Dual Query: Practical Private Query Release for High Dimensional Data

Marco Gaboardi Emilio Jesús Gallego Arias
Justin Hsu Aaron Roth Zhiwei Steven Wu

January 23, 2015

Abstract

We present a practical, differentially private algorithm for answering a large number of queries on high dimensional datasets. Like all algorithms for this task, ours necessarily has worst-case complexity exponential in the dimension of the data. However, our algorithm packages the computationally hard step into a concisely defined integer program, which can be solved non-privately using standard solvers. We prove accuracy and privacy theorems for our algorithm, and then demonstrate experimentally that our algorithm performs well in practice. For example, our algorithm can efficiently and accurately answer millions of queries on the Netflix dataset, which has over 17,000 attributes; this is an improvement on the state of the art by multiple orders of magnitude.¹

1 Introduction

Privacy is becoming a paramount concern for machine learning and data analysis tasks, which often operate on personal data. For just one example of the tension between machine learning and data privacy, Netflix released an anonymized dataset of user movie ratings for teams competing to develop an improved recommendation mechanism. The competition was a great success (the winning team improved on the existing recommendation system by more than 10%), but the ad hoc anonymization was not as successful: Narayanan and Shmatikov [11] were later able to re-identify individuals in the dataset, leading to a lawsuit and the cancellation of subsequent competitions.

Differentially private query release is an attempt to solve this problem. Differential privacy is a strong formal privacy guarantee (that, among other things, provably prevents re-identification attacks), and the problem of query release is to release accurate answers to a set of statistical queries. As observed early on by Blum et al. [2], performing private query release is sufficient to simulate any learning algorithm in the statistical query model of Kearns [10].

Since then, the query release problem has been extensively studied in the differential privacy literature. While simple perturbation can be used to privately answer a small number of queries [5], more sophisticated approaches can accurately answer nearly exponentially many queries in the size of the private database [1, 3, 4, 12, 8, 7, 9]. A natural approach, employed by many of these algorithms, is to answer queries by generating synthetic data: a safe version of the dataset that approximates the real dataset on every statistical query of interest.

¹ This is an extended abstract of the full version of this paper [6], which contains full details of our algorithm and experiments.
Unfortunately, even the most efficient approaches for query release have a per-query running time linear in the size of the data universe, which is exponential in the dimension of the data [8]. Moreover, this running time is necessary in the worst case [13], especially if the algorithm produces synthetic data [14].

This exponential runtime has hampered practical evaluation of query release algorithms. One notable exception is due to Hardt et al. [9], who perform a thorough experimental evaluation of one such algorithm, which they called MWEM. They find that MWEM has quite good accuracy in practice and scales to higher dimensional data than suggested by a theoretical (worst-case) analysis. Nevertheless, running time remains a problem, and the approach does not seem to scale to high dimensional data (with more than 30 or so attributes for general queries, and more when the queries satisfy special structure\(^2\)). The critical bottleneck is the size of the state maintained by the algorithm: MWEM, like many query release algorithms, needs to manipulate an object that has size linear in the size of the data universe (i.e., exponential in the dimension). This quickly becomes impractical as the record space grows more complex.

We present DualQuery, an alternative algorithm which is dual to MWEM in a sense that we will make precise. Rather than manipulating an object of exponential size, DualQuery solves a concisely represented (but NP-hard) optimization problem. Critically, the optimization step does not require a solution that is private or exact, so it can be handled by existing, highly optimized solvers. Except for this step, all parts of our algorithm are extremely efficient. As a result, DualQuery requires (worst-case) space and (in practice) time only linear in the number of queries of interest, which is often significantly smaller than the number of possible records. Like existing algorithms for query release, DualQuery has a provable accuracy guarantee and satisfies the strong differential privacy guarantee.

We evaluate DualQuery on a variety of datasets in the following table by releasing 3-way marginals (also known as conjunctions or contingency tables), demonstrating that it solves the query release problem accurately and efficiently even when the data includes hundreds of thousands of features. We know of no other algorithm to perform accurate, private query release for rich classes of queries on real data with more than even 100 features.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size</th>
<th>Attributes</th>
<th>Binary attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adult</td>
<td>30162</td>
<td>14</td>
<td>235</td>
</tr>
<tr>
<td>KDD99</td>
<td>494021</td>
<td>41</td>
<td>396</td>
</tr>
<tr>
<td>Netflix</td>
<td>480189</td>
<td>17,770</td>
<td>17,770</td>
</tr>
</tbody>
</table>

Figure 1: Test Datasets

References


---

\(^2\)Hardt et al. [9] are able to scale up to 1000 features on synthetic data when the features are partitioned into a number of small buckets, and the queries are chosen to never depend on features in more than one bucket.


