Workshop Week 10 - COMP20008 2017

Questions

1. Consider the 3 party protocol for privacy preserving linkage with exact matching, discussed in lectures.

   - What is a salt? An extra string that is appended to the information being encoded, so that hashed value is not susceptible to a dictionary attack. The two parties doing the linkage would agree on a salt, the 3rd party would not know it.
   - Explain why a salt is used. It is used to prevent a dictionary attack by a third party.
   - Who chooses and knows the salt? The two parties A and B who are integrating their data choose and know the salt.
   - What assumptions should be made about the level of trust required for C? C does not know the salt.

2. From lecture notes

3. A bloom filter is used to store the set of 2-grams from a string. Two strings are then compared for similarity by computing the Dice coefficient for their respective bloom filters (formula in Lecture 19) $sim(b1, b2) = \frac{2h}{b1+b2}$. Consider the following two alternative similarity measures that might be used. Explain their advantages/disadvantages compared to the Dice coefficient for evaluating bloom filter similarity.

   - Hamming similarity: $sim(b1, b2) = s/l$, where s is the number of bits which are the same in $b1$ and $b2$ and l is the bit vector length. Hamming similarity: overestimates the similarity artificially due to the matching of the 0's
   - Jaccard similarity: $sim(b1, b2) = \frac{h}{l}$ where l is the bit vector length and h is the number of bits set to 1 in both bloom filters.. Jaccard similarity: Underestimates the similarity artificially due to the division by the number of bits l

4. For bloom filters of length l and using k hash functions. Consider the ratio l/k.

   - As l/k increases, would you expect the matching accuracy of the system to become better or worse? Would you expect the robustness of the system to frequency attack (by the trusted 3rd party) to become better or worse as l/k increases? Why?

     The ratio results in sparse or less sparse representations. When the representation is sparse then the privacy is lower and accuracy of membership queries is increased, since there are less collisions. The speed is also slower (more bits to scan). Consider the opposite extreme case when the the number of bits is small and we have a large number of hash functions. Then the bloom filter will consist of mostly ones. The privacy is high since we can't determine the original records, however the accuracy of membership queries is very low. The filter is relatively faster because there are fewer bits.

5. Suppose a bank wishes to perform data linkage to match the customers in its loan application database, against public twitter feeds (to help the bank more accurately assess customer risk).

   - Based on your knowledge of Twitter, how feasible do you believe this would be?
   - What legal and ethical issues could be relevant here?

   Discussion question.

6. The question about Phase 4 is a discussion question.