Poster: Autocompletion in Local Differential Privacy

Johnnie C-N. Chang
Computer Science Department
University of California, Santa Cruz
cchan60ucsc.edu

Abhradeep Guha Thakurta
Computer Science Department
University of California, Santa Cruz
aguhatha@ucsc.edu

***Abstract*—Solving problems with sensitive real-world data collected from individuals has raised privacy concerns in the recent years [1] [2]. In this work, we approach the problem of autocompletion in a setting with local differential privacy.**

***Index Terms*—differential privacy, frequency statistics, natural language processing**

I. Introduction

In this work, we study the problem of autocompletion in the local differential privacy model, where data from each individual device is privatized before being collected to a central location to train an autocompletion model. In particular, we use a model that predicts with lexical as well as syntactic information. As natural language is a highly sensitive type of data, we want to provide a solution that gives accurate predictions without undermining the privacy of the users. We approach the differentially private autocompletion task with an existing language model Tags-and-Words [3]. Tags-and-Words improves vanilla N-gram models, which model a given word conditional on $N-1$ previous words, by incorporating syntactic information, using the frequencies of both lexical and syntactic tokens. The token frequencies are collected from users of devices such as cellphones, in a manner that preserves local differential privacy such that raw user information never leaves their device. With a very large domain such as word N-grams, this would require a tremendous amount of space to store the frequencies if we do not tackle it gracefully. We hence use Count Sketch, a sketching algorithm, to significantly reduce the space required with little compromise to accuracy.

II. Background

A. Local differential privacy

In the wake of the age of big data, analytics and learning on data is in increasingly strong demand. Specifically, analyzing user data can often provide insight into trends in user behaviors and assist decision making to provide better service. However, many types of user data may attract privacy concerns due to their sensitive nature, including natural language data, health data and financial data, and approaches to protect privacy is hence of strong interest to both the academia and the industry.

Differential privacy is a mathematically rigorous notion of privacy that provides a guarantee on the amount of difference an individual can cause, whether or not they are part of the dataset. Differential privacy is considered in two contrasting settings, the central model and the local model. In the central model, raw data is centralized and stored in a silo to which queries are made. The validity of this model depends on trust on the data manager to protect the privacy of the data in answering the queries. In contrast, in the local model, only privatized data is collected into the a dataset, and raw data of an individual would never be released. Formally, local differential privacy is defined as below.

Definition II.1. (Local Differential Privacy) An algorithm satisfies ϵ-local differential privacy (LDP) if it accesses the database $\mathbf{D} = (d_1, ..., d_n) \in \mathscr{D}^n$ only via invocations of a local randomizer R, and if $R^{(1)}, R^{(2)}, ..., R^{(k)}$ denote the algorithm's invocations of R on the data sample v_i, then the algorithm $A(\cdot) \triangleq R^{(1)}(\cdot), R^{(2)}(\cdot), ..., R^{(k)}(\cdot)$ is ϵ-differentially private. That is, if for any pair of data samples $\mathbf{D}, \mathbf{D}' \in \mathscr{D}$ and $\forall S \subseteq \text{Range}(A)$, $\Pr[A(\mathbf{D}) \in S] \leq \exp(\epsilon) \Pr[A(\mathbf{D}') \in S]$.

While LDP provides stronger privacy guarantee than the central model, it has the challenge of being more expensive computationally and statistically. Indeed the two settings have a trade-off with respect to accuracy and privacy.

B. Autocompletion with syntactic information

1) Motivation: Autocompletion is a Natural Language Processing (NLP) task where, having observed a number of preceding words entered by a user, we predict a number of the next words. Solution to this problem has a wide range of applications, such as instant messaging and web queries. In particular, systems capable of autocompletion could reduce keystrokes and typing errors, as well as help users recall tip-of-the-tongue words, hence improving the efficiency of human-computer interaction. Traditionally, this problem is tackled with a statistical model, computing the conditional probability of the next word w_i given a sequence of words $w_{i-N}, ..., w_{i-1}$ preceding w_i, $P(w_i \mid w_{i-N}, ..., w_{i-1})$. This is referred to as the N-gram model, where $N-1$ is the number of preceding words considered, and is often used as a baseline method.

In this work, we want to provide accurate predictions, using not only the word tokens but also syntactic information of the observed sequence, in our case represented with part-of-speech tags (POS tags). By integrating syntactic information into the model, predictions can be made with higher accuracy [3].

2) Problem definition: Given a word sequence $\mathbf{w}$ of arbitrary length and a sequence $\mathbf{t}$ of corresponding tags, we define

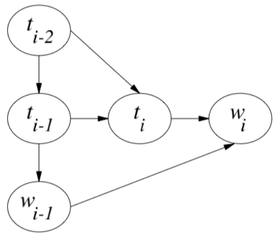

Fig. 1. Bayesian network representing the conditional independence between words and POS tags.

the task of autocompletion to be predicting w_i, the i^{th} word in $\mathbf{w}$, using subsequences prior to w_i and t_i:

$$w_i = \arg\max_{u \in W} \Pr[u | \mathbf{w^{i-1}}, \mathbf{t^{i-1}}]$$

where W is the set of all words, $\mathbf{w^j}$ is the sequence $w_1, ..., w_j$, and $\mathbf{t^j}$ is the corresponding sequence of tags.

III. Locally private autocompletion

A. Algorithm descriptions

1) Autocompletion language model: Tags-and-Words: For our auto-completion task, we use the algorithm Tags-and-Words [3] (TAW), which computes statistical estimates of $\Pr(u \mid w_{i-1}, t_{i-1}, t_{i-2})$ using frequencies of words and POS tags with an assumption of conditional independence described by the Bayesian network in Figure 1. We can hence rewrite $\Pr(u \mid w_{i-1}, t_{i-1}, t_{i-2})$ as below, where $T(u)$ is the set of all tags for a word w, such that u can be modeled by tokens of word 2-grams, tag 3-grams, tag probability and conditional probability of a tag given a word:

$$\Pr(u \mid w_{i-1}) \times \sum_{t_i \in T(u)} \frac{\Pr(t_i \mid u) \times \Pr(t_i \mid t_{i-1}, t_{i-2})}{\Pr(t_i)}.$$

2) Space efficient counting: Count Sketch: Count Sketch is shown in [4] to achieve high counting accuracy estimates while providing significant space savings, an essential component to the autocompletion model described in III-A1. The set up of the algorithm includes a matrix $M = \{0\}^{\ell \times w}$ and ℓ pairs of hash functions, $h_i : \mathscr{D} \mapsto [w]$ and $g_i : \mathscr{D} \mapsto \{+1, -1\}$ for $i \in [\ell]$. For each record d, we update M as follows: $\forall i \in [\ell], M[i, h_i(d)] = M[i, h_i(d)] + g_i(d)$. When one queries for the frequency of any d, $\text{median}_{i \in [\ell]} M[i, h_i(d)]$ is returned. Count sketch guarantees that, with probability at least $1 - \beta$, using a data structure of size $m = O\left(\sqrt{n} \log\left(\frac{1}{\beta}\right)\right)$, the error of frequency estimate of any given $d \in \mathscr{D}$ is at most $\sqrt{n}$.

B. Empirical evaluation

Consider typing with a keyboard on a mobile device which suggests a list of possible next words after a word is typed. Ideally, the most likely word would be the first in the list, with each word after that decreasingly likely. We evaluate the efficacy of our LDP syntactic autocompletion by constructing such a ranked list of words given previous tokens, and comparing it with a ranked list constructed without privacy.

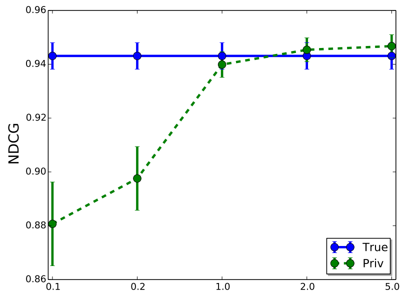

Fig. 2. NDCG of true and private predictions versus ϵ.

1) Corpus: We use tokens from the Brown news corpus in NLTK [5] to compute the TAW model. The corpus contains just over 1 million words after preprocessing (removing punctuation and setting all words to lowercase), where roughly 41 thousand are unique, distributed over 57 thousand sentences. We obtain the POS tags of each sentence using the POS tagger that NLTK provides.

2) Methods: In order to show how closely predictions with LDP (private predictions) resembles those without privacy (true predictions) empirically, we compare the Normalized Discounted Cumulative Gain (NDCG) of the true and private predictions. Discounted Cumulative Gain (DCG) is defined as follows: $\text{DCG} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$, where rel_i is the relevance of the i^{th} document, which is the predicted probability of acceptance of each potential next word. Since the value of DCG grows with the number of words to rank, to effectively compare prediction results of different inputs we must obtain NDCG by normalizing DCG by Ideal DCG (IDCG), the maximum DCG one could achieve and is hence the DCG of the ground truth, $\text{NDCG} = \frac{\text{DCG}}{\text{IDCG}}$.

3) Experiment Results: Figure 2 shows experiment results as we vary the privacy parameter ϵ. Each experiment is averaged over 5 runs. The solid line for True refers to the NDCG values of the true predictions, and the dashed line for Priv refers to that of the private predictions. The error bar accounts for randomness introduced in the Count Sketch algorithm as well as privacy (in the case of private predictions). We observe the following:

1) The error in the private prediction score decreases significantly as the privacy parameter ϵ increases.
2) At $\epsilon = 1$ the private prediction score becomes indistinguishable from the true score.

C. Future directions

In this work we provide solution to privacy preserving space efficient syntactic autocompletion, but to be able to deploy this model in user devices, we would also need to improve communication cost for uploading privatized user data to the server. This can be done with Hadamard transform [4]. In addition, in order to provide more accurate predictions, we may want to consider state-of-the-art machine learning algorithms for autocompletion, such as neural networks.

References

[1] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 111–125.

[2] A. Korolova, "Privacy violations using microtargeted ads: A case study," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. IEEE, 2010, pp. 474–482.

[3] A. Fazly and G. Hirst, "Testing the efficacy of part-of-speech information in word completion," in *Proceedings of the 2003 EACL Workshop on Language Modeling for Text Entry Methods*. Association for Computational Linguistics, 2003, pp. 9–16.

[4] R. Bassily, U. Stemmer, A. G. Thakurta *et al.*, "Practical locally private heavy hitters," in *Advances in Neural Information Processing Systems*, 2017, pp. 2285–2293.

[5] S. Bird, "Nltk: the natural language toolkit," in *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, 2006, pp. 69–72.