adpt: A Differentially Private Tool for Adaptive Data Analysis

Benson Chen, Lingbin Cai, Jake Hart, and Dylan Sun
Advised by Dr. Aaron Roth

April 26, 2016

Abstract

Statistical analysis forms the quantitative backbone for research across a variety of fields. Textbook statistics requires that hypotheses and methods be chosen prior to gathering data. Unfortunately, this is impractical; researchers instead choose statistical methods in response to the data to which those methods will be applied, a process known as adaptive data analysis. This allows researchers to iteratively design studies based upon intermediate results, but can lead to spurious conclusions because it invalidates assumptions underpinning statistics.

We approach this issue both practically and theoretically. We apply current research in differential privacy in an implementation that mitigates the negative consequences of adaptive data analysis. Next, we investigate conditions in which our tool is effective and compare its performance to other, similarly-motivated approaches.
1 Introduction

Problematic statistical analysis is frighteningly prevalent. From claims that men who work out are more likely to be politically conservative to claims that overweight people judge distances less accurately, spurious conclusions abound [2][3]. Unfortunately, research papers that may not be completely statistically valid are still published in leading journals in their respective fields. This unfortunately leads to mainstream media outlets publicizing these often outlandish conclusions and results in the spread of misinformation to the general public.

The root cause of much of this problematic analysis lies in "overfit." This term describes the phenomenon of when a statistical model describes the random error or noise captured in a test dataset as opposed to a true underlying relationship. Overfit is most often caused by adaptive data analysis. Traditional statistics requires that a model to test any particular research hypothesis is chosen ahead of time - even before data collection. Unfortunately, this requirement is too strict in practice for most researchers; researchers find that they need to get a sense of their test data before constructing a finalized model.

Adaptive data analysis is a process by which a researcher adapts their statistical methods in response to the test data at hand. The difference between adaptive and static data analysis is illustrated in Fig. 1. While this allows an analyst far greater flexibility, it also undermines a core assumption in statistics and can lead to fallacious conclusions much more easily [5].

![Figure 1: Static vs. Adaptive Data Analysis](image)

There are a number of similarly-motivated theories that aim to mitigate the problem of overfit. These include KL Divergence from the field of Information Theory and Post-Selection Inference (PoSI) from statistics. We cooperate with Dr. Aaron Roth and utilize new research in Differential Privacy. Although each approach has its
own benefits and drawbacks, we believe Differential Privacy to be the most naturally suited for adaptive data analysis.

We believe that the majority of researchers who engage in statistically problematic analysis do so out of a lack of understanding of overfit and how to avoid it. We are motivated by the dual desire to improve analysts’ theoretical understanding of how to preserve statistical validity and to implement a tool with which to improve the quality of statistical analysis in practice.

2 Approach

Our goals for this project are two-fold. The overarching idea is to help improve the validity of real-world statistical analysis. We divide our work into two fundamental questions. The first is implementation. How well does Differential Privacy work in practice? By implementing adpt, a Differentially Private toolkit, we are not only able to validate the underlying theory, we are enabling other research analysts to benefit from this new research.

Our second main question is theoretical. When should I use Differential Privacy? We compare Differential Privacy to the other previously mentioned approaches to determine their relative strengths and weaknesses. This should enable researchers to decide whether or not Differential Privacy is a suitable technique for their needs.

To answer these questions, we develop differentially private implementations of forward selection and random forests, two common adaptive algorithms. Our empirical testing suggests that differential privacy can yield improvements when the test dataset is small or when it includes many noisy variables.

We also compare differentially private techniques to methods from information-theory and regression-theory which address similar problems. We identify several key advantages of differentially private techniques, specifically they provide a minimal penalty for adaptive research while providing conservative probabilistic guarantees.

2.1 Thresholdout: A Differentially Private Approach

The adpt product is an implementation of the concept of the reusable holdout set (also referred to as “Thresholdout”), introduced by Dwork et al in August 2015 (see [21] for a practical overview and [7] for a more technical presentation). The primary idea behind Thresholdout is to utilize the exponential method from differential privacy (see [9] for a basic introduction) in order to make a holdout set (also called a test set) reusable. In other words, while a normal test set would only be able
Figure 2: Thresholdout Schematic

to test a single model before giving up the guarantee of statistical validity, Thresholdout can test a nearly-quadratic number of models before losing statistical validity.

The basic idea of Thresholdout is a modification of the basic idea of a holdout set. When an analyst wants to test a particular model with a certain quality measure (for example, an RMSE), Thresholdout computes the difference in this quality measure between the model on the training set (i.e. the set of data used to construct the model) and the holdout set. If these values are within a certain accuracy factor $\alpha$, then Thresholdout essentially does nothing, returning the quality measure on the training set (i.e. what the analyst gave it)\cite{21}. The reasoning for this is that because the training and holdout sets had similar quality scores, the model did not overfit on the training set. This prevents the analyst from having to “pay” for model choices which do not overfit. On the other hand, if the quality measures do differ, then Thresholdout returns the quality measure on the holdout set plus some noise (proportional to the threshold). It’s this noise that allows the holdout set to be reusable. That is, the noise added provides differential privacy to the holdout set, preventing the analyst from easily overfitting on future iterations. Please see section 5 for more detail on this; for now, we address how this algorithm was implemented effectively.
2.2 Algorithm Selection

Our initial idea was to implement an overlay that would be able to render any algorithm differentially private. This would have been the most extensible solution, allowing a research analyst to benefit from Differential Privacy no matter what algorithm they may require. To this end, we cooperated with Dr. Andreas Haeberlen to learn more about his research creating Fuzz [4]. However, we find that with this design, it is extremely difficult to guarantee differential privacy across a wide range of scenarios. We therefore instead pursue the implementation of select, common, adaptive algorithms.

For our toolkit, we implement two specific algorithms that incorporates differential privacy in order to test how the theory works empirically. The two algorithms that we implement are forward selection and random forests.

Forward selection is a very common statistical analysis technique used to find the best subset of features to use in a model. This greedy algorithm involves many iterations to find the best feature to add to the model at each step. The repeated querying from a test set is what makes forward selection a prime candidate to apply Differential Privacy to see how we can improve under an adaptive situation.

While forward selection deals with regression problems, random forests are often used for categorical response variables. We could have also chosen to implement logistic regression, but such an algorithm would be very similar to forward selection. In order to determine how well differential privacy can work on an altogether different kind of algorithm, a nonparametric one, we instead chose to implement random forests using Differential Privacy.

The following sections will describe the two algorithms in more detail, and discuss their performance on synthetic and real datasets.

2.3 Testing

In order to determine the efficacy of our implementation, we employ a technique called "cross validation." This method of validating a statistical model assesses whether the results given by the analysis will generalize to an independent data set. We accomplish cross validation by splitting a test data set into three parts. Two of these three parts are used in developing the model according to our implementation. The third and final part is kept separate.

For each algorithm that we implement, we choose the appropriate metric defining the success of the model. For forward selection and regression, we use RMSE. For decision trees, we use MCE. Once we generate our statistical model, we apply it to
the reserved data set and calculate our metric. The more accurate our model is at fitting to our reserved data set, the less it has been subject to overfit.

3 Forward Selection

To test the performance of differential privacy on an actual adaptive data analysis algorithm, we first look at a typical regression method. Forward selection is a well-known and widely used adaptive data analysis algorithm based on regression. It is commonly used across many different disciplines because of its time efficiency when dealing with a dataset with a relatively large number of features. Unfortunately, when the algorithm must iterate over said large number of features, it is vulnerable to serious overfitting. The traditional forward selection algorithm works in the following manner:

1. Start with an empty model with no features selected
2. The algorithm tentatively adds a new feature into the current model, trains the model on the training set, and tests it against the holdout set
3. At the end of each round, the algorithm picks the feature with the best performance metric (e.g. RMSE) on the holdout set and adds that feature into the model
4. The algorithm loops between step 2 and 3, and terminates until a fixed number of features have been added, or until the model performance improvement shrinks under certain predetermined threshold

In our differentially private implementation, we change the testing procedure of each round. When the model is tested on the holdout set, the algorithm checks the difference between the true RMSE on the holdout set and the reported RMSE on the training set. If the difference is within a certain threshold parameter, say the differential privacy threshold $T$, we simply return the RMSE on the training set. If the difference exceeds $T$, we return the true RMSE on the holdout set plus Laplacian noise, whose standard deviation is the differential privacy threshold $T$ divided by a Laplacian factor $L$. The specific version of forward selection we implement uses a predetermined threshold to halt the algorithm when R-square improvement does not exceed a certain level.
3.1 Parameters

As noted, we have two important parameters to pre-select, the differential privacy threshold $T$ and the Laplacian factor $L$. The R-square improvement threshold can be arbitrary and in many cases is decided by the users appetite. In our testing procedure, we use 0.5% as the improvement threshold. However, the optimal value for $T$ and $L$ does not have a theoretically ideal solution. Therefore in most of our tests, we use cross validation to find the optimal set of parameters first, then plug those parameters into the algorithm to compare the performance of forward selection with differential privacy and without.

3.2 Testing on Synthetic Data

We first test the forward selection algorithm with differential privacy on the synthetic data. Each testing data set is randomly generated in Python in the following way. We first decide three important parameters, $N$, number of sample points, $G$, number of variables that are in the true model, and $B$, number of noisy variables that are unrelated to the true model. Then we create $G$ columns of good variables randomly following a standard normal distribution. These "good" variables represent true underlying relationships that drive the response variable. Since we always normalize the variables before applying forward selection, the variance of each variable does not matter. We also create an extra column of residuals using a normal distribution of mean 0 and standard deviation 2.5, which could add sufficient noise to confuse the model selector. Next, we randomly generate $G$ coefficients to linearly combine the $G$ columns of good variables, plus a randomly generated intercept and residuals, to construct the response variable $Y$.

$$Y_i = a + \sum_{j=1}^{G} \beta_j * G_j + \epsilon_j$$

The rest $B$ columns of bad variables are randomly generated following standard normal as well. Finally, we concatenate $Y$, $B$ and $G$ columns of variables together to get a fresh random dataset.

By varying mainly $N$ and $B$, we compare the performance of the algorithm with and without differential privacy based on RMSE on a validation set. Our two major results are presented below.
3.2.1 Dataset Size

Fixing $B$, the number of noisy variables, at 200 and $G$, the number of variables from the true model, at 5, we vary $N$ from 200 to 1500 and take the average performance. We generate the following performance observation:

![Graph showing varying dataset size](image)

Figure 3: Varying Dataset Size

We can see improvement in RMSE using differential privacy when the dataset size is no more than twice the number of noisy variables. However, when boosting the sample size way above the number of noisy variables, the traditional forward selection algorithm tends to outperform our implementation. We hypothesize that this may result from the fact that when the sample size increases and outnumbers the number of noisy variables, the influence of the noisy signal on model selection tends to be weaker and overfit is naturally mitigated. The conservative nature of using differential privacy may sometimes limit the selection of good variables, thus affecting the model performance.

3.2.2 Noisy variables

Next, we fix the dataset size $N$ at 200, 500 and 2000, vary $B$, the number of noisy variables from 50 to 550, and observe the effect in Fig. 4.
When the sample size does not dominate the number of noisy variables, increasing the number of noisy variables allows differential privacy to perform very well. At data set sizes of 200 and 500, one can observe improvement in RMSE versus the traditional algorithm that increases with the number of noisy variables. However when the dataset is large enough, the variation in number of noisy variables does not affect the performance much. Intuitively, the presence of a greater number of noisy variables allows a higher probability that a noisy variable randomly performs well and gets selected; thus the overfitting effect is more evident. We conclude that differential privacy provides stronger protection on model validity and shows decent RMSE improvement with a large number of noisy variables.

In general, the use of a differentially private version of forward selection has merit on small datasets with large number of junk features. We believe this to be a relatively common scenario in the fields of political science and economics research for example, where many false discoveries are published.

As an aside, we note one thing about our testing procedure. Generally in regression techniques, it is not possible to perform a regression if the number of features is greater than the number of observations. However, this does not pose a problem for us, even though the number of noisy features may exceed the number of obser-
vations, because we are using a forward selection algorithm. That is, we will never select enough variables to exceed the number of observations, because if the addition of a variable does not increase the model R-square value by a certain amount, the algorithm terminates.

3.3 Testing on Real Data
Moving on to tests on real data, we test our algorithm on a Florida crime rate dataset which contains about 2000 observations and 100 features. We take various sized subsets from the dataset and conduct a comparison between normal and differentially private forward selection. We corroborate our findings from the synthetic data. With only 100 features, subsets of our data of size less than 200 tend to favor differential privacy. Our hypothesis that a differentially private forward selection algorithm works best with a large number of noisy variables and smaller sample sizes holds in this case.

Figure 5: Real Data Test Results
3.4 Parameter Search

With cross validation implemented, the time efficiency of such algorithm gets sacrificed. Therefore, we also attempt to search for a deterministic model for optimal parameters values for the differential privacy threshold and Laplacian factor, or the ratio between them.

We conduct a regression of the optimal parameter calculated from cross validation on a number of statistics of the corresponding dataset, such as dataset size, number of variables, skewness and kurtosis of variables, etc.

From the regression, the explaining power from the listed statistics is limited. Regression on the optimal ratio of the two parameters only has dataset size as a statistically significant variable, with R-square of 17.2%. The difficulty of finding the theoretically optimal parameter pushed us to check if the effort on cross validation is necessary. With the test result on the Florida crime dataset with arbitrarily determined parameters, we are able to find that RMSE improvement shows similar pattern, which suggests that the importance of cross-validated optimal parameters are less important than expected.

![Parameter Search](image)

Figure 6: Parameter Search

Nevertheless, the problem of how to find the theoretically optimal parameters remains a question to be further explored. Otherwise, the key takeaway is that the power of differential privacy on forward selection is limited on specific sized dataset, where $p > n$ is usually a signal for good performance.
4 Random Forest

The typical random forest algorithm works by constructing many decision trees and averaging the results from the individual trees. Because the construction of a decision tree is deterministic in nature, a regularization technique is used to select a subset of all the features for each tree.

Random forest is a good algorithm to which to apply differential privacy, because it is an algorithm that is easy to overfit. While there are many regularization techniques to avoid the problem of overfitting, due to the nonparametric nature of the algorithm, it is always possible to overfit on the data. Therefore, we predict that there are gains from differential privacy; however, as with the case of forward selection, the gains from differential privacy are most prominent when the data is noisy.

In our implementation of decision trees, we branch off of features based on the entropy, or information gain. More specifically, for each decision tree, we use the following algorithm to determine the structure of the tree.

1. For each decision tree, choose a different random subset of the features

2. At each split do the following for each feature:

   (a) If not using differential privacy: calculate the split value that results in the largest information gain on the test set

   (b) If using differential privacy: for each possible split value, calculate the information gain on both the train and test set. If the two values do not differ by more than some threshold, return the train information gain. Otherwise, return the information gain on the test set with laplacian noise added. Choose the split value that results in the highest information gain.

3. Splitting the data until the maximum depth is reached, or there is no longer any more data to split on.

4. If the current node in the tree is a leaf, calculate the class probabilities for that node.

The random forests simply consist of many of these decision trees. In order to evaluate data on the random forest, we average the result of each individual tree. Each tree will return the probability of being in each of the response classes, and we take an average to compute the final probability of being in each class.
Unlike in forward selection, we do not normalize the data in this case. This is due to the fact that normalizing the data does not change our analysis, because the information gain calculated at each step would be the same, just with different split values.

4.1 Testing on Synthetic Data

We generate the synthetic data to test random forests in a similar fashion to how we generated data sets for forward selection. There are two kinds of features that we generated: good and bad. As before, good features have some predictive power (though they still contain some amount of noise and are not perfect), and bad features are generated completely randomly.

We first generate a dataset with a binary response variable, with half the observations having the positive class, and half having the negative class. The bad features are simply random Gaussian noise regardless of the response variable. On the other hand, each good feature is drawn from one of two Gaussian distributions depending on the response variable. That is, the conditional distribution for each good variable is a Gaussian distribution. We set the parameters of the Gaussians for the good features to ensure that there existed some overlap between the Gaussians for each of the response classes. This ensures that there exists some noise in even the good features. For simplicity, we keep the standard deviation of the two conditional Gaussians the same.

The equations for the features can be seen below:

\[
X_{bad} \sim N(\mu, \sigma) \\
X_{good|y} \sim N(\mu_y, \sigma)
\]

Unlike with the forward selection algorithm, our implementation of decision trees has many more parameters that can be tuned. Therefore, we only include the parameters that we believe to be the most important in the analysis to follow.

Consistent with our observations with forward selection, when there is more noise to the dataset, differential privacy works better. In this case, because the nature of our synthetic data is such that the good variables consist of two conditional Gaussian distributions, when the two Gaussians are closer together, the dataset is more noisy. In other words, it is harder for the model to find a value to split on, because many values will correspond to either a class label of 1 or 0.

Let \(\mu_{diff}\) denote the distance between the means of the conditional gaussians. In the graph below, we test the performance of differential privacy using \(\mu_{diff} = 0.6\sigma\).
and $1.0\sigma$. As explained above, when $\mu_{\text{diff}}$ is smaller, the data is more noisy, so we would expect greater gains from differential privacy. That is exactly what we observe.

![Graph showing improvement in MCE using differential privacy](image)

**Figure 7: Synthetic Data Results**

From Fig. 7, we also observe the general trend of MCE improvement decreasing with dataset size. Consistent with our findings in the forward selection case, increasing the dataset size decreases the gains of using differential privacy.

We also investigate the effects of increasing the number of noisy features. However, due to the nature of our algorithm, which chooses a random subset of the features for each decision tree, we must choose a larger subset when the total number of noisy features are increased. If we do not, the model will not perform well, because there is a lower probability that the good variables are even in the subset of features. In this case, we define the size of the subset to be a fourth of the total number of variables.

The trend that we observe is less obvious as we see in Fig. 8, deviating from that of forward selection. This is probably due to greater complexity of the random forests algorithm - by increasing the number of bad variables, we change many properties of how the decision trees work.

Nevertheless, we still see some improvement when we increase the number of noisy variables in the model.
4.2 Testing on Real Data

We test our random forest model on a real dataset with hospital readmissions data. This dataset contains many features regarding a patient’s initial hospital admission, and the response variable is whether or not a particular patient has been readmitted to the hospital at a later date. Some of the features in this dataset include information about patient demographics, diagnosis, medication, and procedures. We choose this dataset because it contains many variables that are probably not very predictive. That is, we are fairly certain that this dataset contains a lot of noisy variables. For instance, there are over 50 different diagnoses. With so many different diagnoses, one is likely to be significant in a model by pure chance alone.

Indeed, when we run our random forest algorithm with and without differential privacy, we see that the validation error decreases from 46.4% to 43.6%, which is a significant decrease. As mentioned earlier, random forest is very prone to overfitting because it is nonparametric, and can fit training data arbitrarily well. The high amount of MCE in general signifies the noisy nature of the dataset, and as we have seen before, the more noisy a dataset is, the greater the benefits of differential privacy.
5 Theoretical Work

We perform a survey of the literature on work in preventing false discovery, with an aim towards understanding the similarities and dissimilarities between the various proposed methods of avoiding overfit.

5.1 The Theoretical Need for adpt

The work Dwork et al. and Bassily et al. forms the basis for adpt. In this framework, the main objective is based upon statistical learning theory. In particular, there is some distribution $P$ over a finite universe $\chi$. We have a set $x$ consisting of $n$ independent samples taken from $P$. There is also some mechanism $M$ which has access only to $x$ (not $P$), but would like to provide some information about $P$ based upon $x$[6].

In particular, a data analyst (for example, an adpt user) is interested in performing a set of $k$ queries on the data, denoted $q = q_1, q_2, \ldots, q_k$. Each of these queries is drawn from some family of functions which map from $\chi$ to the real line. adpt is an example of such a mechanism, and in the context of adpt, these queries may be the Root Mean Squared Error (RMSE) of a given model, for example. The mechanism’s goal is to answer these queries accurately, which is to say that the answer to the query which the mechanism gives should be close (i.e. within some additive error $\alpha$) of the answer to the query on the entire distribution $P$, with high probability. The central question is: how large a sample $x$ is needed to respond to such queries accurately.

Consider the most simple case of this problem: when the analyst asks only a single query $q_1$. Again, concretely, perhaps this would just ask for the RMSE of a given model. This is just a simple function, and the natural way for a mechanism to respond to such a query is to compute the sample average of the function, then respond with this value. Using Chernoff Bounds, it is easy to show that a sample of size $O(1/\alpha^2)$ is sufficient to answer this single query. This is nothing more than the output one would get from standard statistical software given a sufficiently large sample. Now generalizing to the case of answering $k$ queries, when samples are chosen independent from previous responses from the mechanism, the size of the sample must be $O(\log k/\alpha^2)$. As many authors have pointed out, this is nothing more than the classic Bonferroni correction from statistics, but we are asking about the size of the sample required rather than some statistical quantity. However, previous work by Dinur and Nissim shows that this bound does not apply in the adaptive data analysis setting, where queries can be chosen in response to previously answered queries [8]. In particular, in order to answer $k$ queries which are chosen adaptively,
a mechanism which uses the sample average of the query must use a sample of size $O(k/\alpha^2)$. In other words, the mechanism must have a fresh $O(1/\alpha^2)$ sample for each of the queries. In the context of an actual experiment, this may correspond to performing expensive medical tests or time intensive observational surveys, and this naive approach may be impractical. This is often the motivation for analysts to perform adaptive analyses on a single sample, even though doing so invalidates statistical guarantees. This forms the theoretical need for a product like adpt.

5.2 Differential Privacy

Dwork et al. and Bassily et al. show how the use of differential privacy can be used in order to provide results which are superior to the aforementioned naive approach. Differential Privacy is a well covered topic in theoretical computer science, and a good overview can be found in Dwork and Roth [9]. The original idea of differential privacy was to protect sensitive information about users by ensuring that the effect of any single piece of information is “small”.

Formally, an algorithm $A$ is called $(\epsilon, \delta)$ differentially private if for all subsets $S$ of $A$’s possible outputs and for all input databases $X$ and $Y$ which differ in only a single element:

$$Pr(A(X) \in S) \leq e^\epsilon Pr(A(Y) \in S) + \delta$$

In other words, a local change in a single element of the input will have a small change on the output, so the probability distribution of the algorithm’s outputs is “stable” in an informal sense.

Now we must ask ourselves how this concept connects to the aforementioned issues in statistical analysis? To view this, we must ask ourselves why the naive method cannot answer adaptively chosen queries using the $O(\log k/\alpha^2)$ sample size for when the queries are chosen independently of previous responses. As we’ve discussed before, the problem with this is “overfit”, where the mechanism’s responses will be accurate only on the sample $x$, rather than on the overall distribution $P$. For example, if we perform RMSE queries on the mechanism for many adaptively chosen models, the models selected by the analyst may eventually be designed to fit only the sample which the mechanism possesses, while having a poor fit on the rest of the distribution. However, if our mechanism is “stable” (i.e. differentially private), then the answers given by the mechanisms on similar samples will be similar; thus, the idiosyncrasies of the sample will be harder for the analyst to fit to. In this way, we might say that a differentially private mechanism such as adpt protects the privacy
of the samples it possesses.

5.2.1 Specific Guarantees of Differential Privacy

The most recent results regarding differential privacy and adaptive data analysis is by Basilly et al. (which simplifies earlier results by Dwork et al.). These results were the basis for the concept of a reusable holdout set, a later paper by Dwork et al. which forms the basis for the adpt product. In this work, Basilly et al. show that differential private algorithms can answer $k$ queries based on the following sample sizes([6], inspired by [7]):

When $k$ is less than $n^2$ and the queries are “statistical” (i.e. the queries ask for the expected value of some function which maps to the range $[0,1]$) or low-sensitivity (queries which map vectors from the data universe to the real line, such that vectors which differ in one bit will have query results that differ by a “small” amount), then

$$n = \tilde{O}\left(\frac{\sqrt{k}}{\alpha^2}\right)$$

samples are sufficient to answer queries accurately using differential privacy. Notice that this is a quadratic improvement over the $O\left(\frac{k}{\alpha^2}\right)$ requirement of the naive mechanism. Furthermore, if we generalized even further to optimization queries used in regression and classification, then

$$n = \tilde{O}\left(\frac{\sqrt{k d}}{\alpha^2}\right)$$

samples are sufficient, where $d$ is the dimension of the parameter space. These results form theoretical foundations of the reusable holdout set [21], which in turn is the basis for adpt, but these results can be generalized to answer even larger numbers of queries, albeit with algorithms which would be computationally inefficient.

In particular, to answer $k > n^2$ statistical queries ([6], based on [7]),

$$n = \tilde{O}\left(\frac{\sqrt{\log |\chi| \log k}}{\alpha^3}\right)$$

samples suffice for accuracy, and

$$n = \tilde{O}\left(\frac{\log |\chi| \log k}{\alpha^3}\right)$$
samples suffice for low sensitivity queries. However, these are more complex mechanisms that adpt which will take exponential time, thus making them unfeasible for most practical applications. A similar result also applies for the aforementioned optimization queries, though we leave this aside for the time being.

5.2.2 Near Optimality of adpt

There is also one more interesting result which justifies our use of adpt. As we mention above, adpt requires that the number of queries performed is less than quadratic in the sample size (i.e. $k < n^2$), while techniques to answer more queries are computationally inefficient. A natural question is the following: is there an efficient algorithm which can answer more than $n^2$ queries. However, recent techniques in cryptography give a negative answer to this question [22], showing that any mechanism which answers than $O(n^2)$ queries cannot be efficient (note that adpt can get around this bound due to the logarithmic factor absorbed by $\tilde{O}$ notation). This result follows from fingerprinting techniques in cryptography, which uses combinatorial arguments to show that any technique which answers queries efficiently cannot preserve privacy after $n^2$ queries. Thus, adpt is within a logarithmic factor of being optimal, and differentially private techniques are thus the most realistic tool for adaptive data analysts to use in this setting [22].

5.3 Information Theoretic Techniques

Another approach to the problem of adaptive data analysis and the related problems of overfit is to use information theory to address the problem directly. This was mostly recently addressed in a paper by Russo and Zuo. An introduction to information theory can be found in [10]. We approach this from the same setting as was originally laid out in section 1.1, that is, an analyst is making $q = q_1, q_2, \ldots, q_k$ queries to a mechanism which has access to a sample $x$ drawn from $\mathcal{P}$. However, rather than considering a randomized (differentially private) algorithm, Russo and Zuo make no assumptions about the mechanism $\mathcal{M}$ used to answer queries. Instead, Russo and Zuo focus on the adaptive selection process itself used by the analyst.

Specifically, Russo and Zuo look at the bias introduced by the adaptive selection process [11]. In particular, they show that, given certain conditions on the queries and on the adaptive data selection process, the bias introduced scales with the square root of the mutual information between the reported query values $q_1(x), q_2(x), \ldots, q_k(x)$ and the selection process $T$ which the analyst uses to adaptively chose the queries. Intuitively, this formalizes the notion that if our selection process $T$ is more sensitive
to noise in the sample $\mathbf{x}$, then $T$ will be more likely to give biased results.

Formally, the authors above show that if $q_i(\mathbf{x}) - E[q_i]$ is $\sigma$-sub-Gaussian (meaning that this random variable is dominated by a normal random variable with standard deviation $\sigma$), then,

$$|E[q_T] - E[E[q_T]]| \leq \sigma \sqrt{2I(T, \sigma)}$$

Where $q_T$ is the query reported by the selection process $T$, and $I$ is the mutual information. Note that the expectation $E[q_T]$ depends both on the realized data $\mathbf{x}$ and the selection process $T$, while $E[E[q_T]]$ depends only upon the model selection process $T$, as the inner $E[q_T]$ denotes the mean value if $q_T$ is evaluated out of sample. A final important component of this result is that many typical queries which data analysts would use (such as those discussed in section 1.2) are covered by this result.

### 5.3.1 Information Theoretic Results vs Differential Privacy

There are several relationships between the differential privacy discussed in section 1.2 and the information theoretic results discussed here. First, this framework provides another way to see why the stability of differential privacy allows us to answer many queries accurately, as the noise added in differentially private algorithms keeps the mutual information low and thus reduces the bias of the results.

Beyond this critical relationship, there are also tradeoffs between the information theoretic techniques and differential privacy. On one hand, the information theoretic techniques do not require the use of any special randomized algorithm, so data analysts could use standard statistical software and still have bounds on the bias introduced. However, this involves a large tradeoff in confidence. Whereas differentially private techniques can be used to achieve high probability bounds on the additive error of queries, these results only give a bound on the expectation (i.e. the bias). Thus, it is difficult to ask analysts to make statistically strong conclusions based only upon information theory. Put another way, the information theory results trades off strength for ease of application.

### 5.3.2 Application to p-values

The information theoretic results also have an interesting and useful application to widely used p-values in statistics. A subproblem within the overall issue of false discovery is the idea of reporting bias, that data analysts dismiss results which do not conform to their starting hypothesis [12]. Such issues are well documented in
fields ranging from the social sciences to medicine [12]. How this often plays out is a special case of the false discovery problem: The analyst performs a variety of statistical tests, each of which may be adaptively chosen; then, the analyst reports only the test performed which had a low (i.e. significant) p-value, while not disclosing the other tests. At a standard 5% significance level, even performing a handful of tests in this fashion can lead to wildly inaccurate conclusions [11]. However, Russo and Zuo show that when that if the mutual information between the reported p-values and the selection procedure is low (e.g. if the hypotheses tested were highly correlated), then analysts might not be at such a risk.

5.4 Post Selection Inference

Yet another method for addressing the false discovery problem comes from the realm of statistics, referred to as Post Selection Inference (PoSI) [14]. However, this technique is dramatically different from the aforementioned two. In particular, PoSI is specialized to the problem of linear regression. That is, PoSI assumes that, given your sample \( x \), you are trying to estimate some value \( y \) based upon the relationship

\[
y = x\beta + N(0, \sigma^2 I_n)
\]

Where \( N \) represents the normal distribution and \( \beta \) is determined by minimizing the sum of squared errors. This is a standard technique, an introduction to which can be found in [13]. In this setting, the typical goal is to derive a confidence interval for the coefficients \( \beta \). In other words, PoSI seeks to find a \( K \) such that

\[
Pr(M \max_j |\beta_{j,M}| \leq K\hat{\sigma}) \geq 1 - \alpha
\]

where \( M \) is any possible submodel from the linear regression, \( j \) is any possible coefficient within that submodel, and \( \hat{\sigma} \) is the standard deviation of the full model [15][16]. Intuitively, this is saying that we want to find a \( K \) such that our confidence intervals for every possible sub model is accurate up to factor \( \alpha \). Again, this is similar to a standard linear regression, but typically (i.e. in the case of fitting a single model) the value for \( K \) is a constant (around 2, corresponding to the typical empirical law for the normal distribution). However, PoSI asks what this \( K \) must be if we fit all possible submodels. In this way, PoSI is not a study of adaptivity. Rather, it is asking what is tightest conservative confidence interval one can obtain given no restrictions on the choices the analyst makes, whereas the previous discussed differentially private techniques aim to pay only for the adaptive choices which the analyst makes.
The main result of PoSI is to show that the value for $K \leq 0.89 \sqrt{p} = O(\sqrt{p})$, where $p$ is the number of predictors in the full model [14].

### 5.4.1 PoSI vs Differentially Private Techniques

As previously discussed, PoSI and Differential Privacy have critical differences in the problems they attempt to address. While Differential Privacy attempts to only address adaptive analyses, PoSI works just as well for an analyst who simply wants conservative results for any possible model they could fit.

One important comparison between these two methods is the required sample size when we an adaptive analyst is also interested in trying every possible submodel of a linear regression. Note that, while the differentially private techniques are unnatural for testing confidence intervals of $\beta$ coefficients, but we could likewise use differentially private techniques to test linear models by testing, for example, the RMSE of each model. If we test both techniques at a significance level of $\alpha$, we know that differential privacy will give us a sample size of

$$n = \tilde{O}(\frac{\log |\chi| \log k}{\alpha^3})$$

$$= \tilde{O}(\frac{\sqrt{\log |\chi|} \log 2^p}{\alpha^3})$$

$$= \tilde{O}(\frac{\sqrt{\log 2^p \log 2^p}}{\alpha^3})$$

$$= \tilde{O}(\frac{p^{3/2}}{\alpha^3})$$

where the latter three identities follow simply by recognizing the number of queries required to test every submodel along with the size of the universe for linear regression.

$$\alpha = O(\frac{\sqrt{p}}{n^{1/2}})$$

$$n^{1/2} = O(\frac{\sqrt{p}}{\alpha})$$

$$n = O(\frac{p}{\alpha^2})$$

Where the $n^{1/2}$ term comes from the standard deviation of the full model.
This gives a nice mathematical characterization of what one might expect: when the analyst tries every possible model, the sample size needed for PoSI is superior, as one might expect given that this is exactly the problem they solve. On the other hand, if one is an adaptive analyst who will try a smaller number of models, it is almost certain that differentially private techniques will be less restrictive.

6 Discussion and Future Work

For both our algorithms, forward selection and random forest, that we implemented with differential privacy, we note that there are already certain aspects of the algorithms that try to prevent overfitting. For instance, with respect to forward selection, by imposing a necessary R-square improvement on for each additional feature, we reduce the possibility of overfitting. In the random forest case, we both limit the depth of the tree as well as select a subset of the features for each tree in order to reduce overfitting. If these measures were not in place, then we would probably see more gains from differential privacy, as it would be more likely to occur.

For the random forests, there are other regularization techniques that we have not included within our implementation. For instance, one is pruning, which reduces the complexity of the decision trees by removing the splits that do not give high predictive powers. In our case, we could have stopped splits of the trees that yielded low information gain. We hypothesize that more regularization would reduce the effects of differential privacy, because regularization general reduces the complexity of the model, which helps mitigate effects of overfitting. And it might be interesting, as a future work, to see if adding these regularization techniques do indeed reduce the benefits of overfitting.

For the random forests, we could have also tried more complicated synthetic data to see how differential privacy works in scenarios where the variables with predictive powers are harder to discern. Due to the nonparametric nature of the algorithm, random forests should be able to pick up on very non-linear patterns, something we do not see with forward selection. Therefore, a natural extension of our synthetic data generation is to make the good features come from more gaussian distributions than two. More specifically, the conditional distribution of a good feature might come from one of two gaussian distributions rather than being from just one. We would expect that differential privacy would work better for this case, because overfitting is more likely in this case, in which good features are harder to discern for the model.

There is also ample additional work that could be done in theory. There were several very recent papers (i.e. published in the last month) which make even stronger connections between the various ways to address the problem of false statistical dis-
covery. While our team read all of these, they proved to be too lengthy to introduce in the context of this paper. Examples include the use of minimax theory to address adaptive data analysis [17], applications of this work to PAC learning [18] and Bayesian analysis [19], and more robust connections between differential privacy and standard statistical p-values [20].

7 Ethical Considerations

Although it is not immediately obvious how our work in statistical validity may have ethical considerations, the reality is that statistical analysis has a broader and more impactful reach than any other field. This is simply because statistics is used in research across so many fields, from political science and economics to medicine and psychology and everything between.

We view the misapplication of statistical analysis, whether unintentional or malicious, as a fundamental detriment to the progress of human society. For instance, it is a commonly held joke that almost anything we contact on a daily basis is a carcinogen. The prevalence of studies claiming that a substance causes cancer is partly driven by fallacious statistical conclusions and undermines legitimate work being done in the field to determine substances that are truly and empirically harmful.

We hope that our project and similarly-motivated work will mitigate the negative consequences of problematic statistics in two ways. First, by providing practical and easy-to-use solutions for research analysts, we translate cutting-edge research into an implementation that researchers can use to improve their results immediately. Second, by publicizing the need for analysts to pay more attention to statistical validity, we hope to encourage the research community to rethink its approach and to reduce the number of spurious conclusions published in prestigious journals. This will reduce the spread of misinformation and potentially significantly increase the number of legitimate new discoveries.

8 Summary

Our results demonstrate that Thresholdout and Differential Privacy are indeed effective under certain circumstances. In general, in test data sets with either a small number of data points, a large number of noisy variables, or both, adpt is able to outperform traditional statistical algorithms. We determine that Differential Privacy is very well suited to adaptive data analysis compared to similarly-motivated
approaches. Its main weakness is its difficulty in implementation. This, we hope to ameliorate with our tool, adpt.

Acknowledgment

We would like to thank Dr. Aaron Roth for his guidance during our Senior Design project. Not only was his research the inspiration behind our work, he patiently taught us the appropriate background material and helped inform the design of our tool kit.

We would also like to thank Professors Ani Nenkova and Jonathan Smith for their instruction of CIS 400 and CIS 401.

References


[16] Unpublished notes from Dr. Larry Brown. Please contact Jake Hart (jhart@seas.upenn.edu) for a copy if needed.


