# Exploring Privacy-Accuracy Tradeoffs using DPComp

Michael Hay*, Ashwin Machanavajjhala**, Gerome Miklau†,
Yan Chen**, Dan Zhang†, George Bissias†

* Colgate University
Department of Computer Science
mhay@colgate.edu

** Duke University
Department of Computer Science
{ashwin,yanchen}@cs.duke.edu

† UMass Amherst
College of Computing and Information Sciences
{miklau,dzhang,gbiss}@cs.umass.edu

## ABSTRACT

The emergence of differential privacy as a primary standard for privacy protection has led to the development, by the research community, of hundreds of algorithms for various data analysis tasks. Yet deployment of these techniques has been slowed by the complexity of algorithms and an incomplete understanding of the cost to accuracy implied by the adoption of differential privacy.

In this demonstration we present DPCOMP, a publicly-accessible web-based system, designed to support a broad community of users, including data analysts, privacy researchers, and data owners. Users can use DPCOMP to assess the accuracy of state-of-the-art privacy algorithms and interactively explore algorithm output in order to understand, both quantitatively and qualitatively, the error introduced by the algorithms. In addition, users can contribute new algorithms and new (non-sensitive) datasets. DPCOMP automatically incorporates user contributions into an evolving benchmark based on a rigorous evaluation methodology articulated by Hay et al. [4].

## 1. INTRODUCTION

Privacy concerns are a major obstacle to deriving the scientific insights now possible from increasing data collection and powerful new data analysis techniques. The goal of privacy technology is to permit data mining and analysis tasks to be safely carried out over a collection of sensitive records donated by individuals.

Differential privacy [2] has emerged as an important standard for protection of individuals' sensitive information. Differential privacy offers the compelling guarantee that the output the analyst receives is statistically indistinguishable (governed by a privacy parameter $\epsilon$) from the output the analyst would have received if any one individual had opted out of the collection. Smaller $\epsilon$ implies less disclosure about any single individual.

General acceptance of differential privacy by researchers has led to a steady stream of research in the database community, as well as the data mining, theory, machine learning, programming languages, security, and statistics communities. For any given task (e.g., answering range queries or learning decision trees) there are a number of specialized differentially private algorithms for completing that task. They typically achieve privacy by adding noise. While smaller $\epsilon$ values generally result in more noise added, algorithms differ in how noise is added and its magnitude even for the same privacy level $\epsilon$.

Despite maturing research efforts, the adoption of differential privacy by practitioners in industry, academia, or government agencies has so far been rare, with only a few examples to mention [3,6]. A major obstacle to practical adoption is that the "cost" of privacy, in terms of degraded accuracy, is difficult for practitioners to assess. This is because, for a given task:

- The research community has failed to clearly identify the state-of-the-art in terms of accuracy for a given privacy level (determined by $\epsilon$), and
- Practitioners, who may not be experts in privacy (or even computer science), have no tools to try out state-of-the-art algorithms and understand the impact of noise on their data.

Assessing the accuracy of differentially private algorithms is complex for the following reasons. Theoretical bounds on error, when known, explain only worst case behavior and hide critical constant factors. Despite hundreds of papers on differentially private algorithms, there has been little attention paid to empirical evaluation methodology and no benchmarks have been established. Further, the accuracy of emerging algorithms has complex dependencies on the input data: an algorithm which offers state-of-the-art error rates on one dataset may substantially underperform a simple baseline technique on another dataset. Researchers have unfortunately reused a small collection of datasets for evaluation and it is therefore difficult to extrapolate performance from published results.

All of these factors mean a practitioner cannot easily identify the best algorithm for a particular task and dataset. She would have to survey literature from many different fields, and either extrapolate from multiple theoretical and empirical analyses, or implement all the algorithms and compare their performance. Moreover, there is a paucity of tools for helping practitioners understand the effect of noise on accuracy. For instance, the emphasis of published empirical evaluations often tends to be comparative (does algorithm $\mathbf{A}_{new}$ have lower error than algorithm $\mathbf{A}_{old}$?) rather than absolute (is the error in the output of $\mathbf{A}_{new}$ acceptable?); the latter being more important to the practitioner.

To address these important challenges we are building DPCOMP, a public web-based forum to support the principled evaluation of private data analysis and to encourage dissemination of related code and data. DPCOMP provides the infrastructure to rigorously evaluate a wide range of differentially private algorithms, for a wide range of tasks, and on a wide range of datasets. Users can view the results of careful, thorough empirical evaluation by interacting with dynamic visualizations. But most importantly, users can contribute datasets, task definitions, and algorithms to DPCOMP, triggering automatic evaluations and comparisons of empirical performance.



*SIGMOD'16, June 26-July 01, 2016, San Francisco, CA, USA*



DOI: http://dx.doi.org/10.1145/2882903.2899387

For the researcher, DPCOMP provides an evolving benchmark of datasets for various tasks, and a thorough comparison of existing techniques, illuminating performance gaps of existing techniques and open problems. For the practitioner, DPCOMP provides an easy way to explore the performance of state-of-the-art algorithms as well as tools to quantitatively and qualitatively assess the absolute error of algorithms.

Our proposed demonstration will present Version 1.0 of DPCOMP, which is focused on a limited but expressive class of data analysis tasks, in which sets of one- and two-dimensional counting queries are answered privately. Accurately supporting these tasks is essential for exploratory analysis on private data (e.g., summary statistics, marginal distributions), as well as building more complex privacy algorithms with these tasks as subroutines (e.g., density estimation, machine learning, log-linear modeling). DPCOMP utilizes a benchmark and a principled evaluation methodology that will be presented during the coinciding research track at SIGMOD 2016 [4]. This evaluation methodology has already revealed weaknesses and inconsistencies in the empirical evaluations of published algorithms. By adopting this methodology, DPCOMP provides a sound, reproducible view of the empirical performance of current privacy algorithms.

At the demonstration, an attendee can play the role of researcher or practitioner. As researcher, they can interactively visualize the complex relationships between privacy level $\epsilon$, data characteristics, and accuracy on a large set of benchmark datasets. As practitioner, they may upload a new dataset and use DPCOMP to interactively explore privacy-accuracy tradeoffs and assess the usefulness of the private outputs on the chosen dataset.

## 2. BACKGROUND

Informally, an algorithm $\mathcal{A}$ is differentially private if its behavior is insensitive to small changes in the input database. Formally, $\mathcal{A}$ satisfies $\epsilon$-differential privacy [2] if for all databases $D$ and $D'$ such that $D'$ contains one additional record, and for any subset of outputs $S \subseteq Range(\mathcal{A})$, $Pr(\mathcal{A}(D) \in S) \leq e^{\epsilon} \times Pr(\mathcal{A}(D') \in S)$. An example of a differentially private algorithm is given below.

We use the term *task* to denote a certain analysis on the database. In DPCOMP Version 1.0, we focus on the following task: given a fixed set of attributes and a *workload* of multi-dimensional range-count queries over those attributes, compute answers to each of the queries. The number of attributes (aka dimensions) is small: we focus on the 1- and 2D cases. An algorithm for this task returns a vector of noisy answers. The main performance metric for these algorithms is error, which is measured as the $L_2$ distance between the true answers and the noisy answers. Notice that error is a random variable. In addition to reporting mean error, it is also important to estimate variance since an analyst only observes one output.

We give an example algorithm for the task of answering the workload of all range-count queries over a single attribute $A$. The PARTITION algorithm first constructs an equi-width histogram with each bucket containing $k$ contiguous values from the domain of $A$ and then computes the count of each bucket. To achieve privacy, it invokes the Laplace mechanism [2], which adds independent Laplace noise with scale $1/\epsilon$ to each bucket count. This noisy histogram can then be used to answer any range query in the workload by summing the appropriate noisy bucket counts. (Buckets on the range boundary only contribute a fraction of their count proportional to the number of values included in the range). To invoke this algorithm, one must specify, *a priori*, a value for $k$. Setting $k$ to be large reduces the error due to noise (fewer noisy counts are summed) but coarsens the granularity of the histogram.

The core idea of PARTITION, grouping together sets of values, is one key idea underlying many of the algorithms proposed for this task (cf. [4]). However, the grouping strategies proposed are much more sophisticated than PARTITION and include adaptively selecting the partition based on the data (which is non-trivial and requires careful analysis to prove differential privacy).

The above example illustrates one of the aforementioned challenges with empirical evaluation: algorithm performance can be *data dependent*. In the case of PARTITION, the error decreases the more uniform the data distribution is within each bucket. This problem is further exacerbated by more sophisticated algorithms that respond adaptively to the data, making it hard to theoretically analyze performance.

In a companion paper [4], we identify 4 factors that influence the performance of differentially private algorithms: (i) $\epsilon$, the privacy parameter, (ii) *scale*, the number of records in the database, (iii) *domain size*, the number of possible values each tuple in the database can take, and (iv) *shape*, or the empirical distribution of the data. The data generation procedure in [4] allows us to vary each of these parameters independently and thereby measure its effect on algorithm performance. In addition, we show that almost all private algorithms satisfy a property called *scale-epsilon exchangeability*, wherein for a fixed domain size and shape, increasing scale has an identical effect on error as increasing $\epsilon$. Thus, two parameters, scale and $\epsilon$, can be collapsed into a single one (the scale-$\epsilon$ product) which captures the strength of the "signal" from the data that competes with the "noise" injected by the algorithm.

## 3. THE DPCOMP SYSTEM

### 3.1 Vision and motivation

DPCOMP is a publicly-accessible web-based system designed to support a broad community of users: *data analysts* who are novice users of privacy mechanisms, *researchers* who develop new privacy algorithms, and *data owners* who manage sensitive data.

For a given data analysis task, DPCOMP automatically performs simulations of privacy algorithms running on real and synthetic datasets and presents a comprehensive analysis of algorithm performance. Users can assess accuracy of the output through various metrics, by comparison with informative baselines, and by visualizing the private output. Users can determine the best algorithm for their task and data, as well as the settings of privacy parameters that can be supported while retaining acceptable accuracy.

Users can also contribute to DPCOMP by uploading datasets, new algorithms, or new task descriptions. Doing so automatically triggers new simulations, the results of which are stored and made available to all users.

The main goals of DPCOMP are: (1) to improve the accessibility and transparency of privacy algorithms by allowing easy browsing, performance comparison, and download; (2) to provide a reproducible methodology for evaluating the performance of privacy algorithms (based on the principles and benchmark from [4]); and (3) to help guide future research by highlighting cases for which existing privacy algorithms do not provide acceptable accuracy.

DPCOMP was inspired by the MLCOMP website which was designed for "objectively comparing machine learning programs across various datasets" [5]. MLCOMP is useful because performance comparisons for algorithm selection for classical machine learning tasks has become complex for non-experts as well as algorithm designers. DPCOMP addresses a similar set of issues, but differs substantially from MLCOMP because our emphasis is solely on algorithms that offer formal privacy guarantees.

Note that it is *not* a goal of DPCOMP to host sensitive data sets. Instead, DPCOMP will provide a public environment in which to

explore and evaluate private algorithms on *public* data. To support users with sensitive data, we instead make the entire infrastructure (task descriptions, algorithm implementations, and execution framework) available for local execution, so that users can easily reproduce results on their own sensitive datasets. (This will initially be done through an open-source code base; in future versions, a virtual machine image with code and data will be available.) Even though local execution is available, the public DPCOMP interface allows users to efficiently identify the algorithms likely to be best for their task.

## 3.2 Features of DPComp 1.0

Our proposed demonstration focuses on version 1.0 of DPCOMP which includes the following main capabilities and user-interactions:

**Algorithm exploration** DPCOMP will allow users to review the range of algorithms available for a task and to assess their performance as a function of both privacy parameters (i.e. epsilon), data parameters (e.g. scale, domain size, etc.) and algorithm parameters (i.e. internal divisions of the privacy budget, number of rounds of iterative algorithms, etc.). Results are provided in the form of dynamic, interactive plots showing not just comparisons of error rates for competing algorithms but also comparisons with natural baselines such as sampling error that help the user understand whether measured error rates are acceptable in practice.

**Dataset contribution** Data owners who wish to find an algorithm that works well on their data can submit (non-sensitive) sample data sets. Users can also design and submit synthetic datasets that either challenge existing algorithms or serve to distinguish competing algorithms. New datasets will automatically be tested on all applicable tasks and algorithms. As an important consequence, DPCOMP will thereby increase the variety of input data used for evaluation of algorithms.

**Workload contribution** Algorithms may perform better or worse on different workloads. Users can contribute workloads that distinguish the performance of competing algorithms.

**Algorithm contribution** Privacy researchers can submit new algorithms to DPCOMP which will be automatically evaluated and compared against existing algorithms and baselines. Submitted programs will be required to conform to a standard API but may be implemented in any programming language.

**Algorithm and benchmark download** Algorithms contributed to DPCOMP will be in the public domain and the DPCOMP website will support easy download of all implemented algorithms, along with datasets, workload definitions, and evaluation infrastructure (following [4]). This allows data owners to easily evaluate algorithms on their own sensitive data using their own systems.

## 3.3 System design of DPComp 1.0

DPCOMP is an extensible execution framework implemented in Python that schedules simulated trials on a large compute cluster with a total of 13,000 cores. Results are collected and stored in a cloud-based database system that underlies the web-based plotting and visualization components.

As an example, for the one-dimensional case, and before any user contributions, DPCOMP contains 14 published algorithms which are evaluated with respect to 2 workloads on 18 different source datasets across 4 domain sizes and 6 settings of scale/epsilon resulting in a total of $14 * 2 * 18 * 4 * 6 = 12,096$ experimental configurations. (The 2-D case has a similar number of configurations.) Because the privacy algorithms are randomized, to calculate high-confidence error measures, we run at least 25 trials of each configuration, resulting in approximately 300,000 independent trials. The computation time for each trial varies considerably across algorithms and algorithm parameters (between a few seconds and a few minutes, with an average of 22 seconds per trial) and also depends on the domain size and workload size. This amounts to 76 days of single-core computation time, which can be completed in under 2 hours using 1,000 cores.

In order to provide interactivity, DPCOMP performs selective, incremental evaluation, prioritizing simulations based on user requests and incrementally presenting partial results (e.g. from fewer than 25 trials) as they arrive. Specific examples are provided in the next section.

## 3.4 Features of DPComp 2.0 (Future work)

The future development of DPCOMP will include additional features not part of the proposed demonstration. First, we plan to expand our system to support additional private analysis tasks such as graph analysis, machine learning, and continual observation. Second, we will develop empirical methods for testing claimed privacy guarantees. DPCOMP Version 1.0 currently assumes that privacy claims are correct, which is acceptable since its current focus is on published algorithms for which privacy proofs are provided. Verifying that an arbitrary program is differentially private is not feasible, but we plan to develop black-box, empirical testing methods that can catch overt violations and errors. Third, we hope that DPCOMP may enable privacy contests which might, for example, challenge algorithm designers to achieve an accuracy target for a specific task and reward the winning algorithm. In the spirit of Kaggle [1], DPCOMP will be available to host competitions initiated by outside users.[1]

# 4. DEMONSTRATION EXPERIENCE

A demo attendee can choose one of two user experiences. They can play the role of a *researcher* who is interested in understanding the state of the art in differential privacy. As a researcher, the attendee can use DPCOMP to explore algorithm performance across a range of conditions and observe key relationships. Alternatively, the attendee can play the role of a *practitioner* who wants to perform a differentially private analysis on a particular dataset. They can use DPCOMP to find the algorithms that offer the best privacy-accuracy tradeoff for that dataset and also examine, both quantitatively and qualitatively, how the noise added for privacy impacts the usefulness of the analysis output.

**Researcher experience** The attendee will be directed to the DPCOMP "dashboard," which presents a summary of the complete set of empirical evaluations that have been computed by DPCOMP to date. The sheer volume of results presents an information visualization challenge. Our demo responds to this challenge in two ways. First, the visualizations are informed by our companion benchmark study [4] which relates algorithm performance to three properties of the input: scale-epsilon product, domain size, and data shape. The benchmark's data generator allows us to vary each of these properties independently, making it possible to measure and visualize the effect of each in isolation. The dashboard will have a separate figure for each property with menus and sliders to control the range of values for each. Second, the dashboard is interactive, allowing the user to do things such as focus on particular algorithms of interest, adjust the values of parameters, and link an algorithm's performance *across* multiple figures to see interaction effects between properties.

After the attendee has had a chance to become oriented to the

[1]Privacy competitions are a new phenomenon: e.g., a company called Hypios hosted a data anonymization contest (personal communication, 2013).

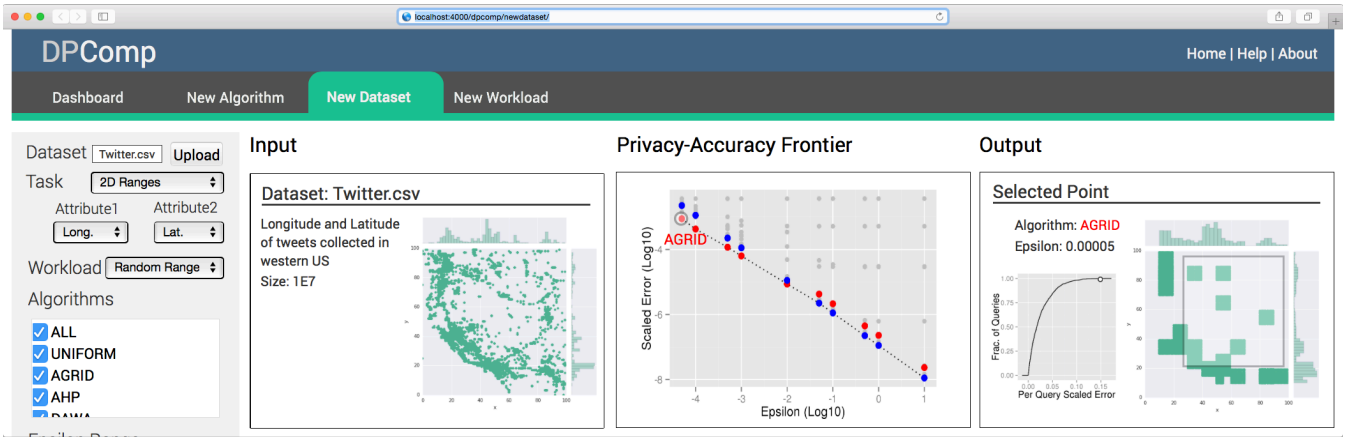


Figure 1: A screen shot of DPCOMP computing the privacy-accuracy frontier on a dataset uploaded by the user (details in Section 4).

dashboard, a demonstrator will guide the attendee to explore some interesting relationships. For example: (1) they will find that the performance "frontier" is diverse: no single algorithm dominates across all settings, but some algorithms do dominate for specific ranges of the input; (2) algorithm performance has a strong dependence on both epsilon and the scale of the data and that these two factors are exchangeable: if a given epsilon value is desired, utility can be improved by collecting more data; (3) in some cases, sophisticated algorithms do not outperform simple baselines.

**Practitioner experience** Given the realities of the SIGMOD demo setting, we do not expect attendees to come with a dataset in hand. Instead, to facilitate interaction, we will make available a set of datasets (in .csv format) that the attendee can upload to the system. (They could alternatively browse the web to find a dataset that is of interest and of suitable size and format for upload.)

Once the dataset is uploaded and the attendee has selected a task, the system will automatically evaluate all available algorithms on this new dataset for the given task. The system will compute and visualize the privacy-accuracy "frontier"—namely, for each setting of the privacy parameter $\epsilon$, it will identify the set of algorithms that achieve highest accuracy on this dataset. (Using our execution framework, initial results can be obtained in under 30 seconds and additional trials are incrementally retrieved.) DPCOMP allows the attendee to explore points along the privacy-accuracy frontier, to get a qualitative sense of what the best algorithm can achieve at the given level of privacy.

An example screen of DPCOMP is shown in Fig. 1. Here, the user has uploaded `Twitter.csv`—a dataset of about $10^7$ tweets collected using the Twitter API—and selected the task of answering 2D range queries over attributes corresponding to latitude and longitude. The user can specify other aspects of the evaluation, such as the workload, the set of algorithms, the range of $\epsilon$, etc. DPCOMP computes and displays the privacy-accuracy frontier. Each point in this figure reports the error (y-axis) of a particular algorithm for the $\epsilon$ shown on the x-axis. Points associated with a particular algorithm are colored if the algorithm has the highest accuracy for at least one value of $\epsilon$.

The user can click on a point along the frontier. In Fig. 1, the user has clicked on the point corresponding to the algorithm that achieves the lowest error at the smallest value of $\epsilon$, which is the algorithm AGRID [7]. For the clicked point, a more detailed analysis is shown. First, it shows a visualization of the data released by the algorithm which can be compared against the input. (The visualization is a 2D histogram over the domain, which can be derived from the output of the algorithm. A point is drawn if the count of the corresponding histogram bucket is above a threshold.) In addition, the error distribution across queries in the workload is displayed. The attendee can click on a point in this inset figure and see the corresponding workload query. In this way, the attendee can get an intuition for which kinds of queries are answered most accurately.

The detailed view for the selected point shows AGRID at $\epsilon = 0.00005$. As is apparent from the visualization, AGRID dynamically partitions the data into a grid and summarizes the data at the grid level. Moreover, the visualization gives the practitioner qualitative feedback about the absolute error of an algorithm. A practitioner may decide $\epsilon = 0.00005$ is impractical from an accuracy standpoint, since even the best algorithm only preserves coarse statistics about the data. In addition, the highlighted range query suggests that some of the highest error queries are those that only partially overlap with dense regions.

In summary, DPCOMP will allow attendees to see that the state-of-the-art in differential privacy contains a diverse collection of algorithms, which can in some settings significantly outperform simple methods like the Laplace and exponential mechanisms. However, their performance depends in complex ways on 4 key characteristics of the input. DPCOMP helps the user identify the best algorithms for a given task and dataset. Further, it provides the user with not only a quantitative evaluation of error but also, through visualizations, a qualitative feel for how the noise added for privacy affects the analysis tasks.

**Acknowledgments** We appreciate the comments of the anonymous reviewers. This material is based upon work supported by the National Science Foundation under Grant Nos. 1012748, 1253327, 1408982, 1443014, 1409125, and 1409143.

## 5. REFERENCES

[1] Kaggle. www.kaggle.com, 2015.
[2] C. Dwork, F. M. K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.
[3] Ú. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *CCS*, 2014.
[4] M. Hay, A. Machanavajjhala, G. Miklau, Y. Chen, and D. Zhang. Principled evaluation of differentially private algorithms using DPBench. SIGMOD Conference, 2016.
[5] P. Liang and J. Abernathy. MLcomp. www.mlcomp.org, 2013.
[6] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *ICDE*, 2008.
[7] W. Qardaji, W. Yang, and N. Li. Differentially private grids for geospatial data. In *ICDE*, 2013.