A Comparison of Input and Output Perturbation based on Statistical and Computational Notions of Privacy

Rathindra Sarathy

Krish Muralidhar

Contents

- Statistical notions of Privacy & Utility
- Computational Privacy & Query-Response Utility – Dinur & Nissim
- Comments on the Main results of D&N
- Input Perturbation and the D&N Framework – the CD Model
- Input vs Output Perturbation – Three Questions
- Analyzing Input vs Output Perturbation using the D&N Computational privacy Framework
- Computational notions of privacy and perturbation for specific statistical utility
- Summary Comments on Statistical and Computational notions of Privacy
- Conclusions

Statistical Notions of Privacy

- SDL (Statistical Disclosure Limitation) Literature has two fundamental notions: identity and value disclosure
- Best framework – Fuller (1993)
  - Given a de-identified data set where the confidential values are masked, an intruder would first try to identify a record in the released data set as belonging to an individual (identity disclosure) and subsequently attempt to estimate the confidential values (value disclosure).
  - Intruder knowledge is modeled as knowledge of the true values of a few of the masked confidential variables
  - Useful in evaluating the efficacy of a masking technique
  - Does not incorporate a query-response framework – i.e., released data maintains a few statistics closely but does not attempt to answer every question well

Computational Notions of Privacy

- Framework – Dinur & Nissim (D&N)
- Given a database with confidential values (WLOG – binary), modify every query response with added (bounded) noise
- Does not preclude static data release where a perturbed data set would give perturbed answers when "queried"
- "privacy is violated if an adversary is capable of computing a confidential attribute d_i *from its identity i*" - Privacy is preserved if it is computationally infeasible to do so
- D&N contend that statistical definitions are inadequate because:
  - "Firstly, it is not clear that large variance necessarily prevents private information from being leaked.
  - Secondly, this kind of definition does not allow us to capitalize on the limits of an adversary."
- More on these later.

Computational Notions of Privacy

- Initial results are for intruder with no prior information, but with different levels of computing power
- Define *non-privacy* – *"*if a computationally-bounded adversary can expose a $1 - \varepsilon$ *fraction of the database* entries for all constant $\varepsilon > 0$*."*
- Unclear what "expose" means – appears to suggest that the computational attack would result in $1 - \varepsilon$ *fraction of the database's* identified entries to be known with certainty
- However, a closer look at results appear to suggest otherwise

Computational Notions of Privacy (continued)

- "Our main result is a polynomial reconstruction algorithm from noisy subset-sums, that – in case the answer to queries are within additive perturbation error $\xi = o(\sqrt{n})$ – *succeeds in reconstructing a 'candidate' database c whose* Hamming distance from *d is at most o(n). In particular, c agrees with d on over 99% of their bits.*"
- Appears to be an impressive compromise at first sight, *but* for most reasonable sized databases there can be a multitude of candidates that are within a very small Hamming distance

Computational Notions of Privacy (continued)

- Consider exponential adversary: Has computing capacity for exponential number of queries
- Regardless of the intruder's prior information (that is regardless of the distribution from which (d) is drawn) D&N result is that we can always find a candidate (c) whose Hamming distance from the true database (d) is such that dist (c,d) < $\varepsilon n <= 4 \xi$ for $\varepsilon > 0$ when the following algorithm stops:
    - *Start with a candidate database* (c)
    - *Issue all queries and check if (True response – perturbed response) > ξ for any query*
    - *If yes, try the next candidate. Else*, STOP
- Algorithm guaranteed to stop because true database (d) is one of the candidates
- Since (c) and (d) will only differ in at most εn bits, $n(1- \varepsilon)$ bits are "exposed"
- *But is it disclosure?*

Computational Notions of Privacy (continued)

- Hamming distance only tells us that the "strings" (d) and (c) differ in at most 4ξ positions
- Cannot identify the exact positions in which they differ
- Example:
  - d = (0 0 0 1 1) & (ξ = 1)
- There are 31 different candidate databases within this Hamming Distance of 4 for n=5. The intruder cannot be any more certain about the true database
  - If n = 1,000,000 and ξ = 10, there are more than 10^{192} potential candidates (c) that are within Hamming distance 4 ξ
  - It is true 10^{192} is much smaller than $2^{1000000}$, but not of much use to the intruder either in terms of knowing which is the true database

Computational Notions of Privacy (continued)

- *So where is the "Disclosure"?*
- Note that privacy is considered violated if we "disclose" the value of the ith observation computationally. The implication is that the value will be revealed with certainty.
- But we cannot do that with a single candidate (unless it is the only candidate) because we will have some uncertainty, no matter how close the candidate is to the true database.
- If we take a guess about the ith observation based on the candidate database value, we are very likely to be right but not 100% correct always.
- The only way to compromise n(1- ε) bits with certainty is to *obtain all possible candidates that are that* εn *Hamming distance away. Then, values for observations common to all candidates are revealed with certainty.*

Computational Notions of Privacy (continued)

- But.. enumerating all candidates is not computationally feasible even for moderately sized databases. The number of candidate databases even for small perturbations will be large and the number of queries will rise exponentially.
- *We can view the D&N results as relating the perturbation level ξ and computational power of the intruder to bounds on the probability of a correct guess of the i^{th} value*. Privacy should be expressed in terms of the bounds on this probability, rather than number of bits "exposed".
- If so, it is the same as the familiar probability of disclosing the true value in SDL. Only the means of arriving there are different (statistical inference vs computing subset sums)
- Alternatively, we can use the reciprocal of the number of candidate solutions for a particular Hamming Distance as a risk measure for exact disclosure of the whole database – again equivalent to measures used in traditional SDL

Other D&N Results for Output Perturbation

- Regardless of perturbation method and regardless of intruder's prior information (that is regardless of the distribution from which d is drawn) a Polynomially Bounded adversary can obtain a candidate solution c with high probability in polynomial time (i.e., needs only polynomial number of queries) such that dist(c,d) < εn if ξ = o*(√n)* (Main Result)
- Although at first glance it seems like a large perturbation, *√n* is actually very small for very large databases. For a million records, this is the equivalent of modifying just over 1000 records or 0.1% of the entire database. So the perturbation level is VERY SMALL.
- If the database is uniformly distributed "over all strings of *n* bits", that is the intruder *has no prior information, p*rivacy is guaranteed if ξ = Õ(n) but that "renders the database effectively useless – users are extremely unlikely to get any non-trivial information by querying the database, and hence they are unlikely to compute any non-trivial functionality of it."

Computational Privacy & Statistical Notions

- As just discussed, computations result in probabilistic and not certain disclosure (no different from SDL). But, D&N suggest by not taking into account the computational power of the adversary SDL folks have missed something. Our argument against this is two-fold:
- An apparent steep increase in disclosure through computations is attributable either to
  1. very little perturbation relative to utility *(every* query must be answered within a relatively small distance from the true answer) and computational power, and to some extent (price of high utility),
  2. the choice of perturbation method (output perturbation).

1) The Price of High Utility

- Traditional SDL methods assume that only a few statistics (computations) are answered well and no bounds are placed on the errors for other (non-essential) computations.
- The D&N results rely heavily on *every* answer being with known and relatively small error bounds (ξ).
- Under traditional SDL assumptions above, D&N type computations will not result in any substantial privacy violation (privacy as defined in D&N).
- In other words "capitalizing on the limits of the adversary" requires perturbation schemes in a query-response framework that guarantees high utility
- Typical data releases (even microdata releases) by government agencies face no substantial threat by intruders even with substantial computing power. They need to worry more about statistical threats that are model-based attacks.
- D&N hint at this:
  - If queries are answered such that they are within $\xi = \tilde{O}(n)$ for *most* queries then it is possible to guarantee privacy even if perturbation is $\tilde{O}(\sqrt{n})$ (better than $\tilde{O}(n)$) so that some utility is achieved
- Thus, we need to be careful about the context where computational attacks are relevant. Access to query responses where the quality of the response to every query is known and guaranteed (such as in remote access databases) is certainly a situation where D&N results are relevant

2) Choice of Perturbation Method Input Perturbation & the CD model

- D&N claim that their main results are oblivious to intruder's prior information *and to the perturbation method* (Input or Output).
- CD Model – The basis for extending output perturbation results to input perturbation
  - Create a private version (d') of the database d and answer all queries from d'
  - As long as the answers to all queries are within the bounds guaranteed (i.e., utility is the same as an output perturbation algorithm), there is no *apparent* difference between input and output perturbation
  - Since a user may retrieve d' by simply querying each observation it is equivalent to releasing a CD of perturbed data to the user

Input vs Output Perturbation

- Question:
  - Are input and output perturbation really equivalent even under the D&N scenario? That is, do they lead to the same candidate solutions or different candidate solutions with different characteristics?

Some Definitions

- a_q – True response to a query
- $\widetilde{a_q}$ – Perturbed response to the same query
- σ represents the amount of input perturbation
- *d* represents the original database
- *c* represents the candidate database from D&N algorithm
- e represents the input perturbed database
- ξ represents the maximum perturbation
- $\left|a_q - \widetilde{a_q}\right| \leq \xi$ for all queries *q*

Input vs Output Perturbation – Candidate Solutions

- Input Perturbation CD Model:
- There are multiple input perturbation schemes possible that can guarantee that all responses from all queries are with ξ of the true response (D&N present one such scheme; other schemes are possible as below).
- To ensure equivalence between input and output perturbation in terms of security measured by Hamming distance, we choose the following scheme: Randomly change ξ 0's in (d) to 1 and ξ 0's in (d) to 1 to get the perturbed database (e).
- Then we can formulate the problem as follows:
  - $|e_i - d_i| \leq \xi$ when $d_i = 0$ and $|e_i - d_i| \leq \xi$ when $d_i = 1$
  - $e_i = [d_i + \sigma_i] \in [0, 1]$ and for all queries q, $|a_q - \tilde{a}_q| \leq \xi$
  - For this formulation we will find that the only binding constraint will be:

    $\sum_{1=1,\ldots,n} |(\sigma_i)| \leq 2\xi$
- The perturbed database (e) will be ≤ 2ξ from the true database (d) due to the binding constraint. When the D&N algorithm is applied to the released input perturbed data e, every candidate solution c obtained will be ≤ 4ξ from (d) and (e).

Input vs Output Perturbation – Candidate Solutions

- It may be argued that both input perturbation and output perturbation are subject to the same disclosure since both result in candidate solutions that are the same maximum distance away from the true database.
- There is one important difference between Input and Output perturbation
  - With output perturbation, it is possible that the *number of candidate solutions is very small*. When ξ is very small, it is possible that there is only one candidate solution … EXACT DISCLOSURE
  - With input perturbation, we are always guaranteed a specific minimum number of candidate solutions … EXACT DISCLOSURE NEVER OCCURS

Input vs Output Perturbation – Candidate Solutions

- Why does this happen?
  - With input perturbation there is only one binding constraint as seen earlier, permitting a full set of candidates (all candidates within 4ξ Hamming distance from (d)) that cannot be eliminated.
  - With output perturbation, independent noise is added to the true sums. Every sum query represents a constraint that is independent of the other constraints. When the maximum amount of noise is bounded, each perturbed sum constrains the values of the true sums, resulting in the elimination of candidate databases. In the worst case, we may end up with a single candidate which would be the true database.
    - For example, the following simulation involving every possible database of 10 observations and employing all possible (1023) queries (that can be easily replicated) shows the following:
      - $\xi = 1$, Unique candidate solutions = 1022 (only all zero's and all 1's have multiple solutions)
      - $\xi = 2$, Unique candidate solutions = 1002
      - $\xi = 3$, Unique candidate solutions = 902
      - $\xi = 4$, Unique candidate solutions = 482
      - $\xi = 5$, Unique candidate solutions = 0
  - Thus, while there may be no difference in computational security performance measured in terms of Hamming distance of the candidates from the true database, if we consider the number of potential candidate solutions as a measure of security, input perturbation would be preferred, for the same level of utility.

Comparison of Input and Output Perturbation

Output Perturbation		Input Perturbation		
ξ	Number of Unique Solutions	ξ	Number of Unique Solutions	# of Candidate Solutions
1	1022	1	0	36
2	1002	2	0	256
3	902	3	0	676
4	482	4	0	961
5	0	5	0	1024

- Thus, while there may be no difference in computational security performance measured in terms of Hamming distance of the candidates from the true database, if we consider the number of potential candidate solutions as a measure of security, input perturbation would be preferred, for the same level of utility.

Input vs Output Perturbation – Disclosure risk

- It must be noted that we have assessed both input and output perturbation in a query-response framework

Conclusion

- Is there reason for pessimism? Has the computational privacy literature exposed a deep flaw in SDL methods that statistical definitions of privacy have missed? We contend otherwise for the following reasons.

1. Computational privacy only underscores the already well understood tradeoff between privacy and security. You cannot eat the cake (have good utility for every query) and have it (privacy) too.
2. If ξ is relatively small and every query is answered, then by definition there can be little that masking techniques can do.
3. Even under the pessimistic privacy scenario of small ξ, employing even an exponential adversary, only probabilistic disclosure is possible using the D&N attack for input perturbation. A single candidate cannot result in knowledge without certainty and therefore does not violate privacy under the following original definition from D&N: "privacy is violated if an adversary is capable of computing a confidential attribute d_i *from its identity I*". The D&N attack can only provide a probability for the value of d_i from a single candidate. There is not 100% certainty about the value of any bit.
4. Even under a more realistic computational power for the adversary, non-privacy can be achieved with a relatively small perturbation that is greater than o*(√n).*