Vol.2014-MPS-98 No.25
Vol.2014-BIO-38 No.25
2014/6/27

Towards HMM Parameter Estimation with Differential Privacy

Nut Sornchumni[1,a)] Kenji Hashimoto[2,b)] Hiroyuki Seki[2,c)]

Abstract: In this work, we study an application of differential privacy to stateful data mining, specifically the parameter estimation of hidden Markov models (HMM). In the differential privacy framework, the computation on two similar datasets is required not to be significantly different from each other, that is, the computation should be insensitive to the presence or absence of a single individual data record, thus, preserving the privacy. In this study, we investigate an HMM parameter estimation algorithm that makes the counts calculated by the algorithm differentially private. Finally, we will show the performance evaluation and the trade-off between privacy protection and model precision.

Keywords: Hidden Markov Model, Estimation Maximization, Differential Privacy, Privacy Preserving

1. Introduction

Privacy concerning issues have been recently emerging as an active discussion [12] in security research field in addition to system protection using conventional cryptographic approaches. As a data-miner, one would want to derive useful information or trend that can improve the system, community and even human world. On the contrary, individuals who provided the data may concern about the usage of the data and the information leakage.

Conventional approaches to protecting the privacy of the individuals are anonymization or de-identification, which are proven to be insufficient [10]. Differential privacy [3] is a recently proposed mathematical model for adequately defining the privacy of the individuals participating in a statistical database. The framework, as opposed to traditional cryptographic definition, captures increased risk of the private data leakage from a specific database. That is to say, the framework measures relative information leakage by computation on data in the database.

Although differential privacy framework was just recently proposed, the strength of the framework made it actively discussed and evolving continuously [4,8,11] including integration into big data processing such as the map-reduce framework.

Many investigations are made regarding data mining under the framework [1,6,7]. However, most of the discussions investigated stateless algorithms, such as k-means and decision tree classifier.

Hidden Markov model(HMM) is one of the powerful modeling tools that can be used to analyze and build a model that resembles the targeted system. HMM excels in the application of temporal pattern recognition such as speech, handwriting and gesture recognition. By learning and extracting information from the sample data, accurate prediction or classification model can be built. However, sensitive information can be leaked through the model. In order to mitigate the risk, privacy protection is required to be applied.

In this paper, we will introduce $\epsilon-$differential privacy to hidden Markov model using differential privacy framework. We will propose a parameter estimation algorithm with $\epsilon-$differential privacy. Finally, we will show the experiment results on the proposed algorithm and evaluate the trade-offs between privacy restriction and model accuracy.

2. Differential Privacy

Differential privacy [3] is another definition of privacy that guarantees the outcome of computations from datasets to be insensitive. To be insensitive means that the results of computations should not be significantly different for two similar datasets, specifically, any two datasets that only differs by one element.

2.1 Definition

Differential privacy requires that the difference of computations on two similar datasets be restricted by a factor $exp(\epsilon)$.

Definition 1. *A randomized function κ gives $\epsilon-$differential privacy if for all datasets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\kappa)$,*

$$Pr[\kappa(D_1) \in S] \leq exp(\epsilon) \times Pr[\kappa(D_2) \in S].$$

The ϵ in the formula can be adjusted to meet specific privacy requirement, that is, lowering ϵ means that the information on whether a single record is present or absent will reduce. We will call ϵ the privacy parameter.

The ϵ-differentially private mechanism has two important properties, compatibilities with sequential composition and parallel composition [9]. Sequential composition allows consecutive computations be differentially private and ϵ of each computation will be accumulated, for example, ϵ-differential privacy with

[1] Nara Institute of Science and Technology
[2] Nagoya University
[a)] nut-s@is.naist.jp
[b)] k-hasimt@is.nagoya-u.ac.jp
[c)] seki@is.nagoya-u.ac.jp

$\epsilon = 2$ can be guaranteed for two consecutive computations that guarantee ϵ-differentially private with $\epsilon = 1$. Parallel composition of computation that process disjoint subsets of the dataset can guarantee ϵ−differential privacy with ϵ being the maximum among the privacy parameters of all the parallel computations.

2.2 Sensitivity

Sensitivity measures the difference of an outcome from function f when a single element in the dataset be inserted or removed.

Definition 2. *For $f : \mathcal{D} \to \mathcal{R}^d$, the L1-sensitivity of f is*

$$S(f) = \Delta f = \max_{D_1, D_2} \| f(D_1) - f(D_2) \|_1$$

for all D_1, D_2 differing in at most one element where $\| x \|_1$ is the L1-norm of x.

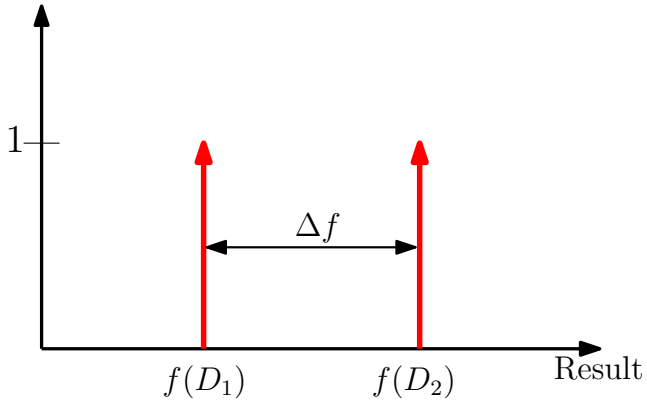

Fig. 1 Sensitivity of f

2.3 Laplace Mechanism

To achieve ϵ−differential privacy, a random noise is drawn from Laplace probability distribution [5] ($Lap(\delta) = (1/2\delta)exp(- \mid x \mid /\delta)$) and is applied to the computation result. This is called *Laplace mechanism*. The scale of Laplace distribution is determined by the sensitivity of the function f and the parameter ϵ.

Theorem 1. *For all $f : D^n \to R^d$, the following mechanism provides ϵ-differential privacy*

$$M(x) = f(x) + Lap(S(f)/\epsilon)$$

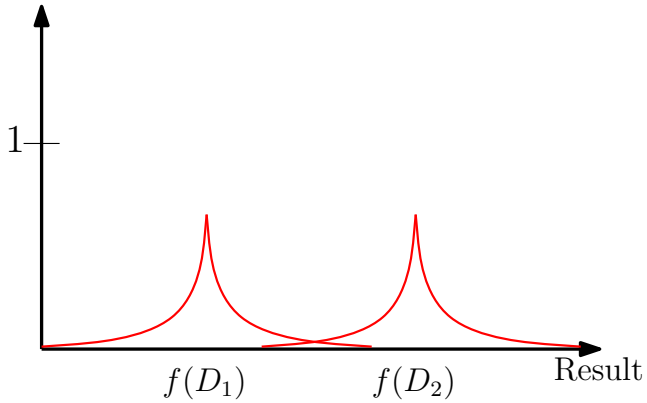

Fig. 2 Outcomes of f when perturbed by Laplacian noise.

3. Hidden Markov Model

3.1 Definition

Hidden Markov Model(HMM) is a statistical model that is widely used especially in temporal pattern recognition applications. A system being modeled is assumed to be a Markov process which consist of hidden states and observable symbols.

An HMM is a tuple $A = (S, \Gamma, \langle a_{kl} \rangle_{1 \le k,l \le n}, \langle e_k(b) \rangle_{1 \le k \le n, b \in \Gamma})$ where $S = \{1, 2, ..., n\}$ is a set of states, Γ is a set of observable symbols, a_{kl} is a transition probability from state k to state l and $e_k(b)$ is a probability of emitting $b \in \Gamma$ in state k.

For a given sequence of states $\pi = \pi_1 \pi_2 \cdots \pi_L (\pi_i \in S, 1 \le i \le L)$, A defines the probability of π and the probability of an observation sequence $x = x_1 x_2 \cdots x_L$ by

$$a_{kl} = P(\pi_i = l \mid \pi_{i-1} = k) \quad (1)$$

$$e_k(b) = P(x_i = b \mid \pi_i = k) \quad (2)$$

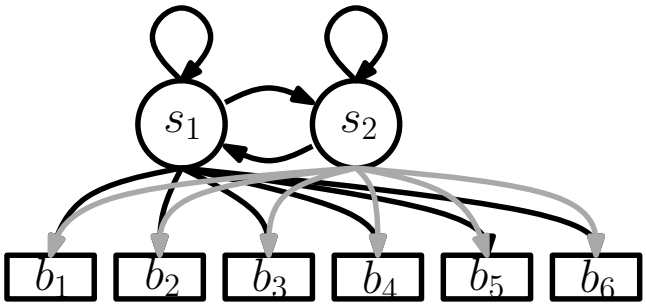

Fig. 3 HMM with 2 hidden states and 6 observable symbols.

State	Symbol	Emission prob.
s_1	b_1	1/6
	b_2	1/6
	b_3	1/6
	b_4	1/6
	b_5	1/6
	b_6	1/6
s_2	b_1	1/10
	b_2	1/10
	b_3	1/10
	b_4	1/10
	b_5	1/10
	b_6	1/5

Table 1 Example of emission probability in HMM.

3.2 Parameters Estimation

When building (or training) an HMM, many factors, such as model structure, initial probability distribution of a_{kl} and $e_k(b)$ should be thoroughly considered as it will impact the precision of the training. In this paper, we do not focus on the method to maximize HMM precision, but only on the effect of the noise onto HMM.

The training of HMM [2] can be done by two types of training datasets. One is the observable sequence with respective hidden state transition. By having both hidden state transition path and observable sequence as a training data, parameters of the model can be estimated effectively and easily by using maximum likelihood estimator on the training sequence,

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \tag{3}$$

State transition probability of state $k \to l$ can be calculated by counting transitions from state $k \to l$ divided by number of transitions from state $k \to l'$ where l' is state other than l.

$$e_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')} \tag{4}$$

In the same way, emission probability of symbol b from state k can be calculated by counting number of symbol b that has been emitted from state k, divided by number of b', the symbol other than b that has been emitted by state k.

In another situation when the training dataset consist only of observable sequences, state transition path can only be guessed. By using the initial randomize (or provided manually) parameters and training sequences, the forward-backword algorithm, is used to derive the most probable state path. After that, maximum likelihood estimator will be used to derive new parameters of the model and iterate until the stopping condition is met, this is called Baum−Welch algorithm, which is a special case of the Expectation Maximization(EM) algorithm.

Let $\theta = (a_{11}, \ldots, a_{ss}, e_1(b_1), \ldots, e_s(b_n))$ be a parameter vector of an HMM. Let $X = \{x^1, \ldots, x^D\}$ be a set of D observation sequences where $x^j = x_1^j \cdots x_L^j$ for each x^j. The forward probability of state k at position i, $f_k^j(i)$, is,

$$f_k^j(i) = P(x_1^j \cdots x_i^j, \pi = k)$$

and the backward probability of state k at position i, $b_k^j(i)$ is,

$$b_k^j(i) = P(x_{i+1}^j \cdots x_L^j, \pi_i = k)$$

The model training uses the following probability:

$$\begin{aligned}\xi_i^j(k,l) &= P(\pi_i = k, \pi_{i_1} = l \mid x^j, \theta) \\ &= \frac{f_k^j(i) a_{kl} e_l(x_{i+1}^j) b_j^j(i+1)}{P(x^j \mid \theta)} \\ \gamma_i^j(k) &= P(\pi_i = k \mid x^j, \theta) \\ &= \sum_{l=1}^{|S|} \xi_i^j(k,l)\end{aligned}$$

We assume here that the length of the observation sequence is L.

1. Let g_1 be a function that computes A_{kl} for each two states k and l, that is, $g_1(X) = (A_{11}, \ldots, A_{ss})$, where
$$A_{kl}^j = \sum_{i=1}^{L-1} \xi_i^j(k,l),$$
$$A_{kl} = \sum_j A_{kl}^j.$$
2. Let g_2 be a function that computes $E_k(b_m)$ for each state k and symbol b_m, that is, $g_2(X) = (E_1(b_1), \ldots, E_1(b_n), \ldots, E_s(b_n))$, where
$$E_k^j(b) = \sum_{i=1, x_i^j = b}^{L} \gamma_i^j(k),$$
$$E_k(b) = \sum_j E_k^j(b).$$
3. Let g_3 be a function that compute a_{kl} and $e_k(b)$ according to (3) and (4).

4. Privacy-Preserving Parameter Estimation

4.1 Privacy Model

We propose a trusted boundary between a data miner and a data provider accessing to raw data. Data in the database contains a set of observable sequences. When miners want to build an HMM model, they have to provide the initial parameters and structure of HMM to the data provider.The data provider then produces a result by running an estimation maximization algorithm on the datasets and the HMM skeleton given by the data miner. After perturbed with noise, the result of the computation will be returned to the data miner.We use noisy versions $\hat{g}_1$ and $\hat{g}_2$ of g_1 and g_2 preserving differential privacy.

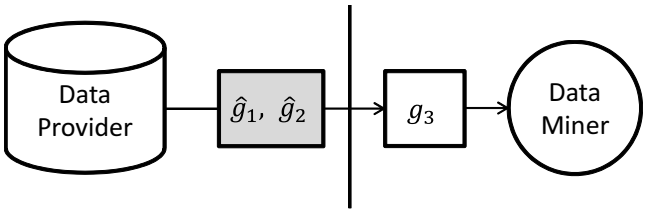

Fig. 4 DP version of parameter estimation

4.2 Adding Noise

The sensitivity of g_1 is as follows.

$$\begin{aligned}S(g_1) &= \max_{X, X' : |X \ominus X'| = 1} |g_1(X) - g_1(X')| \\ &= \max_j \sum_{k=1}^{s} \sum_{l=1}^{s} \sum_{i=1}^{L-1} \xi_i^j(k,l) \\ &\le L\end{aligned} \tag{5}$$

where $X \ominus X' = (X \setminus X') \cup (X' \setminus X)$. The sensitivity $S(g_2)$ of g_2 is L in analogy with g_1.

We use noisy version of g_1 and g_2, adding Laplace noise to each coordinate according to the sensitivity of g_1 and g_2:

$$\hat{A}_{kl} = A_{kl} + n_{kl}^A \ \text{ where } n_{kl}^A \sim Lap(L/\epsilon), \text{and}$$

$$\hat{E}_k(b) = E_k(b) + n_{k,b}^E \ \text{ where } n_{k,b}^E \sim Lap(L/\epsilon)$$

where ϵ is a parameter of differential privacy.

The resulting noisy EM will be,

$$\hat{a}_{kl} = \frac{\hat{A}_{kl}}{\sum_{l'} \hat{A}_{kl'}}$$

and,

$$\hat{e}_k(b) = \frac{\hat{E}_k(b)}{\sum_{b'} \hat{E}_k(b')}.$$

5. Experiments

In this part, we will show the trade-off between the model precision and the privacy preservation. We divided the experiment into two settings. In the first experiment, we only added noise in the last iteration of the training and in the second experiment, we added noise in every iteration of the training. The parameters for HMM training are Δ−threshold = 0.00001, maximum iterations = 80. The training datasets are: 10x100, 10x200, 10x300, 20x100,

20x200, 20x300, 30x100, 30x200, 30x300 where $m \times n$ means $m = L$ (the length of the observable sequence) and $n = \mid D \mid$ (the number of observable sequence). Since our study focuses on how the noise affects the HMM output, we evaluated the similarity of the maximum likelihood state paths produced by A_1 and A_2 when A_1 is the HMM obtained by the normal parameter estimation algorithm (called normally trained HMM) and A_2 is the HMM obtained by the proposed algorithm (called differentially private HMM).

5.1 Numerical Result

In the first experiment, we train HMM by applying Laplacian noise into the probability distribution only in the last iteration of the training in DP.

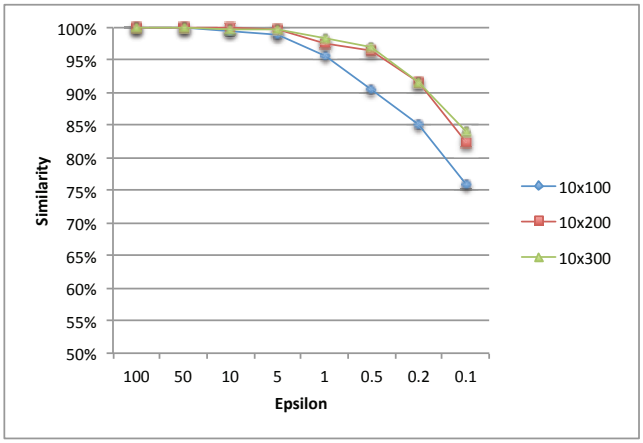

Fig. 5 2 states HMM training with length $L = 10$.

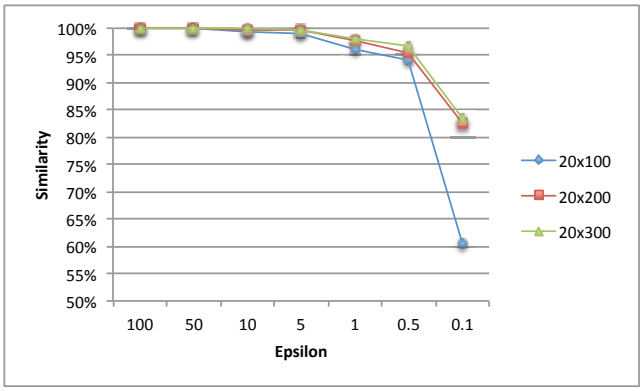

Fig. 6 2 states HMM training with length $L = 20$.

Figures 5, 6 and 7 show the similarity of normally trained HMM and differentially private HMM with different lengths and numbers of training sequences with HMM skeleton shown in Figure 4. Figures 8, 9, and 10 are the results for a three states HMM. As L, the length of observable sequence, becomes longer, the noise also becomes stronger and thus affecting the performance of the proposed algorithm.

Three states HMM trained by the same training dataset, are more affected by noise than two states HMM though the difference them is not so significant as shown in the figures.

In the second experiment, Laplacian noise is added into every training iteration. Adding noise in every iteration means that the

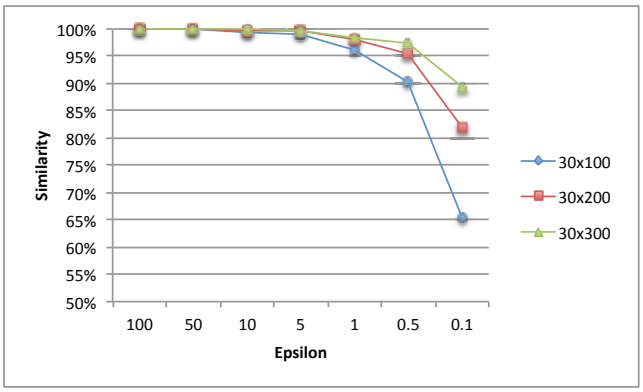

Fig. 7 2 states HMM training with length $L = 30$.

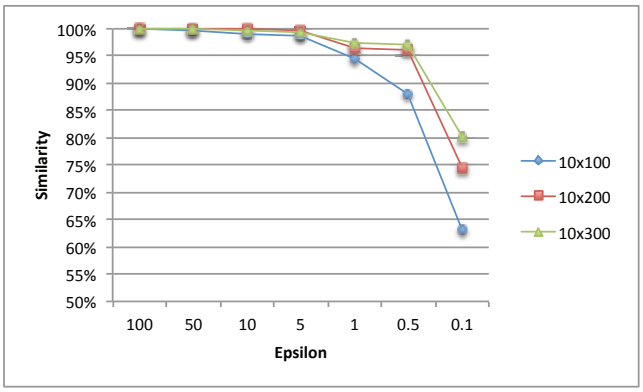

Fig. 8 3 states HMM training with length $L = 10$.

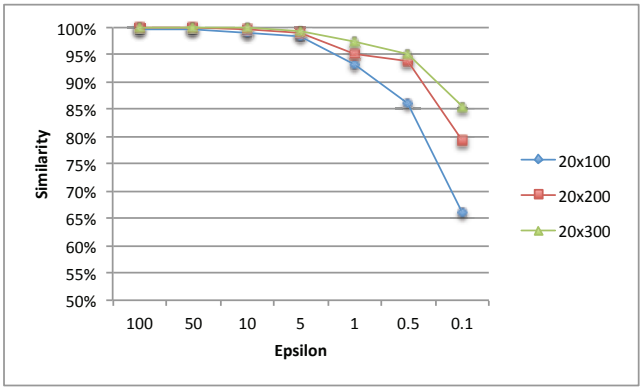

Fig. 9 3 states HMM training with length $L = 20$.

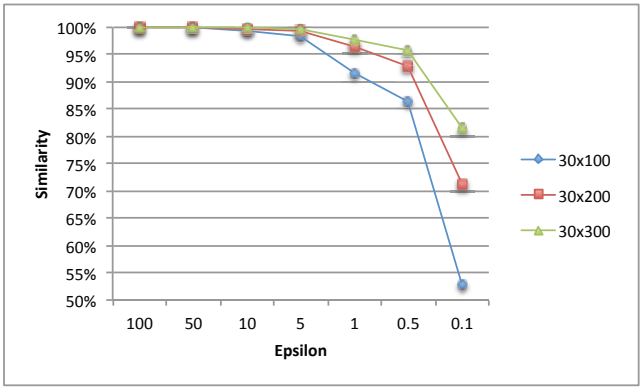

Fig. 10 3 states HMM training with length $L = 30$.

provided ϵ will have to be divided among each training iteration, so the ϵ used when adding noise becomes very low, resulting in a

very strong noise.

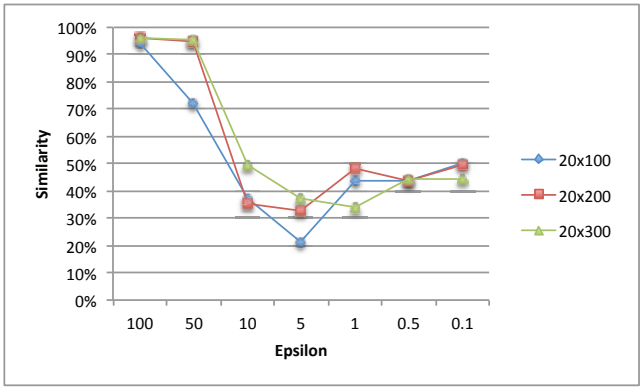

Fig. 11 2 states HMM for which a noise is added in every iteration.

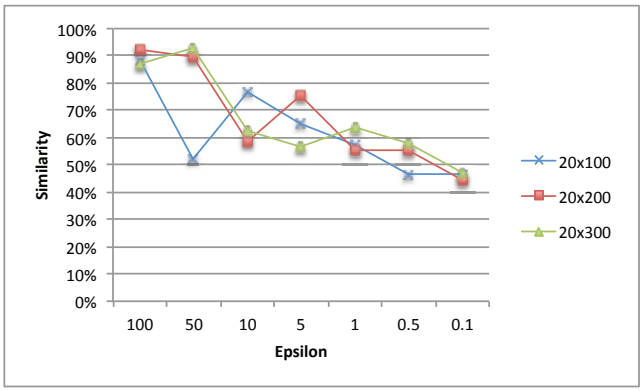

Fig. 12 3 states HMM for which a noise is added in every iteration.

From the result, performance of the proposed algorithm drops drastically when using low ϵ.

5.2 Discussion

As D, the number of observable sequences, becomes larger, the performance (accuracy) of the HMM becomes better. This is because the sensitivity of the proposed algorithm is independent of D.

It is not easy to find the optimal value L, the length of an observable sequence. The reason is as follows. In one hand, the sensitivity of the proposed algorithm is proportional to L and for the same ϵ, a larger L implies a larger noise. On the other hand, an observable sequence with larger L provides more information to the training algorithm.

When $\epsilon = 1$, the similarity of the proposed algorithm and the normal algorithm is not less than 95%.

In related studies, the value of ϵ is often set to 1 or below [4]. From these observations, the tradeoff between privacy and performance of the proposed algorithm is fairly good.

6. Conclusion

We have investigated the application of differential privacy for stateful data mining, the hidden Markov model. In this paper, we proposed an approach which can be realized in the application of HMM, by using noisy count in EM algorithm used by HMM training. Based on the sensitivity of the traditional parameter estimation algorithm for HMM, we proposed an ϵ-differential private algorithm for the parameter estimation. We empirically evaluate the trade-offs between privacy and model precision and we conclude that the proposed algorithm achieves a sufficient model precision while keeping ϵ-different privacy with $\epsilon \sim 1$.

References

[1] Blum, A., Dwork, C., McSherry, F. and Nissim, K.: Practical privacy: the SuLQ framework, *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ACM, pp. 128–138 (2005).

[2] Durbin, R.: *Biological sequence analysis: probabilistic models of proteins and nucleic acids*, Cambridge university press (1998).

[3] Dwork, C.: Differential privacy, *Automata, Languages and Programming*, Springer, pp. 1–12 (2006).

[4] Dwork, C.: Differential privacy: A survey of results, *Theory and Applications of Models of Computation*, Springer, pp. 1–19 (2008).

[5] Dwork, C., McSherry, F., Nissim, K. and Smith, A.: Calibrating noise to sensitivity in private data analysis, *Theory of Cryptography*, Springer, pp. 265–284 (2006).

[6] Friedman, A. and Schuster, A.: Data mining with differential privacy, *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 493–502 (2010).

[7] McSherry, F. and Mironov, I.: Differentially private recommender systems: building privacy into the net, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 627–636 (2009).

[8] McSherry, F. and Talwar, K.: Mechanism design via differential privacy, *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, IEEE, pp. 94–103 (2007).

[9] McSherry, F. D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis, *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, ACM, pp. 19–30 (2009).

[10] Ohm, P.: Broken promises of privacy: Responding to the surprising failure of anonymization., *UCLA Law Review*, Vol. 57, No. 6 (2010).

[11] Roy, I., Setty, S. T. V., Kilzer, A., Shmatikov, V. and Witchel, E.: Airavat: Security and Privacy for MapReduce, *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, Berkeley, CA, USA, USENIX Association, pp. 20–20 (online), available from ⟨http://dl.acm.org/citation.cfm?id=1855711.1855731⟩ (2010).

[12] Verykios, V. S., Bertino, E., Fovino, I. N., Provenza, L. P., Saygin, Y. and Theodoridis, Y.: State-of-the-art in Privacy Preserving Data Mining, *SIGMOD Rec.*, Vol. 33, No. 1, pp. 50–57 (online), DOI: 10.1145/974121.974131 (2004).