Demonstration of Damson: Differential Privacy for Analysis of Large Data

Marianne Winslett[1,2] , Yin Yang[1,2], Zhenjie Zhang[1]

[1]*Advanced Digital Sciences Center*, *Singapore*

{yin.yang, zhenjie}@adsc.com.sg

[2]*University of Illinois at Urbana-Champaign*, IL, USA

[2]*winslett@illinois.edu*

***Abstract*— We demonstrate Damson, a novel and powerful tool for publishing the results of biomedical research with strong privacy guarantees. Damson is developed based on the theory of *differential privacy*, which ensures that the adversary cannot infer the presence or absence of any individual from the published results, even with substantial background knowledge. Damson supports a variety of analysis tasks that are common in biomedical studies, including histograms, marginals, data cubes, classification, regression, clustering, and ad-hoc selection-counts. Additionally, Damson contains an effective query optimization engine, which obtains high accuracy for analysis results, while minimizing the privacy costs of performing such analysis.**

I. INTRODUCTION

Privacy concerns have been a major hurdle in providing researchers with biomedical data. Simple de-identification efforts, such as removing the name and ID of each individual, often fail to provide sufficient privacy protection. The main reason is although the adversary cannot directly retrieve the identities of the individuals in the published data, she can often re-identify the individuals by combining the data with additional background knowledge. Data security researchers have devised robust and generic algorithms for re-identifying individuals. For instance, the authors of [13] successfully re-identified individuals in the anonymized Netflix movie rating dataset (www.netflixprize.com), using only background knowledge obtained from the publicly assessable IMDB database (www.imdb.com).

Furthermore, for some biomedical datasets, even statistical information derived from the data can potentially reveal private information. A well-known example concerns genome-wide association studies (GWASs), a hot topic in bioinformatics which studies the DNA samples of a group of patients of a certain disease (e.g., diabetes), in order to discover possible correlations between specific portions (called SNPs) of the human DNA and the disease. A recently developed attack [7] re-identifies an individual using a DNA sample of the target individual and a reference population (e.g., Europeans, Asians, etc.) from the public HapMap repository (www.hapmap.org). An improved attack [16] threatens the publication of all future GWAS results, which commonly appear in today's journals on biomedical research.

In response to the above issues, recently there is a surge of research interest in privacy-preserving data publication and analysis. Early proposals use simple methods to anonymize the data, e.g., k-anonymity hides each individual in a group of k indistinguishable individuals [15]. These methods are weak against adversaries with background knowledge, e.g., k-anonymity fails to protect an individual's privacy, when the adversary knows all remaining k–1 individuals in her group. Further, due to the lack of a formal privacy guarantee, whether an algorithm that satisfies k-anonymity indeed protects individuals privacy is often questionable, even without background knowledge. For instance, l-diversity [11] addresses the problem that the adversary can infer sensitive information from a k-anonymous group that lacks diversity (e.g., when all k individuals in the group have the same disease). l-diversity again is also repeatedly challenged and patched, e.g., in [9]. These problems motivate effective solutions that are based on a stronger scheme, with solid, provable privacy guarantees.

Differential privacy [5] is such a scheme, which guarantees that it is hard for the adversary to infer the presence or absence of any individual, even if the adversary knows the exact information all remaining individuals in the dataset. The hardness of the inference is controlled by a parameter ε, called the *privacy budget*. A popular methodology to achieve differential privacy is to inject random noise into the published statistical query results [6]. Here a query is any function with a deterministic output, which includes most common biomedical analysis tasks. Due to a property called composability, answering multiple queries q_1, …, q_n with privacy budget ε_1, …, ε_n respectively satisfy $(\varepsilon_1+\ldots+\varepsilon_n)$-differential privacy.

The main challenge in designing a differentially private solution is to maximize query result accuracy. Although generic solutions exist that can theoretically answer all types of queries, their practical accuracy performance tends to be poor. Hence, previous work has proposed a plethora solutions for different types of queries. At ADSC we have been building the Damson [3] system, which integrates a wide range of differentially private algorithms to perform common biomedical analysis tasks. The contributions of Damson are substantial. First, several of the algorithms used in Damson are novel. Second, different solutions for differential privacy often have very different assumptions and requirements for the data and the queries; hence, integrating them into a general-purpose system is non-trivial. Third, answering multiple queries effectively with high accuracy and low

privacy budget requirement is a challenging problem, which necessitates an effective query optimization engine. Finally, Damson has a high practical value, since it enables biomedical analysis on sensitive data with strong privacy guarantee, which is hard to achieve with previous methods.

This demo focuses on two main aspects of Damson: (i) how Damson performs various types of common biomedical analysis tasks; and (ii) how Damson achieves high accuracy for such tasks with minimal privacy cost. In the following, Sections II and III detail the design of Damson on these two important aspects, respectively.

II. Biomedical Analysis under Differential Privacy

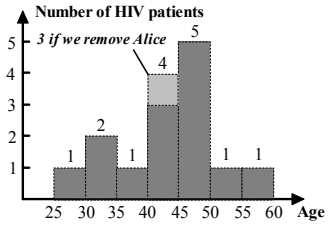
We demonstrate 7 different types of analysis tasks that are currently implemented in Damson: histogram, data cube, marginal, classification, clustering, regression, and ad-hoc selection-count. In the following we detail how Damson handles them effectively with differential privacy guarantees.

A. Histogram

A histogram summarizes data distribution using a set of disjoint bins, each of which contains the count of records that falls into its corresponding attribute range. Fig. 1a shows an example dataset, and Fig. 1b shows a histogram built on the *age* attribute. Both are extracted from Ref. [20]. Each bin in the histogram (e.g., the one for age range 40-45) contains the count of records (4) in the source data. The particular histogram in Fig. 1b is equi-width, meaning that every histogram bin covers the same number of attribute values. Meanwhile, traditionally the most accurate histogram has the finest possible granularity, e.g., each bin covers a single age value. However, as we show [20], the best histogram under differential privacy is often neither equi-width or has the finest granularity, due to the additional noise injected to the published histogram. In particular, merging consecutive bins introduces information loss, but at the same time also reduces the amount of additional noise required by differential privacy.

Name	Age	HIV+
Alice	42	Yes
Bob	31	Yes
Carol	32	Yes
Dave	36	No
Ellen	43	Yes
Helen	42	Yes
Grace	26	Yes
John	44	Yes

(a) Source data



(b) Histogram

Fig. 1: Example histogram (from [20])

A major technical challenge in histogram publication is that in addition to the published count values, the *structure* of the histogram may also leak sensitive information. For instance, in the example of Fig. 1, if we remove the record for Alice, the optimal histogram structure may be different. Damson automatically builds the best histogram under differential privacy, using the dynamic programming algorithm proposed in [20]. This algorithm injects random noise into both the bin counts and the histogram structure, according to differential privacy requirements.

B. Data Cube

Data cube is an important tool for performing OLAP operations on multi-dimensional data. Fig. 2 shows an example count data cube, taken from Ref [4]. The source data (called the *fact table*) contains three attributes, *sex*, *age*, and *salary*. The data cube contains multiple summary tables, called *cuboids*. For instance, Fig. 2b shows the cuboid on the *salary* attribute, where each record contains the count of tuples in the fact table (e.g., 3) that have a particular *salary* value (10-50k). A cuboid can also cover multiple attributes, e.g., the one in Fig. 2d covers both *age* and *salary*, in which every record count the number of fact tuples (e.g., 3) that have a particular *age-salary* combination (*age*:21-30, *salary*:10-50k). Note that the cuboid for *salary* can be computed using the records from the cuboid for *age-salary*, by aggregating all records in the latter that share the same *salary* values. In general, a "coarser" cuboid covering attribute set A can always be obtained from a "finer" cuboid covering a superset of A. The fact table itself can be seen as the finest cuboid, and can be used to derive any other cuboid.

Sex	Age	Salary
F	21-30	10-50k
F	21-30	10-50k
F	31-40	50-200k
F	41-50	500k+
M	21-30	10-50k
M	21-30	50-200k
M	31-40	50-200k
M	60+	500k+

(a) Fact Table T

Sex	Age	Salary	c
*	*	0-10k	0
*	*	10-50k	3
*	*	50-200k	3
*	*	...	...

(b) Cuboid {Salary}

Sex	Age	Salary	c
F	21-30	0-10k	0
F	21-30	10-50k	2
...	...	...	...

(c) Cuboid {Sex, Age, Salary}

Sex	Age	Salary	c
*	21-30	0-10k	0
*	21-30	10-50k	3
*	21-30	50-200k	1
*	21-30	200-500k	0
*	21-30	500k+	0
*	31-40	0-10k	0
*	31-40	10-50k	0
*	31-40	50-200k	2
*	31-40	200-500k	0
*	31-40	500k+	0
*	...	...	...

(d) Cuboid {Age, Salary}

Fig. 2: Example data cube (from [4])

There are two main challenges in publishing a data cube under differential privacy. The first is which cuboids to publish. Publishing a cuboid requires a portion of the privacy budget, which renders other cuboids more noisy (and less accurate). On the other hand, if we choose not to publish a cuboid C, but derive it from a finer one C', the accuracy of C tends to be poor due to noise accumulation. For instance, if we omit the cuboid for *salary* and instead compute it from the cuboid form *age-salary*, counts in the former are less accurate than those the latter, since each count in the *age* cuboid is computed by summing up multiple records in the *age-salary* cuboid, accumulating the noise therein. The second major challenge concerns the *consistency* between correlated cuboids. For example, suppose we publish both the *age* and *age-salary* cuboids. Then, since independent noises are added to the two cuboids, a record in the *age* cuboid is different from the sum of corresponding records in *age-salary*, leading to inconsistency. Damson addresses both problems with elegant

solutions; its data cube module automatically chooses a set of cuboids to publish that maximizes overall accuracy, and enforces consistency among different cuboids.

C. Marginal and Naive Bayes Classification

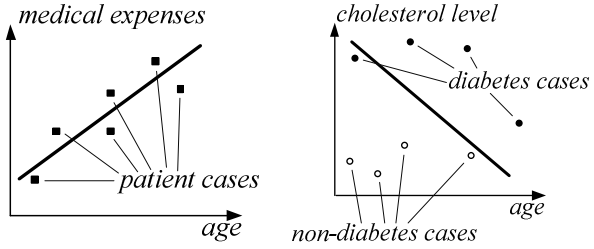
A marginal is equivalent to a cuboid in the data cube, i.e., it is a table that summarizes the number of records in the fact table for each possible attribute combination in a given attribute set. The main difference between the data cube and marginals lies in that a data cube releases the information in all cuboids, whereas in marginal publishing, one is often given a set of attribute combinations, and publishes only the corresponding marginals. For example, in Fig. 2, we may want to publish the marginals for *age-sex* and *age-salary*, respectively. The challenges in data cube publication also exist in marginal publication under differential privacy, i.e., choosing the optimal set of marginal to publish and enforcing their consistency. However, these issues are less important in marginal publication, as one usually publish only a small number of marginals, rather than all cuboids as in the case of data cube.

Meanwhile, the set of marginals to publish is often based on a more complex analysis task. Naive Bayes classification is such a task, which aims to predict the value of one attribute (called the *target attribute*) using the remaining attributes (called *feature attributes*). Specifically, Naive Bayes classification consists of two stages, the *training* stage and the *testing* stage. During training, Naïve Bayes computes a probabilistic model that captures the correlations between the target attribute and each of the feature attribute. Different feature attributes may also exhibit correlations, but Naive Bayes ignores such correlations for simplicity (thus the name Naive Bayes). The model obtained from training is subsequently used in the testing stage to compute the likelihood of each possible value of the target attribute based on the features, and to choose the value with the maximum likelihood. To enable Naive Bayes analysis, we need to publish m+1 marginals, where m is the number of feature attributes. These include a marginal for the target attribute, and m marginals each for the combination of the target and one of the feature attributes. For example, if *salary* is the target attribute and *sex* and *age* are features, Naive Bayes needs 3 marginals, for *salary*, *sex-salary*, and *age-salary*, respectively.

In marginal publication, a common goal is to minimize the overall *relative error*, rather than the overall absolute error [17]. To do this requires adding a smaller amount of noise to smaller counts, and vice versa. Under differential privacy, this is essentially a privacy budget allocation problem, since injecting a smaller amount of noise requires a larger portion of the privacy budget. The main complication is that this budget allocation procedure itself must also be performed in a differentially private way. Damson solves this problem using the iterative solution proposed in [17], which effectively and efficiently finds the best budget allocation that minimizes overall relative error of the published marginals, while satisfying differential privacy requirements.

D. Regression

Regression analysis is particularly useful in medical studies, e.g., to predict the outcome of a patient using various risk factors such as age, body weight, medical history, etc. Currently, Damson supports two common types of regression analysis: *linear regression* and *logistic regression*. The former attempts to identify a linear correlation between different attributes, whereas the latter resembles a linear classifier. Fig. 3a illustrates an example of linear regression with two attributes: *age* and *medical expenses*, where patients cases (points in the plot) surrounds a straight line. The goal of linear regression is to find such a line with minimum overall distances to the points. Fig. 3b shows an example of logistic regression, which aims to distinguish diabetes patients with others. There are two feature attributes, *age* and *cholesterol level*. Logistic regression identifies a line in the feature space (i.e., *age-cholesteroal* level plane) such that (i) most diabetes patients lie on one side of the line, and most non-diabetes patients lie on the other side of the line; (ii) among all patients on the "diabetes" (resp. "non-diabetes") side of the line, the farther away the patient is from the line, the more likely that the patient has diabetes (resp. does not have diabetes).



(a) Linear regression (b) Logistic regression

Fig. 3: Example of regression analysis (from [23])

Performing regression under differential privacy is inherently more complicated than simple counts, due to the difficulty in determining the amount of necessary noise to satisfy the privacy guarantees [1][23]. In particular, both linear and logistic regression involve solving an optimization program. The optimal solution for logistic regression's optimization program does not have a closed-form mathematical expression, and can only be found numerically. Consequently, sensitivity analysis is hard. Although linear regression has a closed-form optimal solution, direct sensitivity analysis on this solution lead to prohibitively high sensitivity, and, thus, unacceptably high noise level.

Damson addresses the above problem using the Function Mechanism [23], which injects random noise on the objective function of the optimization program, rather than its solution. As shown in [23], using a reasonable number of training samples, the Functional Mechanism, and thus Damson, achieve very high accuracy in the regression analysis with low privacy budget consumption.

E. Clustering

Given a set of data points, clustering aims at finding data clusters that are not pre-defined, such that similar points reside in the same cluster, and dissimilar points fall in different

clusters. Damson implements a simple and yet power clustering technique: *k*-means clustering, which is commonly used in a plethora of analysis tasks. Specifically, *k*-means clustering takes a parameter *k*, and finds *k* points (called *centers*) that minimize the total distance between every data point to its nearest center. Each center *c* then forms a cluster, which contains points with *c* as their nearest center. Fig. 4 shows an example of clustering analysis, where the data are clustered based on three centers. Note that a center can be any point in space, and is not necessarily one of the data points. The three lines in the figure shows the boundary of different clusterings.

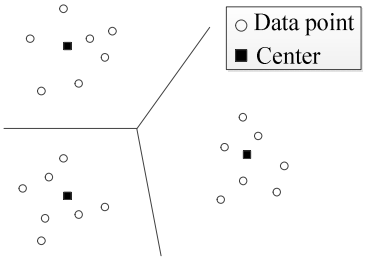


Fig. 4: Example of clustering

Finding the optimal *k*-means solution is a well-known NP-hard problem. Meanwhile, the optimal *k*-means also has a high sensitivity, and, thus, requires a large amount of perturbation which would render the results almost totally random. The current implementation in Damson for differentially private *k*-means clustering uses the iterative algorithm proposed in PINQ [12]. Although this algorithm does not provide any quality guarantee, its practical performance is often good. Work is underway to improve the *k*-means Damson's clustering module, using novel algorithms with quality assurance of the results.

F. Ad-Hoc Selection-Count

Each of the above analyses addresses a specific analysis task that is well-defined and commonly used in the past. Meanwhile, these tasks perform on the entire dataset. In practice, sometimes researchers need to perform an ad-hoc analysis on a subset of the data. Damson handles a large class of such analysis, namely range-count queries. Such queries count of records that fall into a user-specified attribute range. As a real-world example, before carrying out a clinical trial, a pharmaceutical company may want to find out whether there are sufficient patients in a region (e.g., Singapore) that satisfy certain medical requirements, e.g., aged 40-50 with a systolic blood pressure between 120-140. Note that unlike histograms and marginals that have pre-defined ranges, here a query can have an arbitrary range selection, on any combination of attributes.

The solution built into Damson is based on the DP-tree structure [14], which effectively and efficiently handles range count queries of arbitrary dimensionality. The DP-tree improves over our previous proposal Privlet [19][18] by obtaining higher accuracy with the same amount of privacy budget, especially for queries with higher dimensionality. Currently, we are extending the capability of Damson to handle queries with aggregates other than count, and selection criteria besides ranges.

III. Query Optimization in Damson

Traditionally, query optimization mainly aims to save computation time. In a system that complies with differential privacy, however, query optimization must also optimize the accuracy of the analysis results, and the privacy budget usage. Damson incorporates two major techniques for this purpose, which targets a batch of queries and the minimization of the relative error, respectively.

A. Batch Query Processing

It is first shown in [8] that answering a batch of linear counting queries with a good *strategy* can obtain significantly higher overall accuracy than answering them individually. The strategy here is to process a different set of queries, and combine their results to answer the original ones. The specific solution in [8], however, is mainly of theoretical interest, and cannot handle even medium-sized datasets. Damson addresses this problem with the novel approach proposed in [22], which is efficient, scalable, and effective in identifying optimal strategies. The main idea of this approach is to find strategy queries that can answer the input ones approximately, rather than strictly. The approximation error is controllable, and it is usually negligible compared to the amount of noise injected into the query results. This demo will show the strategy queries computed by Damson, and how these queries can be combined to answer the original ones with less error.

B. Relative Error Minimization

Most existing solutions that satisfy differential privacy aim at minimizing the absolute error of query results. However, in many applications, including marginal publications discussed in Section II-C, minimizing the overall relative error is more meaningful. Intuitively, the utility of queries with a small result is more sensitive to added noise. Damson incorporates the iReduct technique [17] to achieve relative error minimization. Besides applications in the computation of differentially private marginals, Damson can also minimize the total relative error of a batch of ad-hoc queries, by carefully allocating the privacy budget to each of them. We will demonstrate how Damson computes the optimal budget allocation that obtains minimum overall relative error for the input queries.

IV. Conclusion

Damson helps researchers perform common analysis on sensitive data, without violating the privacy of the individuals involved in the data. We will demonstrate that Damson supports a large variety of analysis tasks, which range from simpler tasks such as counts, histograms and marginals, to complex ones such as regression and clustering. Meanwhile, we will show that Damson does all these with high result accuracy, and low privacy costs and computational overhead. Given the strengths of the system, we expect Damson to find widespread applications, especially in the biomedical domain.

REFERENCES

[1] K. Chaudhuri and C. Monteleoni. Privacy-Preserving Logistic Regression. *NIPS*, 2008.

[2] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially Private Empirical Risk Minimization. *Journal of Machine Learning Research*, 12:1069-1109, 2011.

[3] Damson. http://differentialprivacy.weebly.com/

[4] B. Ding, M. Winslett, J. Han, and Z. Li, Differentially Private Data Cubes: Optimizing Noise Sources and Consistency. *ACM SIGMOD*, 2011.

[5] C. Dwork. Differential Privacy. *ICALP*, 2006.

[6] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *TCC*, 2006.

[7] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig. Resolving Individuals Contributing Trace Amounts of DNA to Highly Complex Mixtures Using High-Density SNP Genotyping Microarrays. *PLoS Genetics*, 4(8), 2008.

[8] C. Li, M. Hay, V. Rastogi, G. Miklau, A. McGregor. Optimizing Linear Counting Queries under Differential privacy. *PODS*, 2010.

[9] N. Li, T. Li, S. Venkatasubramanian. *t*-Closeness: Privacy Beyond *k*-Anonymity and *l*-Diversity. *IEEE ICDE*, 2007.

[10] Y. Li, Z. Zhang, M. Winslett, Y. Yang. Compressive Mechanism: Utilizing Sparse Representation in Differential Privacy. *WPES*, 2011.

[11] A. Machanavajjhala, J. Gehrke, D. Kifer. *l*-Diversity: Privacy Beyond *k*-Anonymity. *IEEE ICDE*, 2006.

[12] F. McSherry. Privacy Integrated Queries. *ACM SIGMOD*, 2009.

[13] A. Narayanan, V. Shmatikov. Robust De-anonymization of Large Sparse Datasets. *IEEE Symposium on Security and Privacy*, 2008.

[14] S. Peng, Y. Yang, Z. Zhang, M. Winslett and Y. Yu. DP-Tree: Indexing Multi-Dimensional Data under Differential Privacy. *ACM SIGMOD*, 2012, poster.

[15] L. Sweeney. *k*-Anonymity: A Model for Protecting Privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems*. 10(5): 557-570, 2002.

[16] R. Wang, Y. Li, X. Wang, H. Tang, and X. Zhou. Learning Your Identity and Disease from Research Papers: Information Leaks In Genome Wide Association Study. *ACM CCS*, 2009.

[17] X. Xiao, G. Bender, M. Hay, and J. Gehrke. iReduct: Differential Privacy with Reduced Relative Errors. *ACM SIGMOD*, 2011.

[18] X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. *IEEE ICDE*, 2010.

[19] X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. *IEEE TKDE*, 23(8):1200-1214, 2011.

[20] J. Xu, Z. Zhang, X. Xiao, Y. Yang, and G. Yu. Differentially Private Histogram Publication. *IEEE ICDE*, 2012.

[21] Y. Yang, Z. Zhang, G. Miklau, M. Winslett and X. Xiao. Differential Privacy in Data Publication and Analysis. *ACM SIGMOD*, 2012, tutorial.

[22] G. Yuan, Z. Zhang, M. Winslett, X. Xiao, Y. Yang and Z. Hao. Low-Rank Mechanism: Optimizing Batch Queries under Differential Privacy. *PVLDB*, vol. 5, 2012.

[23] J. Zhang, Z. Zhang, X. Xiao, Y. Yang and M. Winslett. Functional Mechanism: Regression Analysis under Differential Privacy. *PVLDB*, vol. 5, 2012.