Anonymization of Data MSDS 7349 Project Draft

Alex Frye, Michael Smith, and Lindsay Vitovsky

Abstract—Big data certainly brings with it a series of challenges, one of them being privacy of these large data sets. In the name of scientific progress, an amazing number of data sets have been released as a way to promote collaboration and development of the data science community. This paper critically reviews an array of anonymization theories and techniques, applying them to real-world situations, with the goal of helping others weigh the cost, benefit, and, most importantly, the ethics of anonymizing data.

Index Terms—micro-data, k-anonymization, quasi-identifiers, unique identifiers, personal ly identifiable information (PII), high-dimensional data sets, sparcity

1 Introduction

On September 18, 2009, Netflix announced the winner of its Netflix Prize competition, a contest that asked contributors to improve upon Netflixs existing algorithm that predicted subscribers future movie ratings. With a prize of $1,000,000, the competition certainly attracted many groups from all over the globe. In the end, Netflix received 44,014 submissions from 41,305 teams, all who had analyzed the seemingly anonymized dataset of over 100,000,000 movie ratings by Netflix subscribers.

According to the Netflix Prize website, which was still live at the time of this paper, The data were collected between October, 1998 and December, 2005 and reflect the distribution of all ratings received by Netflix during this period. The ratings are on a scale from 1 to 5 (integral) stars. To protect customer privacy, all personal information identifying individual customers has been removed and all customer ids have been replaced by randomly-assigned ids. The date of each rating and the title and year of release for each movie are provided. No other customer or movie information is provided.

Unfortunately for Netflix, resulting from this contest was a lawsuit for releasing this data, and an investigation by the Federal Trade Commission. Apparently, one actually could identify actual users from this data set, as proven by two University of Texas researchers in 2008 [1]. To make matters worse, Netflix did not end up using the winning algorithm, citing engineering challenges and the fact that their model was rapidly changing from DVDs to streaming [2].

The repercussions from this anticlimax caused Netflix to cancel its Netflix Prize 2, drawing both praise and criticism. Simply reading the comments section of various online articles covering this news reveals the various viewpoints of the importance (or non-importance) of privacy in todays cyber world.

"I really don't see the problem with this..."
Killer Orca [3]

Alex Frye, amfrye@smu.edu
Michael Smith, michaelsmith@smu.edu
Lindsay Vitovsky, lvitovsky@smu.edu

"You would think with that huge privacy snafu AOL had with their anonymized search data that someone at Netflix would have realized this was a bad idea."
ozziegt [3]

"Just some lawyers looking for profit, hugely inflating the risks"
Anonymous [5]

"Whether you consider it offensive or not should be irrelevant; if the data is private, they cant give it away."
danchr [4]

Even if Netflix had obtained user permission to release these ratings (which they did not), those who opted in surely would not have surmised that they could have been identified personally by a complete stranger unaffiliated with Netflix. Certainly, Netflix would have assured that their ratings were anonymized, since that is what it believed to be the case as well. The trust that the subscribers would have placed in Netflix to better their experience, without compromising their privacy, would have been misplaced.

Today, there are several popular techniques for anonymizing data sets, but none are perfect. Certain methods may work for a particular sized set of records, only to lose its credibility as that set scales over time. Others may change the data so much that its dramatically declines to a data scientist. Is true anonymization even possible in this cyber world? The ability to pull public information from outside sources, to then match it with private data sets, has proven to be a very dangerous risk for institutions and the individuals that have entrusted them.

Our team seeks to investigate these most popular methods, weighing their effectiveness in regards to: 1) the size of a data set, 2) the data type (i.e. categorical v. numerical attributes, 3) the effect on data science results (i.e. is there a change to a 95% confidence interval because of the perturbation?), 4) deployment in various industries, particularly the medical and financial fields due to their collection of personal information and regulatory oversight. We will then try to anonymize a data set we created, investigating how easy or difficult it is to garner identifying information from something seemingly innocuous. Ultimately, we make final

recommendations to the reader as to what we believe are essential questions a company should be reflecting upon as they consider their privacy policies and procedures.

1.1 Background

1.1.1 Existing Research / Techniques

Before traveling too much further into these methods, it warrants mentioning that privacy itself has yet to be universally defined. One could say that privacy is when non-public information is made available to a third party, such as a social security number or the results from a medical test. This insinuates that the nature of the data is what determines the need for privacy. But what about those pesky ads that pop up on a smart phone? Ads are to be expected, but what if they directly reflect the search history a user performed on a completely different device? The discussion of what privacy means, particularly what a user authorizes others to collect, store and share, has yet to be answered in a world where mobile devices outnumber humans [6]. There are many individuals who would tell you that they feel their privacy has been violated by this collection and use of their search data.

Of equal concern to the authors is the other end of the spectrum, which is the evidently voluntary surrender of the collection and use of one's data. There are a large number of users who do not seem to take issue with their non-personal data being stored or sold to other companies. The task of maintaining privacy is certainly overwhelming, and perhaps this is why so many users throw up a white flag and say, "Who cares?" Perhaps, they simply do not want to think about the data that is out there, since there is presumably nothing that can be done about it.

Adding on to this acceptance of the release of unimportant data is the common misconception that less privacy means more security. When the FBI very publicly condemned Apple for not helping it unlock an iPhone that belonged to a gunman who helped kill fourteen people in San Bernardino, CA in 2015, there were many people who joined the FBI in the criticism. The reality is that these keys to our privacy are not held behind a locked safe, guarded by trustworthy individuals. Often our privacy is protected simply on an agreement of trust. Not only would the FBI now have known how to unlock an iPhone, but there would have been employees of Apple that would have then been privy to the information as well. Our message here is that the bounds around privacy should not be thought of as well-defined. The questions must be asked so that those involved can properly reflect on the danger of using anonymized data and weigh it against any supposed benefit.

1.2 Testing of a Dataset

To experience an anonymization as well as de-anonymization for ourselves, we created a fake data set of 4,000 records. The attributes included are:

This data set was generated randomly, and provides us with several opportunities to try common anonymization techniques discussed in this paper. The attributes were chose to reflect common member records that would go under de-identification methods such as 1) changing addresses to zip codes, 2) changing specific ages to an age range, 3) keeping only the area code of a phone number, 4) removing all names and replacing with one field of an assigned identification number, and 5) dividing the columns into less comprehensive tuples.

TABLE 1
Dataset Attributes

Name	string divided into first name, middle name, and last name
Sex	categorical of male or female
Age	numerical
Ethnicity / race	categorical
Hispanic / Latino	binary categorical yes / no
Hair color	categorical
Eye Color	categorical
Address	string
Phone number	string

1.2.1 Analysis of Existing Techniques

Of particular interest to our paper were some of the more frequently employed anonymization techniques. To be clear, our definition for these techniques fall more into the realm of attempts as it has been well documented that anonymization has continued to elude the cyber community [ENTER SEVERAL ENDNOTES OF OUR PAPERS WE READ, XIANG BAI-LI, LOUKIDES and NARAYANAN].

K-anonymization

In researching k-anonymization, this technique is most widely used as a method for maintain as much statistical integrity. By removing instances with less than k number of matching records, the data set seeks to protect the ease of finding loners by taking them out altogether. For example, if you have 30,000 surgery patients, and all personal information has been removed, then it is possible to have multiple instances of the same data (i.e. twelve gallbladder removal surgeries). However, as documented by Greg Loukides team, the efficacy of this method on large sets of anonymized data has its challenges [7]. K-anonymization acknowledges that those records that do not appear as often are at increased risk of being re-identified, so unless that record has k number of other records like it, it is removed. While theoretically sound, the disadvantages are the computational power required on large data sets to find these counts, and the assumption that the remaining tuples are measurably more difficult to match to additional data.

Perhaps the domain comes into play here. For medical information, physical attributes and maladies can often be accompanied by other instances of care. It would be wise for those storing this data to understand the overall web of available data on someone who might only be treated for an infection. This patient will most likely be prescribed antibiotics, which means there might be pharmacy information to match. Consider a more multi-symptomatic issue high blood pressure. Suppose a patient was being treated for this, but by his or her psychotherapist? Now we have potentially devastating information that could alert someone in this

persons circle of friends, family, and work colleagues of a mental illness.

The benefit of k-anonymization is not be ignored, however. It certainly helps to avoid making the easiest to identify as low hanging fruit. The benefit to statistical analysis is also to be given credit. While one would not have the nuanced meaning that outliers can bring, with data sets being so large, the argument stands that these one offs can be ignored with the same level of confidence. [Try this on our data set?]

Sparse Data Sets

Heavily used by many large corporations including Amazon, sparse data sets have become a very prevalent way to store tuples [1], [8], [9]. Sparse data is kept in a much less comprehensive form, only confessing a small number of attributes per record. This results in many more records, perhaps duplicate records, and a harder time for adversaries to draw connections among the many tuples.

As tested by Narayanan and Shmatikov, this method has its drawbacks as well. As the data set grows, it is easier for an adversary to find patterns and relationships in sparse data. We liken it to the experience of purchasing groceries. The cashier (or you, if you find yourself at a self-checkout kiosk) can either place many items in few bags, or only a few items inside of many bags. Say you wanted to pull out all of the fruit. If there are only a few items per bag, a literal GROUPBY function is easier with smaller bags as you can quickly see what is in each bag. Conversely, if you were to have the bags with many items in them, the more bags you have, the longer it is going to take you to go through each one. If there were only two or three bags with many items, it would not be a large task compared to the other method. However, in the same way that large data sets continue to grow, if you now have to look through fifteen or twenty bags, it might take significantly longer just in processing time.

Is this a textbook case of *a priori*? It seems absolutely foolish to contain too much information in a record. One would be giving adversaries much more data to work off of and not making them look for it. But is this the real world scenario? Perhaps the answer is in the nature of the data. Sparse data has been adopted as much safer than comprehensive, and to us, it makes sense. Data engineers are then tasked with comprehending 1) what they know about the data relationships, and 2) what an adversary might know. If relationships are hard to draw conceptually, sparse data is more robust to attacks. However, if each tuple is simple in nature, such as purchases for an online retailer, a company would just need to employ additional measures to further mask the data. Knowing that an anonymous users high blood pressure diagnosis is linked to depression is perhaps a harder reach than, say, knowing someone who purchased milk most likely also bought cereal, childrens foods, or beer on the same visit.

Of course, armed with additional, outside data, such as credit card transactions for a certain zip code, this situation become even more vulnerable. Regarding the blood pressure scenario from above, perhaps an adversary knows from this outside data, the purchase amount at a pharmacy. He or she sees that this amount is very large, and the patient most likely picked up additional items. Even armed with this limited information, an adversary could begin formalizing guesses at to other needs this person had. Indeed, Narayanan and Shmatikov proved how adversaries knowing very little of the data, or even what they were looking for, could use these generalities to completely re-identify a user [1].

Perturbation

Another method used to anonymize data is to actually change the attributes enough to create further distance from a given source, but not so much as to statistically alter data science efforts. Common examples are to create ranges and categories out of numerical fields, or to use internal identifiers instead of more publicly recognizable attributes. In fact, perturbation is also used as a way to actually detect data leaks, based on the theory that if a particular recipient is given a data set that was perturbed with a certain dictionary of identifiers, and this information is found somewhere it should not have been, you can determine which party released the data since there is no other recipient that had that particular set of identifiers.

Common examples of perturbation include:

* Removing Personal Identifiable Information (PIIs) and replacing with an ID number.

* Taking a scalar value and applying ranges. Instead of ages, the owner of the data could use age ranges that mask the exact answer.

* Selectively altering or even deleting data to create doubt in a bad actor. If a malicious party has, say, matched up a perturbed data set with another set found online, the principle here is that they have a reduced chance of correctly determining a subject's attribute values. * Changing the definition of certain attributes. For example, if the need for a data set is to review density of certain attributes, then it does not matter if the values reflect the original values. As long as the correct scale is used, one could change, say, zip codes to names of fruits and vegetables. Of course, this might cause more confusion than value depending on the purpose of the data mining activity, so the owner of the data would need to know the purpose of a data scientist to correctly perform this technique.

* Use a "watermark" to discourage data leaks. Watermarking involves altering the data to embed a secret code per subset / data set that is given out. This helps a data source know who leaked a data set because each recipient received a different watermark.

Of course, no method is perfect, and neither is perturbation. If one does not consider *why* a data set is needed, perturbing certain fields could make the data unusable. The number of fields released directly affects the strength of a perturbed attribute. 500,000 records with their ages changed to age ranges is very different from 100 records. The more a malicious attacker would has to work with, the greater the chance that he or she can figure out a "secret" alteration method, especially if that field is now a category where there are not as many "wrong answers". Remember, an attacker

does not necessarily have to be exactly right in their guess. Even having a broad guess of where someone might fall has been proven dangerous to the confidentiality of a data set, as evidenced by the Netflix prize fiasco [1]. Finally, watermarks are not effective if an attacker has discovered how to delete it.

Perturbation is certainly a way to cause doubt for an attacker, but it must be taken into consideration with other factors. The size of the data set, distribution of the various attribute values, and the confusion it could cause for a data scientist indicate that additional techniques and factors need to be considered before this method is held up as the solution to a problem.

Deletion of Data

Similar to perturbation, deletion of data alters a data set, if not completely eliminates it. One might wonder what the point is, since deleting a data set would effectively make it anonymous, but there are interesting results from this method that warrant further explanation.

Firstly, virtual deletion of part of a data set is routinely performed by creating subsets. Here you limit what is given out to reduce the chances of identifying a subject. This is, of course, a sparse data set technique. However, what if an entity actually truly deleted attributes permanently in the interest of protecting more sensitive fields?

For example, say there is a medical data set that held information showing subjects' zip codes, ages, names of hospitals visited, number of days they were in the hospital, and the diagnostic codes of whatever they were were in the hospital for. Is is imperative that we have a subject's zip code? Is knowing the hospital visited essentially the same thing and would this be enough for the quesion at hand? But does knowing the hospital *likely* to give away a zip code anyway, in which case perturbation of the hospital name would be necessary?

To make matters more challenging, permanent deletion of data, especially partial, can create a new host of problems for databases. Indexing can be disturbed if too many records are deleted, resulting in energy waste as a database interface searches through empty locations. Keys (or lack thereof) also present a problem. Deleting a field such as a social security number might seem like a reasonable idea, but if that field serves as at least one primary key in any of the database records, the results could be catastrophic.

That being said, in an age where data leaks have put consumers at risk and hurt various companies bottom lines (think Yahoo!'s reduced purchase price following a data breach [11] , is deletion of data actually a good move to create good will? While "Big Data" seems to be at the forefront of many companies' minds, many entities seem to have an inability to handle it against malicious attacks. Perhaps a company could receive more value from promoting that they *delete* data instead of store it. This would take a commitment and understanding well in advance of a database design to avoid operational impediments, or simply enough flexibility that a corporation could actually do this without putting themselves at more risk of net lost sales or decreased operational efficiency.

2 De-anonymization

To gain a better understanding of how effective various anonymization techniques are, our team explored de-anonymization methods as well. We did this by reading existing papers and articles and by actually testing certain de-anonymization processes on our sample data set. The following points are worth noting as background knowledge before any type of de-anonymization.

***Assumptions of an adversary's knowledge:** a bad actor might have only general knowledge about the fields in a data set. It is not impossible for someone to explore a data set and find likely identifiers with what they deem as a satisfactory level of confidence. Therefor, just because an exact value cannot be determined by an attacker, a general idea might be enough for their uses.

Acceptance of the power of data: a simple gut check of "what could one possibly get out of this data?" is not a valid test against the vulnerability of a data set. As evidenced by the Netflix prize de-anonymization, Narayanan and Shmatikov were able to take seemingly innocuous information and tie it back to names individuals. [1] To assume that data is not "important enough" or too "vague" could be considered gross negligence in this age of digital security enlightenment.

Acknowledgment of the repercussions of shared data: it is important to know that implications exist for subjects whose data is known by others. Being able to identify who someone is based on their movie reviews may seem harmless, but in the wrong hands, it could be used to punish someone for their political or religious views. For those who would balk at the weight of such a discovery, the ability to then match up names with another publicly available data set, could truly make this situation a nightmare for a company.

Additionally, once an attacker has at least something to work with, even just a name, there is a myriad of other activities one could perform to gain additional knowledge on a data set. For example, an attacker can make physical phone calls or send electronic messages to various parties to retrieve additional knowledge on a very small piece of information they garnered from a dataset. They can then return to that same data set or another data set with this more complete picture of a subject's life to find out additional information. In other words, an attacker's knowledge of a data set is not necessarily limited to mining data sets.

Algorithms can be your enemy: for an adversary with limited knowledge about the subjects or data set, an algorithm can speed up their investigation and narrow down the list of possible values. In fact, it was an algorithm that helped Narayanan and Shmatikov gain significant progress in connecting subjects with their data. For those adversaries who are not working with much information to start with, but who have an effective algorithm, they are no longer looking for a needle in a haystack.

2.0.1 Testing of De-anonymization on Our Data Set

How our data set might be compromised by an adversary
i. real-world implications of re-identification
ii. least likely and most likely de-anonymization risks

3 Recommendation

It is not reasonable to avoid data collection and sharing altogether. Thus, we have compiled a series of recommendations that we would advise to those interested in having a better realization of the remaining vulnerability of their anonymization techniques.

1. Actively attempt to de-anonymize data sets. Depending on the sensitivity of the data, this might need to be performed by individuals not privvy to the anonymization processes used.

2. Put aside personal bias, and spend time exploring the danger of a data set's attributes being known. A "once over" is not enough. Seemingly innocuous data can actually be harmful if in the wrong hands.

3. Consider using multiple anonymization techniques. Test along the way to see how this affects statistical analysis.

4. Work to understand the goal of a data mining activity before providing data sets. Consider using subsets, and weigh the the value of data mining activities with the vulnerability of the subjects' information.

5. Consider the long range goals of an organization before employing the use of public contests. Will you even need this winning algorithm or model in the long term?

4 Conclusion

Where our amazing conclusion will go.

Appendix A

placeholder

Appendix B

placeholder

Acknowledgments

The authors would like to thank...
The IEEE for this template, TRANS-JOUR.DOC

References

[1] Narayanan, Arvind and Vitaly Shmatikov *Robust De-anonymization of Large Sparse Datasets*. SP 08 Proceedings of the 2008 IEEE Symposium on Security and Privacy. 2008.
[2] Johnston, Casy. *Netflix Never Used Its $1 Million Algorithm Due to Engineering Costs*. Wired Magazine. https://www.wired.com/2012/04/netflix-prize-costs/. April 2012.
[3] Anderson, Nate. *Neflix Prize 2 (Privacy) Apocalpyse Now?* ars Technica. https://arstechnica.com/tech-policy/2009/09/netflix-prize-2-privacy-apocalypse-now/?comments=1. September 2009.
[4] Ohm, Paul. *Netflix Cancels the Netflix Prize 2*. Freedom to Tinker.com. https://freedom-to-tinker.com/2010/03/12/netflix-cancels-netflix-prize-2/. March 2010.
[5] Cheng, Jacqui. *Netflix settles privacy lawsuit, ditches $1 million contest*. ars Technica. https://arstechnica.com/tech-policy/2010/03/netflix-ditches-1-million-contest-in-wake-of-privacy-suit/?comments=1. March 2010.
[6] Borne, Zachary Davies. *There Are Officially More Mobile Devices Than People in the World*. Independent.co.uk. http://www.independent.co.uk/life-style/gadgets-and-tech/news/there-are-officially-more-mobile-devices-than-people-in-the-world-9780518.html. October 2014.
[7] Loukides, Grigorios Aris Gkoulalas-Divanis, and Bradley Mali. *Anonymization of electronic medical records for validating genome-wide association studies*. Proceedings of the National Academy of Sciences of the United States of America. Vol. 107, No. 17, pp 7898-7903. April 2010.
[8] Gentry, Jerry. *Big Data vs. Sparse Data*. Data Center Knowledge. http://www.datacenterknowledge.com/archives/2012/10/11/big-data-versus-sparse-data/. October 2012.
[9] Rimes. *The difference between "dense" and "sparse" data*. . Rimes.com. https://www.rimes.com/insights/the-difference-between-dense-and-sparse-data/. February 2012.
[10] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
[11] Albanesius, Chloe. *After Breach, Verizon Drops Yahoo Purchase Price by $350M*. PC News. 1em plus 0.5em minus 0.4em. February 2017.