthetic data. The classifier produced by the algorithm overfits the data when the holdout set is reused in the standard way, but does not overfit if used with our reusable holdout.

In Ref.,[10] we describe additional algorithms for validating results of adaptive queries against the holdout that are based on description length. Our application of this simple and classical technique differs from its standard uses to derive generalization. It leads to algorithms with guarantees that are incomparable to those achieved via differential privacy.

2.1. Related work

The classical approach in theoretical machine learning to ensure that empirical estimates generalize to the underlying distribution is based on the various notions of complexity of the set of functions output by the algorithm, most notably the Vapnik–Chervonenkis (VC) dimension (see Ref.[23] for a textbook introduction). If one has a sample of data large enough to guarantee generalization for all functions in some class of bounded complexity, then it does not matter whether the data analyst chooses functions in this class adaptively or nonadaptively. Our goal, in contrast, is to prove generalization bounds *without* making any assumptions about the class from which the analyst can choose query functions. In this case the adaptive setting is very different from the nonadaptive setting.

An important and related line of work[6, 20, 24] establishes connections between the *stability* of a learning algorithm and its ability to generalize. Stability is a measure of how much the error of a function output by a learning algorithm is perturbed by the changes to its input dataset. It is known that certain stability notions are necessary and sufficient for generalization.[24] Unfortunately, the stability notions considered in these prior works do not compose in the sense that running multiple stable algorithms sequentially and adaptively may result in a procedure that is not stable. Differential privacy is stronger than these previously studied notions of stability, and in particular enjoys strong composition guarantees. This provides a calculus for building up complex

algorithms that satisfy stability guarantees sufficient to give generalization. Our work can thus be interpreted as showing that differential privacy plays the role of stability in the multistep adaptive analysis setting.

There is a very large body of work designing differentially private algorithms for various data analysis tasks, some of which we leverage in our applications (see Ref.[8] for a short survey and Ref.[12] for a textbook introduction to differential privacy).

For differentially private algorithms that output a hypothesis it has been known as folklore that differential privacy implies stability of the hypothesis to replacing (or removing) an element of the input dataset. Such stability is long known to imply generalization *in expectation* (e.g., Ref.[24]). Our technique can be seen as a substantial strengthening of this observation: from expectation to high probability bounds (which is crucial for answering many queries), from pure to approximate differential privacy (which is crucial for our improved efficient algorithms), and beyond the expected error of a hypothesis.

Building on our work, Blum and Hardt[5] showed how to reuse the holdout set to maintain an accurate leaderboard in a machine learning competition that allows the participants to submit adaptively chosen models in the process of the competition (such as those organized by Kaggle Inc.).

Finally, in a recent follow-up work, Bassily et al.[2] strengthen the link between generalization and approximate differential privacy quantitatively and extend it to the more general class of low-sensitivity queries. Their result leads to bounds on the number of samples that are needed to guarantee generalization that improve on our theorems by a factor of $O(\sqrt{\log(m)/\tau})$.

3. DIFFERENTIAL PRIVACY AND GENERALIZATION

Our results rely on a strong connection we make between differential privacy and generalization. At a high level, we prove that if $\mathcal{M}$ is a differentially private algorithm then the empirical average of a function that it outputs on a random dataset will be close to the true expectation of the function with high probability over the choice of the dataset and the randomness of $\mathcal{M}$. More formally, for a dataset $S = (x_1, \ldots, x_n)$ and a function $\phi : \mathcal{X} \to [0, 1]$, let $\mathcal{E}_S[\phi] = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)$ denote the empirical average of ϕ. We denote a random dataset chosen from $\mathcal{P}^n$ by $\boldsymbol{S}$. Standard Chernoff–Hoeffding concentration inequalities for sums of independent random variables imply that for any fixed function ϕ, the empirical average $\mathcal{E}_S[\phi]$ is strongly concentrated around its expectation $\mathcal{P}[\phi]$. However, this statement is no longer true if ϕ is allowed to depend on $\boldsymbol{S}$ (i.e., what happens if we choose functions adaptively, using previous estimates on $\boldsymbol{S}$). However, for a hypothesis output by a differentially private $\mathcal{M}$ on $\boldsymbol{S}$ (denoted by $\phi = \mathcal{M}(\boldsymbol{S})$), we show that $\mathcal{E}_S[\phi]$ is close to $\mathcal{P}[\phi]$ with high probability. Before making our statements formal we review the definition of differential privacy.[11]

Definition 3. *A randomized algorithm $\mathcal{M}$ with domain $\mathcal{X}^n$ is (ε, δ)-differentially private if for all $\mathcal{O} \subseteq$ Range$(\mathcal{M})$ and for all pairs of datasets $S, S' \in \mathcal{X}^n$ that differ in a single element:*

$$\Pr\left[\mathcal{M}(S) \in \mathcal{O}\right] \leq e^{\varepsilon} \cdot \Pr\left[\mathcal{M}(S') \in \mathcal{O}\right] + \delta,$$

where the probability space is over the coin flips of the algorithm $\mathcal{M}$. The case when $\delta = 0$ is sometimes referred to as pure *differential privacy, and in this case we may say simply that $\mathcal{M}$ is ε-differentially private.*

The concentration bounds we obtain for pure differential privacy are almost as strong as those given by the standard Chernoff–Hoeffding concentration inequalities.

Theorem 4. *Let $\mathcal{M}$ be an ε-differentially private algorithm that outputs a function from $\mathcal{X}$ to* $[0, 1]$. *For a random variable $\boldsymbol{S}$ distributed according to $\mathcal{P}^n$ we let $\phi = \mathcal{M}(\boldsymbol{S})$. Then*

$$\Pr\left[\left|\mathcal{P}[\phi] - \mathcal{E}_S[\phi]\right| > \varepsilon\right] \leq 6 \cdot e^{-\varepsilon^2 n}.$$

This statement also holds more broadly for an important class of low sensitivity functions. These are functions of a dataset that for some *sensitivity* Δ satisfy: $|f(S) - f(S')| \leq \Delta$ for all datasets $S, S' \in \mathcal{X}^n$ that differ in a single element. Note that the sensitivity of the empirical average of any function with range $[0, 1]$ on a dataset of size n is at most $1/n$.

We outline the proof idea of this result in Section 3.1. A similar concentration result can also be obtained for (ε, δ)-differentially private algorithms although it is not quite as strong and requires a substantially different and more involved proof. Our result for this case (see Ref.[9]) has been recently strengthened and generalized to low-sensitivity queries using a new proof technique by Bassily et al.[2]

Theorem 4 implies that $|\mathcal{P}[\phi] - \mathcal{E}_S[\phi]| \leq \tau$ holds with high probability whenever ϕ is generated by a differentially private algorithm $\mathcal{M}$. This might appear to be different from what we need in our application because there the queries are generated by an arbitrary (possibly adversarial) adaptive analyst and we only have control over the query answering algorithm. The connection comes from a crucial property of differential privacy, known as its post-processing guarantee: *Any* algorithm that can be described as the (possibly randomized) post-processing of the output of a differentially private algorithm is itself differentially private (e.g., Ref.[12]). Hence, although we do not know how an arbitrary analyst is adaptively generating her queries, we do know that if the only access she has to $\boldsymbol{S}$ is through a differentially private algorithm, then her method of producing query functions must be differentially private with respect to $\boldsymbol{S}$. We can therefore, without loss of generality, think of the query answering algorithm and the analyst as a *single* algorithm $\mathcal{M}$, that is, given a random data set $\boldsymbol{S}$ and returns a differentially private output query $\phi = \mathcal{M}(\boldsymbol{S})$.

We also note that the bound in Theorem 4 gives the probability of correctness for each individual answer to a query, meaning that the error probability is for each query and not for all queries at the same time. The bounds we state in Theorems 1 and 2 hold with high probability for all m queries and to obtain them from the bounds in this section, we apply the union bound.

All we are missing now to get an algorithm for answering adaptively chosen statistical queries is an algorithm that satisfies the following two conditions:

1. The algorithm can answer every query ϕ with a value, that is, close (up to error α) to the empirical average of ϕ on the dataset.
2. The algorithm is differentially private.

The problem of providing accurate answers to a large number of queries for the average value of a function on the dataset has been the subject of intense investigation in the differential privacy literature. Such queries are usually referred to as (fractional) *counting queries* or *linear queries* in this context. This allows us to obtain statistical query answering algorithms by using various known differentially private algorithms for answering counting queries. Specifically, our Theorem 1 relies on the algorithm in Ref.[15] that uses the multiplicative weights update algorithm to answer the queries. Our Theorem 2 relies on the basic Laplace noise mechanism and strong composition properties of differential.

In the resulting algorithm, α should be viewed as bounding the empirical error, ϵ should be viewed as bounding the generalization error and $\tau = \alpha + \epsilon$ as bounding the total error. Notice that the standard approach of using empirical averages has the optimal empirical error—it has $\alpha = 0$. However, it is not ϵ-differentially private for any ϵ and, as we pointed out earlier, does not provide any guarantee on the generalization error. At the opposite end, an algorithm which answers queries with a constant, independent of the data, has optimal generalization error, but horrible empirical error. Differentially private algorithms for answering counting queries allow us to explicitly trade off empirical error α with generalization error ϵ to obtain a strong bound on the total error τ.

3.1. Max-information

Intuitively, one way to ensure that the function output by an algorithm $\mathcal{M}$ generalizes is to guarantee that the function does depend too much on the input dataset S. We demonstrate that this intuition can be captured via the notion of *max-information* that we introduce.

Definition 5. *Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be jointly distributed random variables. The max-information between $\boldsymbol{X}$ and $\boldsymbol{Y}$, denoted $I_\infty(\boldsymbol{X};\boldsymbol{Y})$, is the minimal value of k such that for every x in the support of $\boldsymbol{X}$ and y in the support of $\boldsymbol{Y}$ we have* $\Pr[\boldsymbol{X} = x \mid \boldsymbol{Y} = y] \le 2^k \cdot \Pr[\boldsymbol{X} = x]$.

It follows immediately from Bayes' rule that $I_\infty(\boldsymbol{X};\boldsymbol{Y}) = I_\infty(\boldsymbol{Y};\boldsymbol{X})$. In our use $(\boldsymbol{X}, \boldsymbol{Y})$ is going to be a joint distribution $(\boldsymbol{S}, \phi)$ on (dataset, function) pairs. The dataset $\boldsymbol{S}$ is drawn from distribution $\mathcal{P}^n$ that corresponds to n points drawn i.i.d. from $\mathcal{P}$. Random variable ϕ represents the function generated by the analyst, whereas interacting with $\boldsymbol{S}$ through our mechanism. Importantly, the analyst may arrive at the function after observing the evaluations of other functions on the same dataset $\boldsymbol{S}$. Now with each possible function ϕ in the support of $\boldsymbol{\phi}$ we associate a set of "bad" datasets $R(\phi)$. We later choose $R(\phi)$ to mean the empirical value $\mathcal{E}_S[\phi]$ is far from the true value $\mathcal{P}[\phi]$, that is, ϕ overfits to S. Maximum information gives a bound on the probability that $\boldsymbol{S}$ falls in $R(\boldsymbol{\phi})$.

Theorem 6. *For $k = I_\infty(\boldsymbol{S};\boldsymbol{\phi})$, $\Pr[\boldsymbol{S} \in R(\boldsymbol{\phi})] \le 2^k \cdot \max_\phi \Pr[\boldsymbol{S} \in R(\phi)]$.*

The proof follows easily by first decomposing the event $\boldsymbol{S} \in R(\boldsymbol{\phi})$ into events, $\boldsymbol{S} \in R(\phi)$ & $\boldsymbol{\phi} = \phi$ for all ϕ. Namely,

$$\Pr\left[\boldsymbol{S} \in R(\boldsymbol{\phi})\right] = \sum_\phi \Pr\left[\boldsymbol{S} \in R(\phi) \;\&\; \boldsymbol{\phi} = \phi\right].$$

Since

$$\Pr\left[\boldsymbol{S} \in R(\phi) \;\&\; \boldsymbol{\phi} = \phi\right] = \Pr\left[\boldsymbol{S} \in R(\phi) \mid \boldsymbol{\phi} = \phi\right] \cdot \Pr\left[\boldsymbol{\phi} = \phi\right],$$

we can apply the definition of max-information and obtain that $\Pr[\boldsymbol{S} \in R(\phi) \mid \boldsymbol{\phi} = \phi] \le 2^k \Pr[\boldsymbol{S} \in R(\phi)]$. Substituting this bound back into the decomposition gives the desired result:

$$\begin{aligned}\Pr\left[\boldsymbol{S} \in R(\boldsymbol{\phi})\right] &\le \sum_\phi 2^k \cdot \Pr\left[\boldsymbol{S} \in R(\phi)\right] \cdot \Pr\left[\boldsymbol{\phi} = \phi\right] \\ &\le 2^k \cdot \max_\phi \Pr\left[\boldsymbol{S} \in R(\phi)\right].\end{aligned}$$

Our theorem is completely general in the sense that the random variable $\boldsymbol{\phi}$ does not have to be supported on functions over $\mathcal{X}$ and could instead assume values in any other discrete domain. For example, such output could be a set of features of the data to be used for a subsequent supervised learning task. For our main application ϕ refers to a function, and we denote the set of datasets on which the empirical estimator has error greater than τ as

$$R_\tau(\phi) = \left\{S \in \mathcal{X}^n : \mathcal{E}_S[\phi] - \mathcal{P}[\phi] > \tau\right\}. \tag{1}$$

By Hoeffding's bound we know that $\max_\phi \Pr[\boldsymbol{S} \in R_\tau(\phi)] \le \exp(-2\tau^2 n)$. This gives the following immediate corollary.

Corollary 7. *If $I_\infty(\boldsymbol{S};\boldsymbol{\phi}) \le \log_2 e \cdot \tau^2 n$, then $\Pr[\boldsymbol{S} \in R_\tau(\boldsymbol{\phi})] \le \exp(-\tau^2 n)$.*

To apply this corollary all we need is to show that pure differential privacy implies a sufficiently strong bound on max information $I_\infty(\boldsymbol{S};\boldsymbol{\phi})$.

Theorem 8. *Let $\mathcal{M}$ be an ϵ-differentially private algorithm. Let $\boldsymbol{S}$ be any random variable over n-element input datasets for $\mathcal{M}$ and let $\boldsymbol{Y}$ be the corresponding output distribution $\boldsymbol{Y} = \mathcal{M}(\boldsymbol{S})$. Then $I_\infty(\boldsymbol{S};\boldsymbol{Y}) \le \log_2 e \cdot \epsilon n$.*

The proof of this theorem follows from observing that, any two datasets S and S' differ in at most n elements. Therefore, applying the guarantee of differential privacy n times, we obtain that for every y,

$$\Pr[\boldsymbol{Y} = y \mid \boldsymbol{S} = S] \le e^{\epsilon n} \Pr[\boldsymbol{Y} = y \mid \boldsymbol{S} = S'].$$

As there must exist a dataset y such that $\Pr[\boldsymbol{Y} = y \mid \boldsymbol{S} = S'] \le \Pr[\boldsymbol{Y} = y]$ we can conclude that for every S and every y it holds that $\Pr[\boldsymbol{Y} = y \mid \boldsymbol{S} = S] \le e^{\epsilon n} \Pr[\boldsymbol{Y} = y]$. This yields the desired bound $I_\infty(\boldsymbol{S};\boldsymbol{Y}) = I_\infty(\boldsymbol{Y};\boldsymbol{S}) \le \log_2 e \cdot \epsilon n$.

From Theorem 8 and Corollary 7, we see that ensuring τ^2-differential privacy over the entire interaction with the dataset strictly controls the probability that the adversary can choose a function that overfits to the dataset. This is somewhat worse than the claim in Theorem 4 which requires τ-differential privacy. In Ref.,[10] we show that by considering a simple relaxation of max-information, referred to as approximate max-information, it is possible to prove the stronger bound on max-information of differentially private algorithms for datasets consisting of i.i.d. samples. Interestingly, it is not hard to show that algorithms whose output has short description length (in bits) also have low approximate max-information. Thus approximate max-information unifies generalization bounds obtained via (pure) differential privacy and description length. In addition, composition properties of approximate max-information imply that one can easily obtain generalization guarantees for adaptive sequences of algorithms, some of which are differentially private, and others of which have outputs with short description length.

4. THE REUSABLE HOLDOUT

In this section, we describe a practical application of our framework, which gives a method for safely reusing a holdout set many times. In this application, the analyst splits the dataset into a training set and a holdout set. An advantage of this approach is that the data analyst will have full, unrestricted access to the training set and can use it in any way that she desires. The holdout set can only be accessed through a reusable holdout algorithm. The goal of this algorithm is to validate the results of analyses performed on the training set.

We describe a specific instantiation of reusable holdout, referred to as Thresholdout, that validates the values of statistical queries and is based on the "Sparse Vector" technique from differential privacy (e.g., Chapter 3 of Ref.[12]). Specifically, for every function $\phi: \mathcal{X} \to [0, 1]$ given by the analyst, the algorithm checks if the empirical average of ϕ on the training set is close to the true mean of ϕ (up to some tolerance τ). If the values are close the algorithm does not provide any additional information to the analyst. Only if ϕ overfits the training set does the algorithm provide a valid estimate of the true expectation of ϕ. The result is that for all of the queries that the analyst asks, she has correct estimates of the true expectation—either our algorithm certifies that the estimate from the training set is approximately correct or else it provides a correct estimate using the holdout set. The analysis of the algorithm shows that the number of samples needed by Thresholdout depends only logarithmically on the total number of queries asked by the data analyst as long as the total number of queries that overfit the training set (and have to be answered using the holdout set) is not too large. As a result, this simple and computationally efficient algorithm can potentially answer an exponential (in n) number of queries.

More formally, Thresholdout is given access to the training dataset S_t and holdout dataset S_h and a budget limit B. It allows any query of the form $\phi: \mathcal{X} \to [0, 1]$ and its goal is to provide an estimate of $\mathcal{P}[\phi]$. To achieve this the algorithm gives an estimate of $\mathcal{E}_{S_h}[\phi]$ in a way that prevents overfitting of functions generated by the analyst to the holdout set. In other words, responses of Thresholdout are designed to ensure that, with high probability, $\mathcal{E}_{S_h}[\phi]$ is close to $\mathcal{P}[\phi]$ and hence an estimate of $\mathcal{E}_{S_h}[\phi]$ gives an estimate of the true expectation $\mathcal{P}[\phi]$. Given a function ϕ, Thresholdout first checks if the difference between the average value of ϕ on the training set S_t (or $\mathcal{E}_{S_t}[\phi]$) and the average value of ϕ on the holdout set S_h (or $\mathcal{E}_{S_h}[\phi]$) is below a certain threshold $T + \eta$. Here, T is a fixed number such as 0.01 and η is a Laplace noise variable whose standard deviation needs to be chosen depending on the desired guarantees. If the difference is below the threshold, then the algorithm returns $\mathcal{E}_{S_t}[\phi]$. If the difference is above the threshold, then the algorithm returns $\mathcal{E}_{S_h}[\phi] + \xi$ for another Laplacian noise variable ξ. Each time the difference is above threshold the "overfitting" budget B is reduced by one. Once it is exhausted, Thresholdout stops answering queries. In Figure 1, we provide the pseudocode of Thresholdout.

We now state the formal generalization guarantees that the entire execution of Thresholdout enjoys. They are based on the privacy guarantees of the "Sparse Vector" technique given in Chapter 3 of Ref.[12] and the generalization properties of differential privacy. For pure differential privacy we rely on Theorem 4 and for (ε, δ)-differential privacy we use the bound in Ref.[21]

THEOREM 9. *Let $\beta, \tau > 0$ and $m \geq B > 0$. We set $T = 3\tau/4$ and $\sigma = \tau/(96 \ln(4m/\beta))$. Let $\mathbf{S}$ denote a holdout dataset of size n drawn i.i.d. from a distribution $\mathcal{P}$ and S_t be any additional dataset over $\mathcal{X}$. Consider an algorithm that is given access to S_t and adaptively chooses functions $\phi_1, \ldots, \phi_m$ while interacting with* Thresholdout *which is given datasets $\mathbf{S}$, S_t and values σ, B, T. For every $i \in [m]$, let $\mathbf{a}_i$ denote the answer of* Thresholdout *on function $\phi_i: \mathcal{X} \to [0, 1]$. Further, for every $i \in [m]$, we define the counter of overfitting*

$$\mathbf{Z}_i \doteq |\{j \leq i : |\mathcal{P}[\phi_j] - \mathcal{E}_{S_t}[\phi_j]| > \tau/2\}|.$$

Then

$$\Pr[\exists i \in [m], \mathbf{Z}_i < B \ \& \ |\mathbf{a}_i - \mathcal{P}[\phi_i]| \geq \tau] \leq \beta,$$

whenever $n \geq n_0$ for

$$n_0 = O\left(\frac{\ln(m/\beta)}{\tau^2}\right) \cdot \min\{B, \sqrt{B \ln(\ln(m/\beta)/\tau)}\}.$$

Note that in the bound on n, the term $O\left(\frac{\ln(m/\beta)}{\tau^2}\right)$ is equal (up to a constant factor) to the number of samples that are necessary to answer m *nonadaptively* chosen queries with tolerance τ and confidence $1 - \beta$. Further, this bound allows m to be exponentially large in n as long as B grows subquadratically in n (that is, $B \leq n^{2-c}$ for a constant $c > 0$).

We remark that the same approach also works for the class of low sensitivity queries. In Ref.,[10] we also give an incomparable version of this algorithm with guarantees that derive from description length arguments rather than from

differential privacy. The advantage of that variant is that its use is not limited to low sensitivity queries.

4.1. Illustrative experiments

We describe a simple experiment on synthetic data that illustrates the danger of reusing a standard holdout set and how this issue can be resolved by our reusable holdout. In our experiment the analyst wants to build a classifier via the following common strategy. First the analyst finds a set of single attributes that are correlated with the class label. Then the analyst aggregates the correlated variables into a single model of higher accuracy (e.g., using boosting or bagging methods). More formally, the analyst is given a d-dimensional labeled data set S of size $2n$ and splits it randomly into a training set S_t and a holdout set S_h of equal size. We denote an element of S by a tuple (x, y) where x is a d-dimensional vector and $y \in \{-1, 1\}$ is the corresponding class label. The analyst wishes to select variables to be included in her classifier. For various values of the number of variables to select k, she picks k variables with the largest absolute correlations with the label. However, she verifies the correlations (with the label) on the holdout set and uses only those variables whose correlation agrees in sign with the correlation on the training set and both correlations are larger than some threshold in absolute value. She then creates a simple linear threshold classifier on the selected variables using only the signs of the correlations of the selected variables. A final test evaluates the classification accuracy of the classifier on both the training set and the holdout set.

In the experiments, we used an implementation of Thresholdout that differs somewhat from the algorithm we analyzed theoretically (given in Figure 1). Specifically, we set the parameters to be $T = 0.04$ and $\sigma = 0.01$. This is lower than the values necessary for the proof (and which are not intended for direct application) but suffices to prevent overfitting in our experiment. Second, we used Gaussian noise instead of Laplacian noise as it has stronger concentration properties (in many differential privacy applications similar theoretical guarantees hold for mechanisms based on Gaussian noise—although not for ours).

No correlation between labels and data. In our first experiment, each attribute is drawn independently from the normal distribution $N(0, 1)$ and we choose the class label $y \in \{-1, 1\}$ uniformly at random so that there is no correlation between the data point and its label. We chose $n = 10{,}000$, $d = 10{,}000$ and varied the number of selected variables k. In this scenario no classifier can achieve true accuracy better than 50%. Nevertheless, reusing a standard holdout results in reported accuracy of over 63% for $k = 500$ on both the training set and the holdout set (the standard deviation of the error is less than 0.5%). The average and standard deviation of results obtained from 100 independent executions of the experiment are plotted in Figure 2 which also includes the accuracy of the classifier on another fresh data set of size n drawn from the same distribution. We then executed the same algorithm with our reusable holdout. The algorithm Thresholdout was invoked with $T = 0.04$ and $\sigma = 0.01$ explaining why the accuracy of the classifier reported by Thresholdout is off by up to 0.04 whenever the accuracy on the holdout set is within 0.04 of the accuracy on the training set. Thresholdout prevents the algorithm from overfitting

Figure 1. The details of `Thresholdout` algorithm.

Algorithm Thresholdout
Input: Training set S_t, holdout set S_h, noise rate σ, budget B, threshold T.
Set $\hat{T} \leftarrow T + \gamma$ for $\gamma \sim \text{Lap}(2{\cdot}\sigma)$
Query step: Given a function ϕ: $\mathcal{X} \rightarrow [0, 1]$, do:

1. If $B < 1$ output "$\perp$"
2. Else sample $\xi \sim \text{Lap}(\sigma)$, $\gamma \sim \text{Lap}(2{\cdot}\sigma)$, and $\eta \sim \text{Lap}(4{\cdot}\sigma)$
   (a) If $|\mathcal{E}_{S_h}[\phi] - \mathcal{E}_{S_t}[\phi]| > \hat{T} + \eta$, output $\mathcal{E}_{S_h}[\phi] + \xi$ and set $B \leftarrow B - 1$ and $\hat{T} \leftarrow T + \gamma$.
   (b) Otherwise, output $\mathcal{E}_{S_t}[\phi]$.

Figure 2. No correlation between class labels and data points. The plot shows the classification accuracy of the classifier on training, holdout, and fresh sets. Margins indicate the standard deviation.

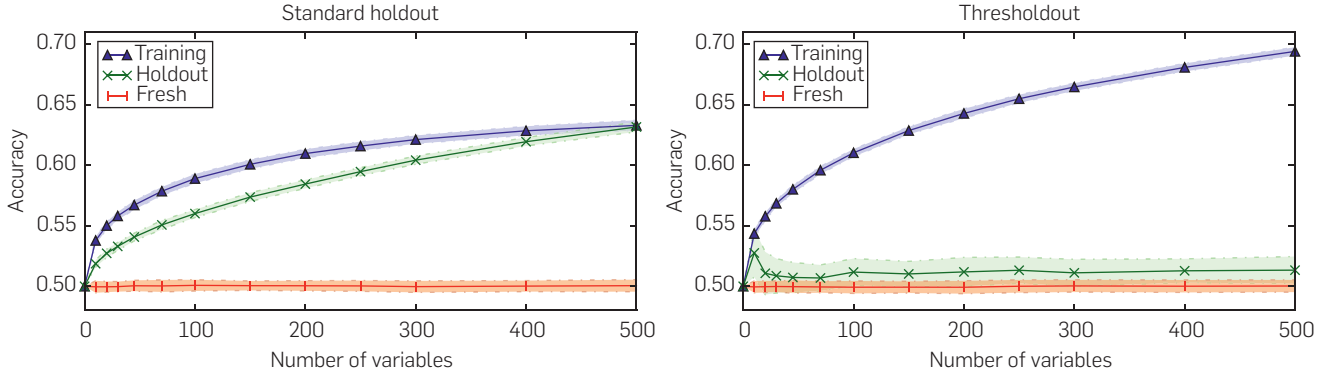
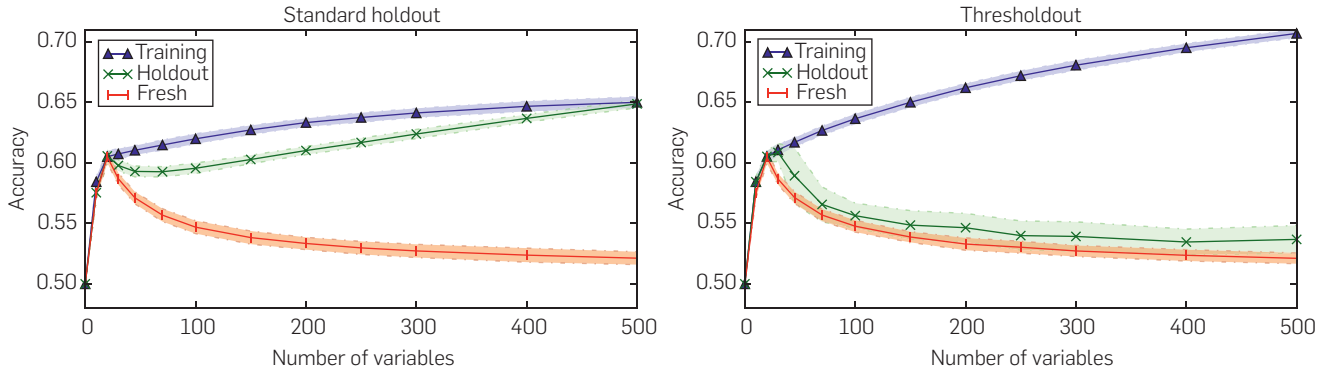


Figure 3. Some variables are correlated with the label.



to the holdout set and gives a valid estimate of classifier accuracy.

High correlation between labels and some of the variables. In our second experiment, the class labels are correlated with some of the variables. As before the label is randomly chosen from $\{-1, 1\}$ and each of the attributes is drawn from $N(0, 1)$ aside from 20 attributes which are drawn from $N(y \cdot 0.06, 1)$ where y is the class label. We execute the same algorithm on this data with both the standard holdout and Thresholdout and plot the results in Figure 3. Our experiment shows that when using the reusable holdout, the algorithm still finds a good classifier while preventing overfitting. This illustrates that the reusable holdout simultaneously prevents overfitting and allows for the discovery of true statistical patterns.

Acknowledgments

We would like to thank S. Arora, M.F. Balcan, A. Blum, D. Foster, M. Kearns, J. Kleinberg, A. Rakhlin, P. Rigollet, W. Su, and J. Ullman for the enlightening discussions about this work. We also thank the Simons Institute for Theoretical Computer Science at Berkeley where part of this research was done. This work was supported by the Alfred P. Sloan Foundation and NSF grant CNS 1253345.

References

1. Aschwanden, C. Science isn't broken.
2. Bassily, R., Nissim, K., Smith, A.D., Steinke, T., Stemmer, U., Ullman, J. Algorithmic stability for adaptive data analysis. In *STOC*, Cambridge, MA, USA (2016), 1046–1059.
3. Benjamini, Y., Hochberg, Y. Controlling the false discovery rate – A practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B 57* (1995), 289–300.
4. Blum, A., Dwork, C., McSherry, F., Nissim, K. Practical privacy: The SuLQ framework. In *PODS*, Baltimore, Maryland, USA (2005), 128–138.
5. Blum, A., Hardt, M. The ladder: A reliable leaderboard for machine learning competitions. In *ICML*, Lille, France (2015), 1006–1014.
6. Bousquet, O., Elisseeff, A. Stability and generalization. *JMLR 2* (2002), 499–526.
7. Cawley, G.C., Talbot. N.L.C. On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res. 11* (2010), 2079–2107.
8. Dwork, C. A firm foundation for private data analysis. *CACM 54*, 1 (2011), 86–95.
9. Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O., Roth, A. Preserving statistical validity in adaptive data analysis. *CoRR*, abs/1411.2664, 2014. Extended abstract in STOC 2015.
10. Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O., Roth, A. Generalization in adaptive data analysis and holdout reuse. *CoRR*, abs/1506, 2015. Extended abstract in NIPS 2015.
11. Dwork, C., McSherry, F., Nissim, K., Smith, A. Calibrating noise to sensitivity in private data analysis. In *TCC*, New York, NY, USA (2006), 265–284.
12. Dwork, C., Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci. 9*, 34 (2014), 211–407.
13. Freedman, D.A. A note on screening regression equations. *Am. Statist. 37*, 2 (1983), 152–155.
14. Gelman, A., Loken, E. The statistical crisis in science. *Am. Statist. 102*, 6 (2014), 460.
15. Hardt, M., Rothblum, G. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS*, Las Vegas, Nevada, USA (2010), 61–70.
16. Hardt, M., Ullman, J. Preventing false discovery in interactive data analysis is hard. In *FOCS*, Philadelphia, PA, USA (2014), 454–463.
17. Hastie, T., Tibshirani, R., Friedman, J.H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics*. Springer-Verlag, New York (2009).
18. Ioannidis, J.P.A. Why most published research findings are false. *PLoS Med. 2*, 8 (Aug. 2005), 124.
19. Kearns, M. Efficient noise-tolerant learning from statistical queries. *J. ACM 45*, 6 (1998), 983–1006.
20. Mukherjee, S., Niyogi, P., Poggio, T., Rifkin, R. Learning theory: Stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Adv. Comput. Math. 25*, 1–3 (2006), 161–193.
21. Nissim, K., Stemmer, U. On the generalization properties of differential privacy. *CoRR* (2015), abs/1504.05800.
22. Reunanen, J. Overfitting in making comparisons between variable selection methods. *J. Mach Learn Res. 3* (2003), 1371–1382.
23. Shalev-Shwartz, S., Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 32 Avenue of the Americas, New York, NY 10013-2473, USA (2014).
24. Shalev-Shwartz, S., Shamir, O., Srebro, N., Sridharan, K. Learnability, stability and uniform convergence. *J. Mach Learn Res. 11* (2010), 2635–2670.
25. Steinke, T., Ullman, J. Interactive fingerprinting codes and the hardness of preventing false discovery. In *COLT*, Paris, France (2015), 1588–1628.

Cynthia Dwork (dwork@microsoft.com), Microsoft Research, Mountain View, CA.

Vitaly Feldman (vitaly@post.harvard.edu), IBM Almaden Research Center, CA.

Moritz Hardt (m@mrtz.org), Google Research, Mountain View, CA.

Toniann Pitassi (toni@cs.toronto.edu), Department of Computer Science, University of Toronto, Toronto, Ontario, Canada.

Omer Reingold (reingold@stanford.edu), Computer Science Department, Stanford University, Stanford, CA.

Aaron Roth (aaroth@cis.upenn.edu), Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.