Real-Time Aggregate Monitoring with Differential Privacy

Liyue Fan
Math&CS Department
Emory University, Atlanta, GA
liyue.fan@emory.edu

Li Xiong
Math&CS Department
Emory University, Atlanta, GA
lxiong@mathcs.emory.edu

ABSTRACT

Sharing real-time aggregate statistics of private data has given much benefit to the public to perform data mining for understanding important phenomena, such as Influenza outbreaks and traffic congestion. However, releasing time-series data with standard differential privacy mechanism has limited utility due to high correlation between data values. We propose FAST, an adaptive system to release real-time aggregate statistics under differential privacy with improved utility. To minimize overall privacy cost, FAST adaptively samples long time-series according to detected data dynamics. To improve the accuracy of data release per time stamp, filtering is used to predict data values at non-sampling points and to estimate true values from noisy observations at sampling points. Our experiments with three real data sets confirm that FAST improves the accuracy of time-series release and has excellent performance even under very small privacy cost.

Categories and Subject Descriptors

H.2.8 [**DATABASE MANAGEMENT**]: Database Applications—*Data Mining*; G.3 [**PROBABILITY AND STATISTICS**]: Time series analysis

Keywords

Differential Privacy, Estimation, Sampling, Time Series

1. INTRODUCTION

Sharing real-time aggregate statistics of private data enables many importatnt data mining applications. Consider examples below:

Disease Surveillance: A health care provider gathers data from individual visitors. The collected data, e.g. daily number of Influenza cases, is then shared with third parties, for instance, researchers, in order to monitor and to detect seasonal epidemic outbreaks at the earliest.

Traffic Monitoring: A GPS service provider gathers data from individual users about their locations, speeds, mobility, etc. The aggregated data, for instance, the number of users at each region during each time period, can be mined for commercial interest, such as popular places, as well as public interest, such as congestion patterns on the roads.

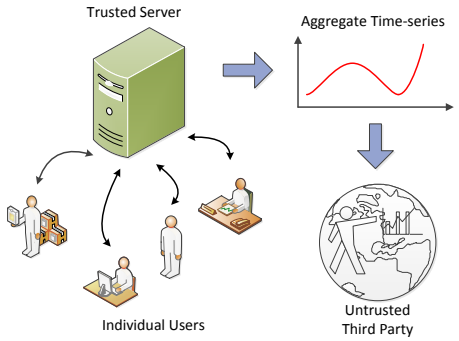


Figure 1: Aggregate Data Sharing Scenario

A common scenario of such applications can be summarized by Figure 1, where a trusted server gathers data from a large number of individual subscribers. The collected data may be then aggregated and continuously shared with other un-trusted entities for various purposes. The trusted server, i.e. publisher, is assumed to be bound by contractual obligations to protect the user's interests, therefore it must ensure that releasing the data does not compromise the privacy of any individual who contributed data. The goal of our work is to enable the publisher to share useful aggregate statistics over individual users continuously (aggregate time series) while guaranteeing their privacy.

The current state-of-the-art paradigm for privacy-preserving data publishing is *differential privacy*, which requires that the aggregate statistics reported by a data publisher be perturbed by a randomized algorithm $\mathcal{A}$, so that the output of $\mathcal{A}$ remains roughly the same even if any single tuple in the input data is arbitrarily modified. Given the output of $\mathcal{A}$, an adversary will not be able to infer much about any single tuple in the input, and thus privacy is protected.

Most existing works on differentially private data release deal with one-time release of static data [4, 7, 13, 14, 15, 16]. In the applications we consider, data values at successive timestamps are highly correlated. A straightforward application of differential privacy mechanism which adds a Laplace noise to each aggregate value at each time stamp can lead to a very high overall perturbation error due to the composition theorem [11]. Few recent works [3, 5, 12] studied the problem of releasing time series or continual statistics . Rastogi and Nath [12] proposed an algorithm which perturbs k Discrete Fourier Transform (DFT) coefficients of the entire time series and reconstructs a released version from the Inverse DFT. Since the entire time-series is required to perform those operations, it is not applicable to real-time applications. We included this method in the empirical study for utility comparison. Dwork et al. [5] proposed a differentially private continual counter over

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

a binary stream with a bounded error at each time step. Chan et al. [3] studied the same problem and concluded with a similar upper bound. However, both works adopt an event-level privacy model, with the perturbation mechanism designed to protect the presence of an individual event, i.e. a user's contribution to the data stream at a single time point, rather than the presence or privacy of a user.

In this paper, we propose FAST, a novel approach with Filtering and Adaptive Sampling for releasing Time series under differential privacy. It uses sampling to query and perturb selected values in the time series with the differential privacy mechanism, and simultaneously uses filtering to dynamically predict the non-sampled values and correct the sampled values. To improve the accuracy of data release at each time stamp, we propose a state space model and use of the Kalman filter [8].To minimize the overall privacy cost, hence, the overall perturbation error, we propose an adaptive sampling algorithm with PID control. We study the accuracy and robustness of FAST with real world time-series data sets, which confirms that FAST provides accurate results in real-time and stability despite different data dynamics.

The rest of the paper is organized as follows: Section 2 provides the problem formulation and the background for differential privacy. Section 3 presents an overview of our solution, as well as the technical details of filtering and adaptive sampling. Section 4 presents a set of empirical results. Section 5 concludes the paper and states possible directions for future work.

2. PRELIMINARIES

2.1 Problem Statement

Formally a **time series** is defined as follows:

Definition 1. [Aggregate Time Series] *A univariate, discrete time series* $\mathbf{X} = \{x_k\}$ *is a set of values of a variable* x *observed at discrete time* k*, with* $0 \leq k < T$*, where* T *is the length of the series.*

In our example applications, $\mathbf{X}$ is an aggregate $count$ series, such as, the daily number of patients diagnosed of Influenza, or the hourly count of drivers passing by a gas station. This assumption will hold true for the rest of the paper.

We measure the quality of a released time series $\mathbf{R} = \{r_k\}$ by **average relative error** E:

$$E = \frac{1}{T}\sum_{k=0}^{T-1} |r_k - x_k| / max\{x_k, \delta\} \quad (1)$$

where δ is a user-specified constant (also referred to as *sanitary bound* in [13]) to mitigate the effect of excessively small query results. Here we set $\delta = 1$ throughout the entire time-series.

Clearly, the quality of a published series increases as each r_k approaches x_k, the extreme case of which would have $r_k = x_k$ for each k. However, a privacy-preserving algorithm is likely to perturb original data values in order to protect individual privacy. Thus, a publishing mechanism that guarantees user privacy and yields high utility is desired.

2.2 Differential Privacy

A mechanism is differentially private if its outcome is not significantly affected by the removal or addition of a single user. An adversary thus learns approximately the same information about any individual user, irrespective of his/her presence or absence in the original database.

Definition 2. [Differential Privacy [2]] *A non-interactive privacy mechanism* $\mathcal{A}$ *gives* α*-differential privacy if for any dataset* D_1 *and* D_2 *differing on at most one record, and for any possible anonymized dataset* $\widetilde{D} \in Range(\mathcal{A})$*,*

$$Pr[\mathcal{A}(D_1) = \widetilde{D}] \leq e^{\alpha} \times Pr[\mathcal{A}(D_2) = \widetilde{D}] \quad (2)$$

where the probability is taken over the randomness of $\mathcal{A}$*.*

The privacy parameter α, also called the privacy budget [11], specifies the degree of privacy offered. Intuitively, the lower value of α implies stronger privacy guarantee and a higher value implies a weaker guarantee while possibly achieving higher accuracy.

Laplace Mechanism. Dwork et al. [4] show that α-differential privacy can be achieved by adding i.i.d. noise to each query result:

$$\tilde{q}(D) = q(D) + \tilde{N} \quad (3)$$

The magnitude of the noise added conforms to a *Laplace distribution* with the probability density function $p(x|\lambda) = \frac{1}{2\lambda}e^{-|x|/\lambda}$, with $\lambda = GS(q)/\alpha$, where $GS(q)$ denotes the *global sensitivity* [4] of a query q which is defined as the maximum difference between the query results from any two neighboring databases. In our example, $GS(count) = 1$.

Composition. The composition properties of differential privacy provide privacy guarantees for a sequence of computations.

THEOREM 1. [Sequential Composition [11]] *Let* $\mathcal{A}_i$ *each provide* α_i*-differential privacy. A sequence of* $\mathcal{A}_i(D)$ *over the dataset* D *provides* ($\sum_i \alpha_i$)*-differential privacy.*

Given the composition theorem, a baseline solution to sharing differentially private time series can be derived: Laplace perturbation is applied at every time stamp to guarantee α/T-differential privacy, where T is the length of entire series.

3. FAST

We propose FAST: a novel solution to sharing time-series data with differential privacy. It allows for fully automated adaptation to changing data dynamics and highly accurate time-series prediction/estimation. Figure 2 shows the framework of FAST.

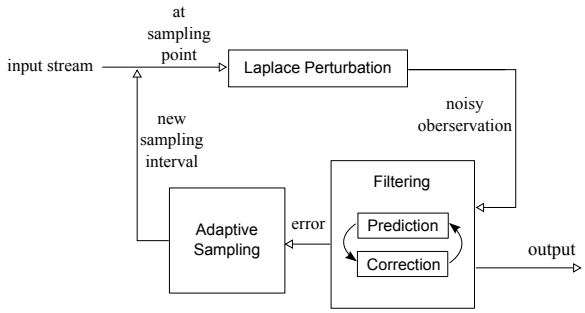


Figure 2: FAST Framework

We outline the workflow of FAST below. For detailed algorithm, we refer the readers to our full study [6].

- For each time stamp k, the **adaptive sampling** component determines whether to sample/query the **input** time-series or not.
- If k is a sampling point, the data value at k is perturbed by the **Laplace mechanism** to guarantee α_0-differential privacy.
- The **filtering** component produces a $prediction$ of data value based on an internal state model at every time stamp. The prediction, i.e. prior estimate, is released to **output** at a non-sampling point, while a posterior estimate, i.e. *correction* of the noisy observation and prediction, is released at a sampling point.
- The error between the prior and the posterior is then fed through the **adaptive sampling** component to adjust the sampling rate. Once the user-specified privacy budget α is used up, the system will stop sampling the input series.

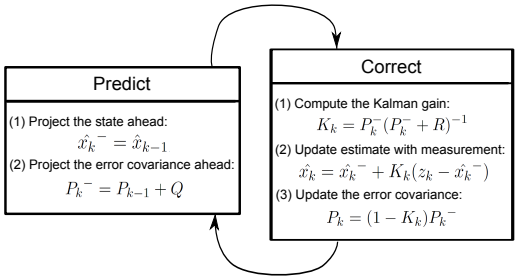


Figure 3: Complete Picture of the Kalman Filter

THEOREM 2. *FAST satisfies α-differential privacy.*

PROOF. *Suppose the maximum number of samples FAST allows for a time series is M. By setting $\alpha_0 = \alpha/M$, the above algorithm satisfies α-differential privacy according to Theorem 1.* □

There are two types of error which we would like to balance in our solution: *perturbation* error by Laplace perturbation mechanism at sampling points and *prediction* error by the filtering prediction procedure at non-sampling points. The more we sample, the more *perturbation* error we introduce, while the *prediction* error might be reduced due to more available feedback, and vice versa. Our goal is to balance the trade-off between the two types of error by adaptively adjusting the sampling rate.

3.1 Filtering

Each released data value is an estimate derived by the filtering component in FAST. To improve the accuracy of each released value, we established a state space model for time series data and proposed to use the Kalman filter for estimation.

Process Model. Let x_k denote the internal state (true value) of a process at time stamp k. The states at consecutive time stamps can be modeled by the following equations:

$$x_{k+1} = x_k + \omega \tag{4}$$

$$p(\omega) \sim N(0, Q) \tag{5}$$

This constant process model indicates that adjacent data values from the original time-series should be consistent except for a white Gaussian noise ω with variance Q .

Measurement Model. The noisy observation, which is perturbed data from the Laplace mechanism, can be modeled by:

$$z_k = x_k + \nu \tag{6}$$

The measurement noise ν is a Laplace noise which follows $p(\nu) \sim Lap(0, \lambda)$ where λ is determined by differential privacy mechanism. In this paper, we approximate the Laplace noise with a white Gaussian error with variance R and define the distribution of ν as

$$p(\nu) \sim N(0, R) \tag{7}$$

We have also studied in [6] several more complex filtering techniques, such as Masreliez filter [10] and particle filters [1], with more precise modeling of the Laplace noise. The results showed that the Kalman filter with the Gaussian measurement noise is sufficient in utility and computationally efficient. For brevity, we only present the Kalman filter model in this paper.

Prior and Posterior Estimates. If noisy observation is available at every time stamp,the Kalman filter will repeat a pair of operations: *Prediction* and *Correction*, for every k. Figure 3 gives a high-level diagram of the two procedures specific to our state space model.

At time k, the *apriori* state estimate $\hat{x}_k^-$ is made based on the process model in Equation (4) and is related to the *aposteriori* state estimate of last step, while the *aposteriori* state estimate $\hat{x}_k$ is based on a linear combination of $\hat{x}_k^-$ and the noisy measurement z_k :

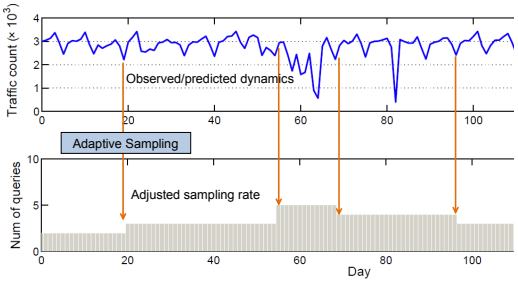


Figure 4: Adaptive Sampling with Traffic Data

$$\hat{x}_k^- = \hat{x}_{k-1} \tag{8}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-). \tag{9}$$

The value K_k, called *Kalman Gain*, is adjusted with each measurement in order to minimize the error variance P_k of the posterior estimate $\hat{x}_k$. We refer interested readers to [6] for a complete set of definitions and detailed derivation.

One advantage of the Kalman filter is that it maintains and updates the best estimate of the internal state by properly weighing and combining all available data (prior estimate and noisy observation) to form an educated guess. Another advantage is the computation efficiency which is crucial to real-time applications, as only $O(1)$ computations are required for each time stamp from Figure 3.

When combined with sampling in the overall solution, a noisy observation, which is needed by the *Correction* step, may not be available at every time stamp. Therefore, we propose to release the prior estimate at non-sampling points instead. We will evaluate the estimation accuracy in the experiment section.

3.2 Sampling

Since each noisy observation from Laplace mechanism comes with a cost (privacy budget spent), we are motivated to sample data values through the differential privacy interface only when needed in our overall solution. Below we briefly introduce two sampling strategies and detailed algorithms can be found in [6].

Fixed Rate Sampling. Given a pre-defined interval I, the fixed-rate algorithm samples the time series periodically and releases the posterior estimate per I timestamps. As for the time points between two adjacent samples, a predicted value is released. Privacy budget $(\alpha I)/T$ will be spent on each sample to guarantee α-differential privacy according to Theorem 2.

The challenge of fixed-rate sampling is to define the interval I. Increasing the sampling rate, i.e. when I is low, an extreme case of which is to issue a query at each time step as in the baseline solution, the overall perturbation error will grow with the number of samples. On the other hand, when we decrease the sampling rate, i.e. when I is high, the perturbation at each sampling point will drop, but the published series will not reflect up-to-date data values, resulting large prediction error. *Apriori* knowledge of the data is required to find the optimal sampling rate in order to minimize the average relative error. However, that is impractical for a real-time publishing scenario.

Adaptive Sampling. With no *apriori* knowledge of the time series, it is desirable to detect data dynamics and to adjust the sampling rate on-the-fly. Figure 4 illustrates the idea of adaptive sampling. We plot the original time-series, *traffic count*, as well as the number of queries (samples) issued by the adaptive sampling mechanism during each corresponding time unit. As is shown, the adaptive sampling mechanism increases sampling rate between day 20 and

day 100, when the traffic count exhibits significant fluctuations, and decreases sampling rate beyond day 100, when there's little variation among data values.

We implement adaptive sampling in FAST with feedback control. It is based on the model error between the posterior and the prior estimates. At time step k_n ($0 \leq k_n < T$), where the subscript indicates the n-th sampling point ($0 \leq n < M$), we define this error E_{k_n} as follows:

$$E_{k_n} = |\hat{x}_{k_n} - \hat{x}^-_{k_n}| / \max\{\hat{x}_{k_n}, \delta\}. \quad (10)$$

Note that no error is defined at a non-sampling point.

The model error measures how well the internal state model describes the current data dynamics, assuming $\hat{x}_{k_n}$ is close to the true value. Since $\hat{x}^-_{k_n}$ is given by a constant state model, we may infer that data is going through rapid changes if the error E_{k_n} increases with time. In response, the controller in our system will detect the error and increase the sampling rate accordingly.

FAST adopts a PID controller, the most common form of feedback control [9], to measure the performance of sampling over time. We re-define the three PID components, *Proportional, Integral*, and *Derivative*, with the model error defined in Equation (10). The full PID algorithm is thus

$$\Delta = C_p E_{k_n} + \frac{C_i}{T_i} \sum_{j=n-T_i+1}^{n} E_{k_j} + C_d \frac{E_{k_n} - E_{k_{n-1}}}{k_n - k_{n-1}} \quad (11)$$

Control gains C_p, C_i, and C_d denote how much each of the *proportional*, *integral*, and *derivative* counts for the final calibrated PID error and T_i represents the integral time. The control gains are constrained by:

$$C_p, C_i, C_d \geq 0 \quad (12)$$

$$C_p + C_i + C_d = 1 \quad (13)$$

Given the PID error Δ, a new sampling interval I' can be determined:

$$I' = I + \theta(1 - e^{\frac{\Delta - \xi}{\xi}}) \quad (14)$$

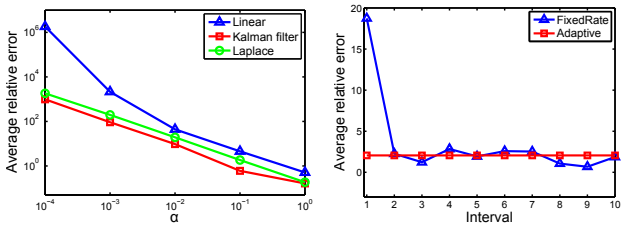
where θ and ξ are pre-determined parameters. Intuitively, the sampling interval increases as the Δ error drops, and vice versa.

4. EXPERIMENT

We have implemented FAST in Java with JSC (Java Statistical Classes[1]) for simulating the Laplace distribution. Our study has been conducted with three real time-series data sets:

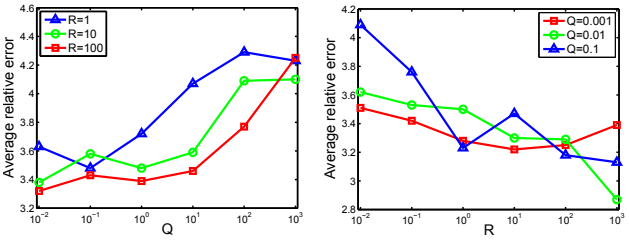
- **Flu** is the weekly surveillance data of Influenza-like illness provided by the Influenza Division of the Centers for Disease Control and Prevention[2]. We collected the weekly outpatient count of the age group [5-24] from 2006 to 2010. This time-series consists of 209 data points.
- **Traffic** is a daily traffic count data set for Seattle-area highway traffic monitoring and control provided by the Intelligent Transportation Systems Research Program at University of Washington[3]. We chose the traffic count at location I-5 143.62 southbound from April 2003 till October 2004. This time-series consists of 540 data points.
- **Unemployment** is the monthly unemployment level of black or African American women of age group [16-19] from ST. Louis Federal Reserve Bank[4]. This data set contains observations from January 1972 to October 2011 with 478 data points.

[1]http://www.jsc.nildram.co.uk
[2]http://www.cdc.gov/flu/
[3]http://www.its.washington.edu/
[4]http://research.stlouisfed.org/



(a) Estimation Comparison (b) Sampling Comparison

Figure 5: Estimation and Sampling Methods with Traffic Data Set



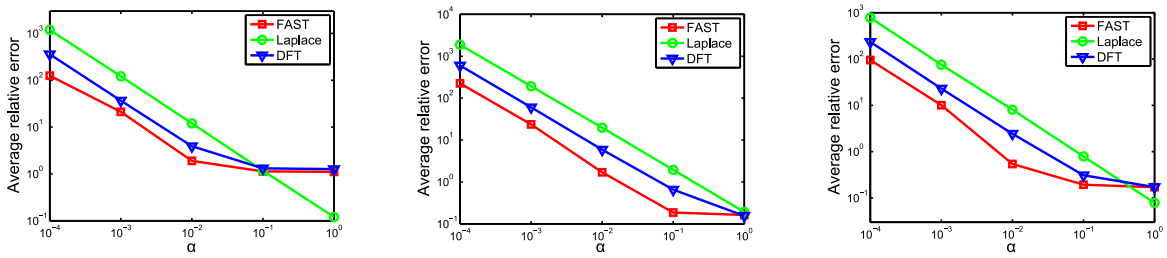
(a) Impact of Q (b) Impact of R

Figure 6: Impact of Noise Parameters with Unemployment Data Set

The default parameter settings are as follows: the overall privacy budget $\alpha = 0.01$, the estimation noise variances $(Q, R) = (0.001, 10)$, the control gains $(C_p, C_i, C_d) = (0.9, 0.1, 0)$, and the interval adaptation parameters $(\theta, \xi) = (10, 1000)$. We will describe how to choose default values and show the impact of individual parameters later in this section.

Accuracy of Filtering and Adaptive Sampling. We first evaluate the accuracy of the Kalman filter estimation and adaptive sampling against alternative methods. Figure 5 shows the results with *traffic* data set, while the other data sets exhibit similar trends. We compare the Kalman filter posterior estimation against linear regression (fitted with 2 data points) and the baseline Laplace perturbation algorithm mentioned in Section 2. As seen in Figure 5(a), the Kalman filter outperforms linear predictor as well as Laplace perturbation algorithm especially when given small privacy budgets($\alpha = 0.0001, 0.001, 0.01$). With large budget ($\alpha = 1$), which we note does not provide sufficient privacy protection, we observed no substantial advantage of using the Kalman filter, which can be explained by the nature of the *aposteriori* estimate defined by Equation (9): it only partially relies on the noisy measurement z_k hence does not fully reflect reduced perturbation error. As for sampling strategies, we plot the utility comparison in Figure 5(b), with the interval for fixed-rate sampling varying from 1 to 10. We found the adaptive sampling strategy is comparable to the optimal fixed-rate with no need of *a priori* knowledge. Thus we believe that adaptive sampling can be adopted by a wider range of applications.

Effects of Kalman Filter Parameters. To understand the impact of process noise variance Q and measurement noise variance R, we vary their values independently and plot the estimation accuracy. The results with the *unemployment* data set are presented in Figure 6, while the other data sets show similar trends. Given R fixed, we observe that the accuracy of estimation drops as Q increases; given Q, the accuracy rises as R increase. This can be interpreted by the definition of the Kalman gain K_k in Figure 3: K_k increases as Q does, resulting the *aposteriori* estimate favoring the noisy observation. Similarly, when increasing R, the Kalman gain decreases, resulting the *aposteriori* estimate favoring the state prediction. Both results confirm that it's beneficial to rely more on the state prior than the noisy measurement when privacy budget is small, which implies larger noise in observed values.



(a) Flu data set (b) Traffic data set (c) Unemployment data set

Figure 7: Comparison against Alternative Methods with Three Data Sets

Effects of Control Parameters. We study the impact of control gains and find that as long as their values are chosen according to the common practice: $proportional > integral > derivative$, the resulting relative error does not differ beside the randomness introduced by the Laplace mechanism. Therefore we conclude that there's no extra "rule of thumb" than the common practice for tuning the control gains in our system.

We also study the impact of T_i, ξ, and θ on the accuracy of the released series. Varying the value of any of them does not result in substantial change in relative error. They are considered not influential and thus can be set by users as needed. Detailed figures are excluded for brevity.

FAST vs. Alternative Approaches. Figure 7 presents the performance of FAST against the baseline Laplace mechanism (which adds a Laplace noise to each aggregate value at each time stamp) and the DFT [12] algorithm (mentioned in Section 1) with respect to different scales of privacy budget. The number of DFT coefficients to preserve is set to be 20, which is near optimal according to [12]. Again, our adaptive approach shows superior performance when the privacy budget α is limited. This confirms our hypothesis that with accurate estimate by the Kalman filter, the PID control mechanism can adjust the sampling rate as needed, thus improving the overall utility of the published series. Note that when α is high and approaching 1, the baseline Laplace perturbation algorithm achieves smaller relative error because of the reduced perturbation error. Since the reconstruction error of the DFT approach and the prediction error of our adaptive approach both outweighs the perturbation error in this case, their released series contain larger relative error. However, a privacy budget greater than 1 does not provide sufficient privacy protection any more. We find that our adaptive approach outperforms the alternative methods under strong privacy guarantee.

5. CONCLUSION

We have proposed FAST, an adaptive approach with filtering and sampling for monitoring real-time aggregate under differential privacy. The key innovation is that our approach utilizes feedback loops based on observed (perturbed) values to dynamically adjust the prediction/estimation model as well as the sampling rate. To minimize the overall privacy cost, FAST uses the PID controller to adaptively sample long time-series according to detected data dynamics. As to improve the accuracy of data release per time stamp, the Kalman filter is used to predict data values at non-sampling points and to estimate true values from perturbed values at sampling points. Our experiments with three real data sets show that it is beneficial to incorporate feedback into both the estimation model and the sampling process. The results confirmed that our adaptive approach improves utility of time-series release and has excellent performance even under very small privacy cost.

As for the future, we plan to expand our solution to enable monitoring of differentially private spatial-temporal statistics, for example, real-time traffic conditions at all intersections of a city.

6. ACKNOWLEDGMENTS

This research was supported in part by NSF grant CNS-1117763, AFOSR grant #12RSE136, and an Emory URC grant.

7. REFERENCES

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2001.

[2] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *STOC*, 2008.

[3] T.-H. H. Chan, E. Shi, and D. Song. Private and continual release of statistics. In S. Abramsky, C. Gavoille, C. Kirchner, F. M. auf der Heide, and P. G. Spirakis, editors, *ICALP (2)*, volume 6199 of *Lecture Notes in Computer Science*, pages 405–417. Springer, 2010.

[4] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *In Proceedings of the 3rd Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[5] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 715–724, New York, NY, USA, 2010. ACM.

[6] L. Fan and L. Xiong. Adaptively sharing time-series with differential privacy. *CoRR*, abs/1202.3461, 2012.

[7] M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endow.*, 3(1-2):1021–1032, Sept. 2010.

[8] R. E. Kalman. A new approach to linear filtering and prediction problems. 1960.

[9] M. King and M. King. *Process Control: A Practical Approach*. John Wiley & Sons, 2011.

[10] C. Masreliez. Approximate non-gaussian filtering with linear state and observation relations. *Automatic Control, IEEE Transactions on*, 20(1):107 – 110, feb 1975.

[11] F. McSherry. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *SIGMOD*, 2009.

[12] V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *SIGMOD*, 2010.

[13] X. Xiao, G. Bender, M. Hay, and J. Gehrke. ireduct: differential privacy with reduced relative errors. In *Proceedings of the 2011 international conference on Management of data*, SIGMOD '11, pages 229–240, New York, NY, USA, 2011. ACM.

[14] X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. *Knowledge and Data Engineering, IEEE Transactions on*, 23(8):1200 –1214, aug. 2011.

[15] Y. Xiao, L. Xiong, and C. Yuan. Differentially private data release through multidimensional partitioning. In W. Jonker and M. Petkovic, editors, *Secure Data Management*, volume 6358 of *Lecture Notes in Computer Science*, pages 150–168. Springer Berlin / Heidelberg, 2010.

[16] J. Xu, Z. Zhang, X. Xiao, Y. Yang, and G. Yu. Differentially private histogram publication. In *ICDE*, 2012.