A Secure Protocol for Computing String Distance Metrics

Pradeep Ravikumar
CyLab,
Center for Automated
Learning and Discovery,
School of Computer Science,
Carnegie Mellon University
Pittsburgh PA, 15213, USA

William W. Cohen
Center for Automated
Learning and Discovery,
School of Computer Science,
Carnegie Mellon University
Pittsburgh PA, 15213, USA

Stephen E. Fienberg
CyLab,
Center for Automated
Learning and Discovery,
Department of Statistics,
Carnegie Mellon University
Pittsburgh PA, 15213, USA

Abstract

An important problem is that of finding matching pairs of records from heterogeneous databases, while maintaining privacy of the database parties. As we have shown in earlier work, distance metrics are a useful tool for record-linkage in many domains, and thus secure computation of distance metrics is quite important for secure record-linkage. In this paper, we consider the computation of a number of distance metrics in a secure multiparty setting. Towards this goal, we propose a stochastic scalar product protocol that is provably consistent, and is also as secure as an underlying set-intersection cryptographic protocol. We then use our stochastic dot product protocol to perform secure computation of some standard distance metrics like TFIDF, SoftTFIDF and the Euclidean Distance Metric. While asymptotically consistent, experiments show that the stochastic estimates are quite close to the true values after just 1000 samples. These secure distance computations can then be used to perform secure matching of records.

1. Introduction

A common goal is to perform some data mining task across multiple sets of data that are shared by several parties, without having any party divulge its data to the others. One technique for doing this is to randomize each data record, while maintaining the overall utility of the data for data-mining [2, 11]. Another technique uses *secure multiparty computation* to perform the data-mining task. Secure multiparty computation uses cryptographic primitives such as pseudo-random generating functions, homomorphic one-way encryption functions, etc., to achieve privacy.

However, in many cases, the data shared by the different parties will be to some degree *heterogeneous*—for instance, it might well be the case that two hospitals use slightly different strings to describe the name and address of the same patient. In this case, many proposed secure multiparty protocols will not work. Indeed, finding matching pairs of records accurately and efficiently is difficult, even when all parties are willing to divulge their data. Often called *record linkage*, this problem has been extensively studied, particularly in statistics (e.g., [12, 27]), AI (e.g., [24, 3]), and the database community (e.g., [15, 7, 20]).

One offshoot of prior work on record linkage has

been the development of a number of distance metrics for strings which are useful for matching in many domains [8, 9]. In this paper we consider computation of these metrics in a secure multiparty setting. In particular we consider a setting in which party R knows string r, party S knows string s, and the goal is to compute the distance between r and s without divulging additional information.

In addition to performing secure record linkage, a number of other plausible applications might use our secure distance metrics as cryptographic primitives. One example is performing near-neighbor searches against a database: for instance, matching a DNA pattern to a DNA pattern database, a fingerprint pattern to a fingerprint databases, or a patent application to a database of documents. This is sometimes called the *private information matching* problem [10].

Below, we first review related work in this area, and then summarize the formal model we use for privacy. We then describe three particular distance metrics—the TFIDF metric [21], the SoftTFIDF [8, 9, 4] metric, and Euclidean distance. We show that each of these distance metrics can be reduced to computing a secure dot-product of two vectors. We then present a stochastic sampling based method which securely *approximates* a dot-product, and prove a consistency result for its correctness. Our dot-product protocol uses an intersection protocol as an underlying cryptographic primitive. Finally, we present experiments evaluating the efficiency and accuracy of our scalar product protocol. We conclude with a summary of our work.

2. Related Work

Secure two-party computation was first investigated by Yao [28], and later work generalized it multi-party computation [14, 5]. These works propose very general results, yielding multi-party protocols for any probabilistic function, but are very inefficient. More recent work in privacy preserving data mining have proposed efficient cryptographic solutions to several specialized data-privacy problems. For instance, Kantarcioglu and Clifton [17] describe algorithms for mining association rules in horizontally partitioned data and Vaidya and Clifton [25] develop analogous algorithms for vertically partitioned data. Karr *et al*. [18, 19] provide extensions for doing secure regressions for horizontally partitioned data and Sanil *et al*. [22] for vertically partitioned data. These latter approaches require non-standard aspects because of the interest in quantities such as residuals and allow for statistically principled analyses.

Song *et al*. [23] describe a protocol to perform searches on encrypted data. Du and Atallah [10, 26] describe some approximate matching protocols which are appropriate for certain distance metrics, but not all of the metrics we consider here. They also describe secure multi-party dot-product protocols (in the context of larger constructions) but these are somewhat inefficient. Ioannidis *et al*. [16] describe a protocol for computing a dot-product; however, unlike our protocol, this method does not have provable security guarantees. Finally, Cohen and Lewis [6] present a random sampling based algorithm to identify the set of all vectors in a database which have a high dot product with a given query vector in sublinear time, but their algorithm does not carry over to the provable security framework.

3. Security Model

We develop our protocols in the setting of *semi-honest* or *honest-but-curious* behavior [13]. In this setting, each party follows the protocol properly, except that it may keep a record of all the intermediate computations and messages from the other party, and analyze these to get more information. Thus, each party is not *malicious* and does not alter its input. For example, in a secure data-mining protocol, if one party maliciously defines it's input to be the empty database, the output would be the result of applying the algorithm to the other database alone, and this would result in loss of privacy.

We use the minimal necessary information sharing paradigm [1] for our notion of *privacy*. Briefly, this entails that no information other their own inputs and some *minimal* additional information is revealed to the agents in the process of executing the protocol.

Clearly, just as repeated use of exact matching can be used to extract large amounts of information, repeated use of the secure distance protocol we present can also lead to divulging large amounts of information that could allow an intruder to penetrate individual data bases in a fashion that would not occur in the absence of the secure computation activities. In the present paper we will not consider this issue, but merely present the distance computation as a primitive.

4. Secure Distance Metric Protocols

In this section we shall describe three generally useful distance metrics, formally state the problems of computing the various distance metrics securely, and then propose protocols for the same.

4.1. TFIDF Distance

Problem Statement:*There are two parties R and S with strings r and s respectively. Compute the TFIDF distance (defined below) between the two documents privately in the minimal information sharing paradigm.*

The *TFIDF distance* is widely used in the information retrieval community to compare documents [21]. Subsequent experiments showed that it is also useful for comparing shorter strings in data integration [7].

To compute this distance metric, a string s is first broken into a set W_s of "tokens" (i.e., words). Each token w in W_s is given a numeric weight $weight(w, s)$. Following Salton [21], we use the formula

$$weight(w, s) = \log(\mathrm{TF}_{w,s} + 1) \cdot \log(\mathrm{IDF}_w)$$

where $\mathrm{TF}_{w,s}$ is the frequency of word w in s, N is the size of the "corpus" (e.g., the set of all strings known to either party), and IDF_w is the inverse of the fraction of strings in the corpus that contain w. Finally the distance between the two words sets W_r and W_s is defined to be

$$\textit{TFIDF}(W_r, W_s) = \frac{\sum_{w \in W_r \cap W_s} weight(w, r) \cdot weight(w, s)}{\sqrt{\sum_{w' \in W_r} weight(w', r)^2}\sqrt{\sum_{w' \in W_s} weight(w', s)^2}}$$

Is it well known that TFIDF distance can also be computed by computing the dot-product of two vectors VT_r and VT_s, where VT_r has dimensions corresponding to the terms w, and the value for $VT_r(w)$ is

$$VT_r(w) = weight'(w, r) = \frac{weight(w, r)}{\sqrt{\sum_{w' \in W_r} weight(w', r)^2}}$$

The TFIDF distance between r and s is then $VT_r \cdot VT_s$. Hence, if both parties can compute VT_s for a string s, one could use a secure scalar product protocol to obtain a secure TFIDF distance.

Notice that computing VT_s requires both parties to know IDF_w—in other words, they must share frequency statistics over some common corpora for a common vocabulary. In our setting, we will assume this to be the case. Alternatively, these frequency statistics might be approximated using two corpora, each known only to one party.

4.2. SoftTFIDF Distance

Problem Statement: *Given agents R and S as before, with strings r and s respectively, compute the SoftTFIDF distance (defined below) between the two documents.*

SoftTFIDF [8, 9] is a "softer" version of TFIDF, in which similar tokens are considered as well as tokens in $W_s \cap W_r$. Let sim be a secondary similarity function that is suited to comparing tokens. (In previous work [8], we achieved good results using the Jaro-Winkler distance, an easily-computed heuristic distance function, as sim). We use a thresholded version sim' wherein similarity values less than a threshold θ are taken as zero.

Let $CLOSE(w, S) = \max_{v \in S} sim'(w, v)$. We define

$$\textit{SoftTFIDF}(R, S) = \sum_{w \in R} weight'(w, R) \cdot weight'(CLOSE(w, S), S) \cdot sim'(w, CLOSE(w, S))$$

As $weight(w, R) = 0$ for $w \notin R$, we can extend the summand in the above equation to the whole vocabulary:

$$\textit{SoftTFIDF}(R, S) = \sum_{w \in VOCAB} weight'(w, R) \cdot weight'(CLOSE(w, S), S) \cdot sim'(w, CLOSE(w, S))$$

Let VT_r, VT_s be vectors such that $VT_r(w) = weight'(w, R)$, and $VT_s(w) = weight'(CLOSE(w, S), S) \cdot sim'(w, CLOSE(w, S))$.

The SoftTFIDF distance between r and s is then $VT_r \cdot VT_s$. Thus, one could use a secure dot-product protocol to compute the SoftTFIDF distance securely.

4.3. Euclidean Distance

Problem Statement: *Given agents R and S as before, with documents r and s respectively represented as weighted feature vectors V_r and V_s respectively, compute the Euclidean Distance between V_r and V_s securely.*

For vector V_r, replace each component r_i with three components $r_i^2, -2r_i, -1$. For vector V_s, replace each component s_i with three components $1, s_i, s_i^2$. The

dot product for these three components will then be $r_i^2 - 2s_i r_i - s_i^2 = (r_i - s_i)^2$. Extending this idea a little further, if $\mathbf{x}$ and $\mathbf{y}$ are arbitrary vectors,

$$\sum_i (x_i - x_i)^2 = \\ (\sum_i x_i^2, -2x_1, \ldots, -2x_n, 1) \cdot (1, y_1, \ldots, y_n, -\sum_i y_i^2)$$

and thus the Euclidean distance between two feature vectors can also be expressed as a scalar product of two vectors. Hence one could use the secure scalar product protocol to obtain a secure Euclidean distance.

5. Scalar Product Protocol

In this section, we shall describe a protocol for computing the scalar product of two vectors securely. We will make use of a secure intersection protocol as a cryptographic primitive.

Problem Statement: *Let there be two parties R and S with real-valued vectors V_r and V_s respectively, and let I denote some categories of information (specified below). Compute the scalar product of the two vectors securely, i.e., without revealing any additional information to either party except for the information contained in I.*

5.1. Steps of the Protocol

- Agent R computes and stores the normalization of V_r with respect to the L1 norm. Let the normalization factor be $Z_r = \sum_i V_r(i)$
- Let the dimension of both vectors be k. For ctr = 1 to numSamples, agent R samples $i \in \{1 \ldots k\}$ with probability $V_r(i)/Z_r$. He then adds (ctr, i) to his set T_r
- Similarly, Agent S computes and stores the normalization of V_s with respect to the L1 norm, Let the normalization factor be $Z_s = \sum_i V_s(i)$ Again, for ctr = 1 to numSamples, agent S then samples $j \in \{1 \ldots k\}$ with probability $V_s(j)/Z_s$ and adds (ctr, j) to his set T_s.
- The agents follow the secure intersection protocol for computing the intersection of T_r and T_s, $vp = T_r \cap T_s$. This is then averaged, $vp' = vp/numSamples$.

  The agents then multiply the answer with their respective normalization constants to get the vector product $i.e.\ vp'' = vp' * Z_r * Z_s$
- The categories of additional information I released by this protocol is Z_r, Z_s as well as the information I' released by the intersection protocol.

5.2. Proof of Correctness

While the intersection of the samples does not give the exact scalar product, it is consistent as given by the following lemma.

Lemma: $E(vp'') = V_r \cdot V_s$

Proof:

The size of the intersection of T_r and T_s is $T_r \cap T_s = \sum_{ctr=1}^{numSamples} I((ctr, T_r(ctr)) == (ctr, T_s(ctr)))$ where I is the indicator function.

Thus,

$$E(T_r \cap T_s) = \sum_{ctr=1}^{numSamples} Pr(T_r(ctr) == T_s(ctr))$$

The probability of two individual sampled entities matching is

$$Pr(T_r(ctr) = T_s(ctr)) = \sum_{i=1}^{k} ((V_r(i)/Z_r) * (V_s(i)/Z_s))$$

Thus,

$$E(T_r \cap T_s) = numSamples * \sum_{i=1}^{k} (V_r(i)/Z_r) * (V_s(i)/Z_s)$$

This gives:

$$\begin{aligned} E(vp'') &= numSamples * Z_r * Z_s * E(vp) \\ &= \sum_{i=1}^{k} (V_r(i) * V_s(i)) \end{aligned}$$

which is the scalar product of V_r and V_s.

5.3. Proof of Security

Steps 1 to 3 in the protocol are private computations by both parties. The only exchange of messages is in Step 4, via the Set-Intersection protocol. Thus, given a secure set-intersection protocol, the above scalar product protocol is also secure.

6. Experiments

We performed some experiments to test the empirical convergence of the stochastic TFIDF and SoftTFIDF

estimates to their true values. Due to our consistency result in Section 5.2, we do know that the stochastic estimates will converge to their true values asymptotically. As the graphs show, they appear to converge to the true values in about a thousand samples. We used the Cora [7] dataset for our evaluation. It contains record-strings with the fields author, title, date, and venue. We ranked by distance all candidate pairs from the dataset. We then computed the non-interpolated average precision of this ranking, which for a total of N pairs with m correct matches is $\frac{1}{m}\sum_{r=1}^{N}\frac{c(i)\delta(i)}{i}$, where $c(i)$ is the number of correct pairs ranked before position i, and $\delta(i) = 1$ if the pair at rank i is correct and 0 otherwise.

In Figure 2, we compare the performances of the vanilla distances, and the stochastic versions of the distances. We see that the precision values of the stochastic distances approach that of the true distances in very few samples.

In Figure 1, we compare the actual distance values of the stochastic string distances and the true string distances for the following record-pair:

- "harris drucker, robert schapire, and patrice simard. 7(4) boosting performance in neural networks. international journal of pattern recognition and artificial intelligence, 1993. 705-719"
- "harris d., robert s., and patrice s. 7(4) boosting performance in neural networks. international journal of pattern recognition and artificial intelligence, pages 705–719"

Again, we empirically observe near convergence with few samples.

7. Conclusions

We have proposed a stochastic scalar product protocol that is provably consistent, and is also as secure as an underlying set-intersection cryptographic protocol. We then use our stochastic dot product protocol to perform secure computation of some standard distance metrics like TFIDF, SoftTFIDF and the Euclidean Distance Metric. While asymptotically consistent, experiments show that the stochastic estimates are quite close to the true values after just 1000 samples. Such secure distance computations can then be used towards the task of secure matching of records. We also noted some issues regarding the vulnerability of the original separate secure data bases as a result of the computation. Extensions to more complex statistical calculations such as secure regression calculations go beyond the methods we present here.

8. Acknowledgments

The preparation of this paper was supported in part by National Science Foundation Grant No. EIA-0131884 to the National Institute of Statistical Sciences and by the U.S. Army Research Office Contract DAAD19-02-1-3-0389 to CyLab at Carnegie Mellon University. We also thank John Lafferty and Latanya Sweeney for helpful discussions.

References

[1] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 86–97. ACM Press, 2003.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 439–450. ACM Press, May 2000.

[3] M. Bilenko and R. Mooney. Learning to combine trained distance metrics for duplicate detection in databases. Technical Report Technical Report AI 02-296, Artificial Intelligence Lab, University of Texas at Austin, 2002. Available from http://www.cs.utexas.edu/users/ml/papers/marlin-tr-02.pdf.

[4] M. Bilenko, R. Mooney, W. Cohen, P. Ravikumar, and S. Fienberg. Adaptive name matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23, 2003.

[5] D. Chaum, C. Crepeau, and I. Damgard. Multiparty unconditionally secure protocols. *In Proceedings of the 20th Annual ACM Symposium on Theory of Computing, STOC'88*, 1988.

[6] E. Cohen and D. D. Lewis. Approximating matrix multiplication for pattern recognition tasks. In *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*, pages 682–691. Society for Industrial and Applied Mathematics, 1997.

[7] W. W. Cohen. Data integration using similarity joins and a word-based information representation language. *ACM Transactions on Information Systems*, 18(3):288–321, July 2000.

[8] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web*, 2003.

[9] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string metrics for matching names and records. In *Proceedings of the KDD-2003 Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, 2003.

[10] W. Du and M. Atallah. Protocols for secure remote database access with approximate matching. *In Proc. of the First Workshop on Security and Privacy in E-Commerce*, 2000.

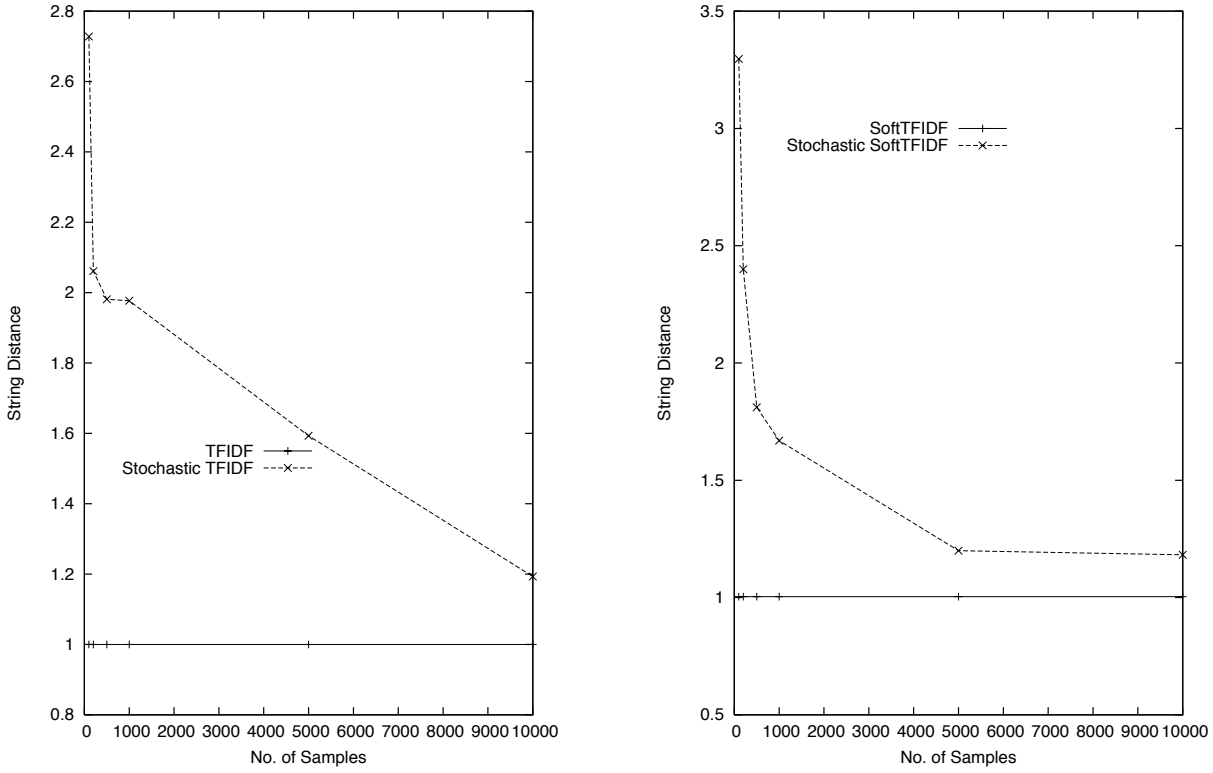

Figure 1. Convergence of Stochastic TFIDF Distance of a record-pair (left) and Stochastic SoftTFIDF Distance (right)

[11] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. *Proc. of 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2002.

[12] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Society*, 64:1183–1210, 1969.

[13] O. Goldreich. *The Foundations of Cryptography - Volume 2*. Cambridge University Press, 2004.

[14] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. *Nineteenth Annual ACM Symposium on Theory of Computing, pages 218-229.*, 1987.

[15] M. Hernandez and S. Stolfo. The merge/purge problem for large databases. In *Proceedings of the 1995 ACM SIGMOD*, May 1995.

[16] I. Ioannidis, A. Grama, and M. Atallah. A secure protocol for computing dot products in clustered and distributed environments. *In Proceedings of the International Conference on Parallel Processing, Vancouver, Canada*, 2002.

[17] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *In The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'02), June 2 2002.*, 2002.

[18] A. F. Karr, X. Lin, J. P. Reiter, and A. P. Sanil. Secure regressions on distributed databases. *Journal of Computational and Graphical Statistics*, 2004.

[19] A. F. Karr, X. Lin, A. P. Sanil, and J. P. Reiter. Analysis of integrated data without data integration. *Chance*, 17(3):27 – 30, 2004.

[20] V. Raman and J. Hellerstein. Potter's wheel: An interactive data cleaning system. In *The VLDB Journal*, pages 381–390, 2001.

[21] G. Salton, editor. *Automatic Text Processing*. Addison Welsley, Reading, Massachusetts, 1989.

[22] A. P. Sanil, A. F. Karr, X. Lin, and J. P. Reiter. Privacy preserving regression modelling via distributed computation. In *KDD 2004 Conference Proceedings*, 2004.

[23] D. Song, D. Wanger, and A. Perrig. Practical techniques for searches on encrypted data. *Proceedings of the IEEE Security and Privacy Symposium*, 2000.

[24] S. Tejada, C. A. Knoblock, and S. Minton. Learning object identification rules for information integration. *Information Systems*, 26(8):607–633, 2001.

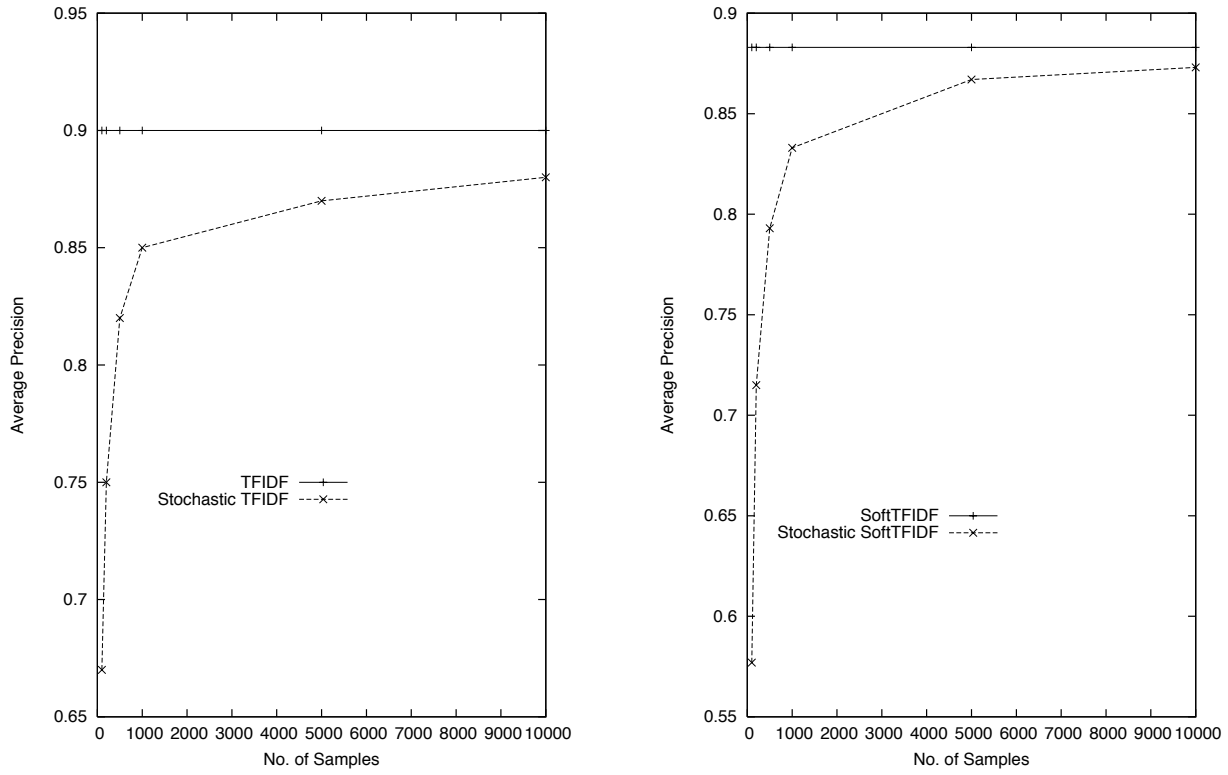

Figure 2. Average Precision values of Stochastic TFIDF Distance (left) and Stochastic SoftTFIDF Distance (right)

[25] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. *In The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26 2002*., 2002.

[26] W.Du and M. J. Atallah. Privacy-preserving cooperative scientific computations. In *14th IEEE Computer Security Foundations Workshop*, pages 273–282, Nova Scotia, Canada, June 11-13 2001.

[27] W. E. Winkler. The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service Publication R99/04. Available from http://www.census.gov/srd/www/byname.html, 1999.

[28] A. C. Yao. Protocols for secure computation. In *Proceedings of the 23rd IEEE Symposium on Foundations of Computer Science*, pages 160–164, 1982.