The Computational Complexity of K-anonymity: A Survey

Ahmad Charba

December 2023

1 Introduction

Data publication can greatly benefit private and public entities. This, however, shouldn't be done at a great cost to the privacy of individuals whose data is published. In order to protect the anonymity of individuals in a data set, the propriety of k-anonymity was developed for data sets, giving a strong privacy guaranty [1]. Many k-anonymous versions of a data set can exist, leading to the development of k-anonymization techniques that attempt to minimize the amount of information lost (in other words, maximizing the usefulness) while guaranteeing a k-anonymous data set.

This paper will look into the computational complexity of the optimal solution, as well as the complexity of different proposed approximate algorithms.

2 Background

2.1 Quasi-identifiers

Stripping a data set of identifying information, such as name and phone number, is an obvious step to protect the privacy of the individuals whose data is included in it. However, it is not enough; data can be de-anonymized with great success by using "quasi-identifiers", a term first used in 1986 by Dalenius to describe a set of data points which alone may not be enough to identify an individual, but gain that capability when combined [2]. A good example of this is the set (zip code, sex, date of birth) which was shown by Sweeney in 2000 to be sufficient to identify 87% of the US population, only using public data sets [3].

2.2 K-anonymity

This shows the need to further anonymize data sets prior to their publication, which is why k-anonimity was proposed by Sweeney in 2002 [1]. This property is defined in the following way.

2.2.1 Definition

Let a table $V = (v_1, v_2, ..., v_n)$ and a quasi-identifier Q. Each vector v_i represents a row of m values. Then, V is k-anonymous if and only if each row in V is in a set of k rows where each row has the same set of values for its quasi-identifier. In other words, the rows are in a group of k rows indistinguishable with regards to their quasi-identifiers.

2.3 K-anonymization methods

Several operators have been developed to anonymize data sets. The more prominent ones are suppression of an attribute, suppression of an entry, and generalization.

2.3.1 Suppression of an attribute

Suppression of an attribute is done by completely removing an attribute (ex. zip code, date of birth) from the data set.

2.3.2 Suppression of an entry

Suppression of an entry is done by replacing a data point by a symbol outside of the original set of possible values, representing a suppressed information. This is more granular than the suppression of an attribute, often allowing for less suppressing of information to achieve k-anonymity.

2.3.3 Generalization

Like suppression of an attribute, generalization modifies an attribute to make its values more general, thus making more rows indistinguishable. This is easily done for numerical attributes such as age, which could be generalized to ranges of ages.

2.3.4 Supressor

Let V be a data set able to take values in Σ for its entries. We define a suppressor, t, which is only able to suppress entries, as follows: t is a suppressor for V if for any v_i, for any j in [1, m], $t(v_i)[j] \in \Sigma, *$, where * represents a suppressed entry.

3 NP-Hardness of Finding the Optimal Solution using Suppression

One of the early and important results in the study of the complexity of k-anonimity is the proof in 2004 by Meyerson and Williams that k-anonimity is NP-hard for $k \geq 3$ using suppression of entries [4]. I will outline the original proof of the result in the following section.

3.1 Proof of NP-hardness for $k > 3$

The proof is a reduction from the k-dimensionnal perfect matching problem, which was shown in 1978 to be NP-complete [15] and is defined in the following way:

Given a k-hypergraph H = (U, E) with n = #U and m = #E, is there a subset S $\subseteq$ E of n/k hyper edges such that each vertex of U is contained in exactly one hyperedge of S.

The proof will be done for k=3 but can trivially be generalized to k¿3. We suppose $|\Sigma| \geq \#V$ for the purpose of this proof, though Williams and Blocki showed in 2010 that the NP-hardness of k-anonymity persists even when the attributes are binary [6].

Let G a simple 3-hypergraph (in the case of a graph which isn't simple, we can apply the same argument after removing the duplicate edges). We will construct a data set V from G and show that G has a 3-dimensional perfect matching if and only if there is a suppressor t such that t(V) is 3-anonymous and has at most n(m-1) suppressed entries.

Let $e_1, ..., e_m$ be the edges in G. We define $V = (v_1, ..., v_n)$ such that for each node u_i in G, there is a vector v_i in V and

$$v_i[j] = \begin{cases} 0 & \text{if } u_i \in e_j \\ i & \text{otherwise} \end{cases}$$

For $0 < j < m$.

3.1.1 Perfect matching implies existence of t

For the first direction, suppose that G has a 3-dimensional perfect matching M. Now, we define a suppressor t such that:

$$t(v_i)[j] = \begin{cases} 0 & \text{if } u_i \in e_j \wedge e_j \in M \\ * & \text{otherwise} \end{cases}$$

Since $u_i \in e_j$, we have that $v_i[j] = 0$ by construction, and since every other entry in $t(v_i)$ is a *, t is a suppressor for V.

We also have that t(V) is a 3-anonymous data set. Notice that there is one and only one entry in each v_i where $v_i[j] = 0$, since u_i is contained in exactly one hyper-edge in the perfect matching M. That means that for all 3 vectors v_g, v_h, v_i in an edge e_p of M, their values are * for every $j \neq p$ and p when $j = p$, making them indistinguishable. So, each vector v_i is in a group of 3 indistinguishable vectors (v_g, v_h, v_i).

3.1.2 Existence of t implies perfect matching

Now for the other direction, suppose a 3-anonymous suppressor t which suppresses at most n(m-1) entries.

We can show that any $t(v_i)$ contains at most 1 non-* entry by contradiction: if it had another non-* entry, then there must be 2 other vectors with the same two (or more) non-* entries. That would apply that a group of 3 vectors have the

same 2 edges. But, the two edges would link the same vectors, which contradicts that the graph is simple.

This implies that that t(V) has at most n non-* entries. And, since t suppresses at most n(m-1) entries, at least n entries must not be suppressed. This means that exactly n entries are non-*, in other words, exactly 1 entry per vector is non-*.

From t(V), we can construct a perfect matching M by selecting the edges e_j where there is a $t(v_i)[j]$ which is non-*. Then, we have at most n/3 edges (since each non-* entry has at least two more identical entries, and there is only 1 non-* entry per vector) and each vector is in at least 1 edge. This means that M is a perfect matching for G.

3.2 Proof that 2-anonymity is in P

In a the 2010 paper mentioned earlier, Blocki and Williams showed that 2-anonymity is in P. This section will give an outline of the proof.

The result is proven by reducing 2-anonymity to Simplex Matching, a problem which is a special case of hypergraph matching.

3.2.1 Simplex Matching

The Simplex Matching problem is defined as:

Given a hypergraph H containing edges of sizes 2 or 3 with edge costs c(e), find a perfect matching H with minimal costs [7].

This problem has been shown in 2007 by Elliot Anshelevich and Adriana Karagiozova to be polynomially computable under the following condition, called the Simplex condition [7]:

$c(u_1, u_2) + c(u_2, u_3) + c(u_1, u_3) \leq 2c(u_1, u_2, u_3)$

3.2.2 Reduction of 2-anonymity to Simplex Matching

Take a data set D with n vectors. We can build a hypergraph G that respects the Simplex condition the following way:

1. Add a node u_i in G for every vector v_i in D
2. Add a 2-edge u_i, u_j for every possible pair of vector v_i, v_j, with cost $c(u_i, u_j) = C_{i,j}$, the cost in number of stars to make the vectors identical
3. Add a 3-edge u_i, u_j, u_k for every possible pair of vector v_i, v_j, v_k, with cost $c(u_i, u_j, u_k) = C_{i,j,k}$, the cost in number of stars to make the vectors identical

We can show the Simplex condition is respected for this hypergraph. Take two vectors v_i, v_j. The cost to make them identical is $C_{i,j}$, meaning each of the vectors will have $\frac{1}{2}C_{i,j}$ stars when anonymized. Similarly, the cost of each of the three vectors v_i, v_j, v_k is $\frac{1}{3}C_{i,j,k}$. We know that the number of stars per vector

cannot decrease when adding the vector k to the vectors to me identical. This gives us the following equations

$\frac{1}{3}C_{i,j,k} \geq \frac{1}{2}C_{i,j}$
$\frac{1}{3}C_{i,j,k} \geq \frac{1}{2}C_{i,k}$
$\frac{1}{3}C_{i,j,k} \geq \frac{1}{2}C_{j,k}$

By adding all three inequalities, we get

$C_{i,j,k} \geq \frac{1}{2}C_{j,k} + \frac{1}{2}C_{i,k} + \frac{1}{2}C_{i,j}$
$2C_{i,j,k} \geq C_{j,k} + C_{i,k} + C_{i,j}$

Therefore, we respect the Simplex condition. So 2-anonymity is in P.

4 Other Hardness Results

This section will cite some additional results obtained about the hardness of k-anonymity.

In 2008, Sun, Wang and Li showed the problem to be NP-Hard even after restricting it to having a finite alphabet[14].

In the same paper as the one where 2-anonymity was shown to be in P, Blocki and Williams showed the following results:

- 3-anonymity is NP-hard even with binary attributes, using a reduction from the problem of partitioning the edges of a triangle-free graph into 4-stars (degree-three vertices), which was shown to be NP-hard.
- 3-anonymity with 27 attributes is MAX SNP-hard.
- A bound for k-anonymity: $O(4^n poly(n))$
- Given an alphabet size c, l attributes, and n rows, k-anonymity can be solved in $2^{O(k^2(2c)^l)} + O(nl)$ time

In 2009, Gionis and Tassa proposed an alternative measure of the loss of information, able to measure the loss from generalization, and showed that the problem is still NP-Hard with this new measure[8].

5 Approximations

Since finding the optimal solution is an NP-Hard problem, finding a good approximation algorithm has been the focus of many studies. The performance of these algorithms is often measured by the asymptotic bound on their approximation ratios, their time complexity with regards to n, and their time complexity with regards to k.

Algorithms which include a better bound on k solve a more general problem, however in practice, the k that is most often used is often no more than 5 or 6 [1], a smaller approximation ratio may be preferred to a smaller time complexity with regards to k.

In their original paper showing the NP-hardness of k-anonymity (2004), Meyerson and Williams proposed [4] :

- A polynomial algorithm on n with an approximation ratio of O(k log(k)). But, the algorithm was exponential in k. This was a greedy algorithm based on the number of coordinates in which two vectors differ.
- A polynomial algorithm on k and n with a O(k log(m)) approximation ratio.

In 2007, Park and Shim proposed [13]:

- A O(log(k)) approximation which performs better than previous proposals in experiments.
- A O(β log(k)) approximation which also performs better in practice, and increases its efficiency when increasing β the tolerance for the approximation.

These algorithms are polynomial on n, but the authors did not analyze how the time complexity scales with k.

In 2011, Kenig and Tassa proposed algorithms using their improved measure of information loss[9]:

- A O(log(k))-approximate algorithm polynomial in n and exponential in k.
- A O(k) approximation algorithm polynomial in both k and n

6 Recent Works

Since around 2015, the focus has shifted from deterministic algorithms with approximation guaranties to more non-deterministic algorithms. Examples include the 2017 paper "Density-based Clustering Method for K-anonymity Privacy Protection" [10],the 2018 paper "K-Anonymity Algorithm Based on Improved Clustering" [12] the 2023 paper "Optimization-Based k-Anonymity Algorithms" [11], none of which provide a bound for the approximation ratio.

This makes it difficult to compare deterministic algorithms to non-deterministic ones, as one side is based solely on experimental results and the other mainly on theoretic results. It also complicates the comparaison of time complexity, as some non-deterministic algorithms may run for as long as they are allowed, rather than ending and giving the best answer it could have given.

7 Future Works

Based on this survey, the topic of k-anonymity and its complexity still would benefit from further research. Here are directions that may yield interesting results:

- Research towards a deterministic approximation with approximation ratio less than logarithmic, or evidence against its existence.

- Research towards a log-approximate algorithm that is polynomial with regards to both k and n.
- A comparison of deterministic and non-deterministic algorithms, with regards to their scalability and performance, and research on hybrid algorithms.
- Research on the complexity of related properties, such as l-diversity, which solves the weakness of k-anonymity against homogeneity attacks (which was shown to be NP-hard as well [5]).
- Additional complexity results for other alternate measures of information loss.

8 Conclusion

Though it may not be sufficient on its own, k-anonymity is a very important tool for protecting the privacy of people around the world. The problem of finding an optimal k-anonymization is NP-hard under most forms of the problem, but good approximations for it do exist. The current direction seems to be non-deterministic algorithms, similarly to many other problems that can benefit from approximate solutions. Further research in the area would be useful in improving the scalability of k-anonymity.

9 Acknowledgements

My thanks to Professor Andrej Bogdanov for teaching the Computational Complexity course and for the helpful feedback throughout the year, and especially for this project.

References

[1] Sweeney, Latanya. "k-anonymity: A model for protecting privacy." International journal of uncertainty, fuzziness and knowledge-based systems 10.05 (2002): 557-570.

[2] Dalenius, T. (1986). "Finding a needle in a haystack or identifying anonymous census records." Journal of Official Statistics, 2(3), 329. Retrieved from https://login.proxy.bib.uottawa.ca/login?url=https://www.proquest.com/scholarly-journals/finding-needle-haystack-identifying-anonymous/docview/1266806751/se-2

[3] Latanya Sweeney. 2000. "Simple Demographics Often Identify People Uniquely." Carnegie Mellon University, Data Privacy. Type of Work: Working paper. Project website

[4] Adam Meyerson and Ryan Williams. "On the complexity of optimal k-anonymity." Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. 2004.

[5] Dondi R. et al. "The l-diversity problem: Tractability and approximability." Theoret. Comput. Sci. (2013)

[6] Blocki, J., Williams, R. (2010). Resolving the Complexity of Some Data Privacy Problems. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds) Automata, Languages and Programming. ICALP 2010. Lecture Notes in Computer Science, vol 6199. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-14162-1_33

[7] Elliot Anshelevich and Adriana Karagiozova. 2007. Terminal backup, 3D matching, and covering cubic graphs. In Proceedings of the thirty-ninth annual ACM symposium on Theory of computing (STOC '07). Association for Computing Machinery, New York, NY, USA, 391–400. https://doi.org/10.1145/1250790.1250849

[8] A. Gionis and T. Tassa, "k-Anonymization with Minimal Loss of Information," in IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 2, pp. 206-219, Feb. 2009, doi: 10.1109/TKDE.2008.129.

[9] Kenig, B., Tassa, T. A practical approximation algorithm for optimal k-anonymity. Data Min Knowl Disc 25, 134–168 (2012). https://doi.org/10.1007/s10618-011-0235-9

[10] Liu, J., Yin, S., Li, H., & Teng, L. (2017). A Density-based Clustering Method for K-anonymity Privacy Protection. J. Inf. Hiding Multim. Signal Process., 8, 12-18.

[11] Liang, Yuting; Samavi, Reza (2023). Optimization-Based k-Anonymity Algorithms. Toronto Metropolitan University. Preprint. https://doi.org/10.32920/24132903.v1

[12] Zheng, W., Wang, Z., Lv, T., Ma, Y., Jia, C. (2018). K-Anonymity Algorithm Based on Improved Clustering. In: Vaidya, J., Li, J. (eds) Algorithms and Architectures for Parallel Processing. ICA3PP 2018. Lecture Notes in Computer Science(), vol 11335. Springer, Cham. https://doi.org/10.1007/978-3-030-05054-2_36

[13] Hyoungmin Park and Kyuseok Shim. 2007. Approximate algorithms for K-anonymity. In Proceedings of the 2007 ACM SIGMOD international conference on Management of data (SIGMOD '07). Association for Computing Machinery, New York, NY, USA, 67–78. https://doi.org/10.1145/1247480.1247490

[14] Sun, X., Wang, H., Li, J. (2008). On the Complexity of Restricted k-anonymity Problem. In: Zhang, Y., Yu, G., Bertino, E., Xu, G.

(eds) Progress in WWW Research and Development. APWeb 2008. Lecture Notes in Computer Science, vol 4976. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-78849-2_30

[15] David G. Kirkpatrick and Pavol Hell. 1978. On the completeness of a generalized matching problem. In Proceedings of the tenth annual ACM symposium on Theory of computing (STOC '78). Association for Computing Machinery, New York, NY, USA, 240–245. https://doi.org/10.1145/800133.804353