Leveraging Stratification in Twitter Sampling

Joshi, V., Padmanabhan, D., & Subramaniam, LV. (2016). Leveraging Stratification in Twitter Sampling. In *ECAI 2016* (Vol. 285, pp. 1212-1220). (Frontiers in Artificial Intelligence and Applications). IOS Press. https://doi.org/10.3233/978-1-61499-672-9-1212

Published in:
ECAI 2016

Document Version:
Publisher's PDF, also known as Version of record

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1212

Leveraging Stratification in Twitter Sampling

Vikas Joshi[1] **Deepak P**[2] **L V Subramaniam**[1]
[1]**IBM Research - India** [2]**Queen's University Belfast, UK**
vijoshij@in.ibm.com deepaksp@acm.org lvsubram@in.ibm.com

Abstract. With Tweet volumes reaching 500 million a day, sampling is inevitable for any application using Twitter data. Realizing this, data providers such as Twitter, Gnip and Boardreader license sampled data streams priced in accordance with the sample size. Big Data applications working with sampled data would be interested in working with a large enough sample that is representative of the universal dataset. Previous work focusing on the representativeness issue has considered ensuring that global occurrence rates of key terms, be reliably estimated from the sample. Present technology allows sample size estimation in accordance with probabilistic bounds on occurrence rates for the case of uniform random sampling. In this paper, we consider the problem of further improving sample size estimates by leveraging stratification in Twitter data. We analyze our estimates through an extensive study using simulations and real-world data, establishing the superiority of our method over uniform random sampling. Our work provides the technical know-how for data providers to expand their portfolio to include stratified sampled datasets, whereas applications are benefited by being able to monitor more topics/events at the same data and computing cost.

1 Introduction

Microblogging sites have seen massive penetration over the last many years. The importance of microblogging as a social signal is immense in this age when Twitter has been shown to be useful in the context of natural disasters[17] and political uprisings[10]. The usefulness of the data has led to new business models to monetize social media data. Data providers like Twitter, Gnip and Boardreader provide access to data through different application programming interfaces (APIs). They constantly innovate with pricing models to sell data. With the number of tweets generated daily measuring as much as 500 million[1], massive computation infrastructure is needed to analyze such big-data. In order to expand the customer base to include small-scale Twitter intelligence applications who have limited compute infrastructure and capital, data providers offer (uniformly) *sampled* data streams. For example: Twitter provides three popular sampled APIs namely: *Powertrack API*, which returns all the data for the given keywords at a higher base cost, *Decahose API*, returns 10% of entire data (uniformly and randomly sampled) at lower cost than Powertrack API, *Free 1% API* which is 1% of the entire data stream and is free of cost. In the quest to enrich the sampling portfolio without compromising on probabilistic guarantees, we study the use of stratified sampling to improve sample size estimates. Leveraging stratification can improve the quality of the sample by providing one of (a) tighter bounds than uniform random sampling for the same sample size, or (b) smaller sample sizes than uniform random sampling conforming to the same probabilistic bounds. We now look at usage of uniformly sampled streams in big data applications, and introduce the task of stratified sampling for Twitter.

[1] http://uk.businessinsider.com/twitter-tweets-per-day-appears-to-have-stalled-2015-6?r=US&IR=T

1.1 Using Uniformly Sampled Data

For a big data application, it is of interest to ensure that the sampled data used is representative of the global data, given the topic of interest. Probabilistic bounding of large deviations from global mean values [5] has been a popular way to ensure the same. The intuition is that ensuring reliable estimation of global occurrence rates would help in reliable estimation of the global results for end applications too. It is desirable to obtain a sampled set, such that the end result of any application (such as finding trending hashtags, sentiment analysis, topic clustering, summarization, etc) be close to the end result of the same application applied on the universe. However, given the complexity and variety of analytics tasks, such bounds are application-specific and need to be analytically developed separately for each application. As an example, an attempt to bound the results of a simple sentiment analyzer was done in [16]. In the interest of providing generic bounds that are likely to benefit any application, bounding occurrence rates of key entities such as words/hashtags [4, 16] has been explored as a natural first step in probabilistically guaranteed sampling.

Results from deviation theory [5] suggest that sample sizes to ensure probabilistic bounds need to be increased as the rate of presence of the monitored topic decreases in the global dataset. For example, a Twitter application monitoring national elections in India can afford to sample much fewer than another application focusing on a regional film festival, to achieve the same probabilistic bounds on deviation. This is because the former topic has a higher rate of presence (e.g., the hashtags occur more frequently) as compared to the latter topic that generates interest within a smaller audience. This has obvious cost implications; the application can switch from the paid Decahose API to the Free 1% stream while shifting focus from a niche topic to a much more popular one. Usage of such results requires that the rate of occurrence (of hashtags of interest to) the event monitored is known before-hand, to determine the sample size. For practical scenarios, the occurrence rates are available with data providers who already maintain indexes on data to support search functionalities. In short, an application using Twitter data for day-to-day monitoring of a topic would first characterize the topic of interest by keyword/hashtags and then query the data provider for a sample with a specification of the desired probability (e.g., $> 90\%$) and permissible deviation (e.g., $< 10\%$). The data provider would internally use the occurrence rate statistics of the hashtags, and estimate the required

uniform random sample size with a corresponding pricing. This may be done using Chernoff bounds formulae [5] that provide the minimum sample size required to ensure that the occurrence rate in the sample, for *each* hashtag, does not deviate by more than the specified deviation with the specified probability. A data provider offering such a probabilistic bounded sampling API is obviously attractive to the users since it allows them to be frugal on sample sizes especially while monitoring popular topics.

1.2 Why Stratified Sampling?

We now motivate using an example as to why stratified sampling would be of interest in this scenario. Consider an intelligence application looking to assess global opinion polarity on the *US Presidential Elections*. Given the geographic focus of the topic, it is conceivable that core hashtags for this topic are twice as frequent in tweets from US when compared to the rest of the world (RoW), even if the overall tweet volumes from the US and RoW are comparable. Geographic stratification is already performed by data providers for tasks such as geo-specific trends estimation, and is thus readily available with them. Uniform sampling would require us to sample as many tweets from RoW as from US. Due to the low occurrence rates of pertinent hashtags in RoW, marginal utility of a tweet from RoW in determining opinion on the event would be lower than that from US. However, since the task is to gauge global opinion, we cannot readily use results from a pure US sample; in particular, analogous to the uniform random sampling case, we would like to ensure that the sample would enable us to estimate global occurrence rates accurately. There exists an opportunity to exploit the knowledge of differential occurrence rates across US and RoW to work with smaller samples without compromising on the desired probabilistic guarantees.

However, to the best of our knowledge, there exists no technology to exploit differential occurrence rates across strata to derive smaller sample sizes that agree to probabilistic bounds (as given by Chernoff bounds [5] for uniform random sampling) within or outside the context of Twitter sampling. Our focus in this paper is to precisely develop that technology. Data providers would be able to leverage our method to provide a newer set of sampling APIs, stratified sampling APIs, that will output stratified samples agreeing to the same probabilistic bounds as in the uniform random sample case. The data consumer provides the same input to the data provider, a set of hashtags and the desired specification of probabilistic bounds; the data provider would then use our formulation and provide a smaller stratified sample to the user. The smaller sample sizes help the data user to monitor more topics for the same data cost.

1.3 Our Contributions

Our main contributions are as follows:

- For the first time, we consider the problem of bounding deviations in occurrence rate estimates of words/hashtags in stratified sampling in the context of Twitter, and outline methods to estimate sufficient sample sizes.
- We analyze the quantum of gains achieved using our method over corresponding estimates from uniform random sampling under the same setting, on simulations as well as real-world data.

It may be noted that even small reductions in sufficient sample size estimates are critical since procurement of tweets is practically the costliest aspect of maintaining a Twitter-based intelligence application. Data providers can leverage our technology to diversify their API portfolio to include stratified sampling. On the other hand, the improved sample sizes enable big-data analytics applications to broaden their footprint at the same data procurement and compute costs. Thus, our work is squarely targeted at players in the big data space. ■

Roadmap: We start off with some background on probabilistic guarantees and occurrence rate bounding in Section 2. We will outline related work in Section 3, define the problem in Section 4 and describe our method in Section 5. This is followed by extensive simulation and experimental analysis in Section 6 and conclusions in Section 7.

2 Background

2.1 Probabilistic Guarantees

The data user/application would like the data provider to provide guarantees on the sampled set in representing the universe. A probabilistic guarantee on occurrence rate ensures that the global occurrence rate as estimated from the sample does not deviate from the actual global occurrence rate more than a specified tolerance, with at least a specified probability. Thus, such a guarantee is fully specified by a combination of tolerance, and probability threshold. Consider an example application seeking to summarize Twitterati's opinion on the US Presidential Election. If the hashtag *#HilaryClinton* appears in 20% of the tweets in the whole Twitter stream, the application designer might like to ensure that the frequency of the hashtag as estimated from the sample be within $20\pm2\%$ (i.e., 10% tolerance), with a high probability (say, 90%). This is so since the hashtag *#HilaryClinton* is central to the problem that the application is trying to address. If the estimated frequency of *#HilaryClinton* from the sample turns out to be 30%, it could mean that our application's opinion summary is skewed in favor of users who mention *#HilaryClinton* (and vice versa). As in previous work, we will work with relative bounds expressed as percentages. For each application domain, one may intuitively expect that there would be some such core hashtags, noun phrases, or words of interest whose frequencies as estimated from the sample be close to the dataset frequency. A probabilistic guarantee on the occurrence rate for a set of hashtags/words specifies that the occurrence rate of each word in the set be estimated to within the specified tolerance subject to the probabilistic bound. For a particular sample, the occurrence rate condition is said to fail even if the occurrence rate of one word deviates further than the tolerance bound.

2.2 Occurrence Rate Bounding

Chernoff bounds [5] are tailored to address the probabilistically guaranteed sample size estimation problem in Random Sampling; i.e., the task of bounding the probability of tail events, specifically that of large divergence of occurrence rate estimates (from the sample) from their values in the universe. While being generally applicable to get bounds of large deviations from the mean, they provide sufficient sample sizes to probabilistically bound the deviation of the occurrence rate or frequency of a word/hashtag. Consider a Binomial random variable X that is the sum of iid Bernoulli random variables, then, for any $0 < \epsilon < 1$, the following hold:

$$\mathbb{P}\{X < (1-\epsilon)\mathbb{E}[X]\} \leq e^{-\frac{\epsilon^2\mathbb{E}[X]}{2}}$$

$$\mathbb{P}\{X < (1+\epsilon)\mathbb{E}[X]\} \leq e^{-\frac{\epsilon^2\mathbb{E}[X]}{3}}$$

where $\mathbb{P}\{Y\}$ denotes the probability of event Y, and $\mathbb{E}[X]$ the expectation of the random variable X; $\mathbb{E}[X] = s \times p$ where p is the success rate of the underlying Bernoulli random variable and s is the number of trials (i.e., the sample size in the sampling case). For the word occurrence rate scenario, the Bernoulli random variable (that X sums over) is one that has success probability equivalent to the occurrence rate of the word in the corpus. For example, if the word appears in 10% of the tweets in the corpus, the corresponding Bernoulli random variable would have $p = 0.1$. These bounds can be generalized to multiple words/hashtags and have been explored in AI for data-intensive applications such as mining; for example, previous work has addressed the task of preserving the status of objects as being θ-frequent (i.e., have a frequency more than θ) or not [4]. This has been adapted to the context of sampling in Twitter as well [16] with further extensions to derive bounds on preserving the dominant sentiment of a word. Thus, there has been recent interest in deriving sufficient sample size estimates towards preserving specific statistics in uniform random sampling within the data analytics community.

3 Related Work

There are a variety of applications for mining Twitter data, including ones for tweet summarization [11][18], topic analysis [3][20] and twitter sentiment analysis [13]. Since most such methods would need to analyze content and are thus computationally intensive, sampling would be an essential step for them to be applied to large scale twitter data.

There has been much empirical work on sampling such as analysis of sampled streams [15, 7, 12, 9, 1]. In [15] the authors compare Twitter sampling API's feed with the tweets obtained from Twitter Firehose API, which contains all the tweets. Authors empirically find that the analysis from the data obtained using Twitter's Streaming API (1% random sample) do not conform with Twitter's Firehose data (100% sample) for a set of end applications. In [7], the authors empirically compare sampling done with the help of human "experts" against random sampling. In [14], the authors analyze the bias in Twitter's API without using the costly Firehose data. In [12], the effects of using multiple streaming APIs is studied. The authors conclude that the Twitter's 1% Streaming API is rather biased than being random. All these studies empirically evaluate the quality of the Tweets for the existing Twitter APIs which mostly employ uniform random sampling. In contrast, we provide a theoretical treatment to determine the sample size needed to produce representative samples using stratified sampling. A recent work on Twitter sampling [16] looks at classifying words into frequent or infrequent using a threshold; it then builds upon ideas from work on frequent itemset mining [4] to bound the probability of words having a status in the sample different from their status in the whole. They also extend the bounds to derive necessary sample sizes for preserving the dominant sentiment of words in the sample. Our work, while related due to addressing sampling on Twitter, focuses on a different problem.

Stratified sampling has been studied extensively in the statistics literature [6, 8, 19]. The existing methods find the optimal sample size to minimize the variance of the estimates. However, they do not transcend into the probabilistic guarantees in bounding the estimates as done by Chernoff bounds [5]. We advance the state-of-art in stratified sampling, by deriving expressions to find the probability of bounding the estimates for the chosen sample size; these can in turn be used for arriving at sufficient sample sizes.

4 Problem Formulation

Consider a dataset $\mathcal{D}$ of tweets that is stratified/split into two strata $\mathcal{D}_1$ and $\mathcal{D}_2$; we will consider two-strata stratification for narrative simplicity and will later show that the problem definition as well as our method easily generalizes to any number of strata. Now, consider a set of words/tags/phrases[2] of interest, $w = \{w^1, w^2, \ldots\}$ for whom the occurrence rate is known in each stratum; $\hat{x}^i_j$ denotes the rate of occurrence of w^i in $\mathcal{D}_j$ whereas $\hat{x}^i$ denotes the occurrence rate of w^i in the whole dataset $\mathcal{D}$. Occurrence rates measure the fraction of tweets from the stratum of interest, and are thus in $[0, 1]$. We also have a chosen tolerance level ϵ indicating the amount of fractional deviation from expected occurrence rate, and a probability h that bounds the probability of larger deviations.

The task of interest is to identify a stratified sampling strategy $[S_1, S_2]$ where S_1 and S_2 tweets be uniformly randomly sampled separately from $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively, so that such a sample $\mathcal{S}$ ($|\mathcal{S}| = S_1 + S_2$) confirms to the following:

$$\mathbb{P}\{\cup_i \big(X^i_{\mathcal{S}} \leq (1-\epsilon)\hat{x}^i|\mathcal{D}| \ \cup \ X^i_{\mathcal{S}} \geq (1+\epsilon)\hat{x}^i|\mathcal{D}|\big)\} < h \quad (1)$$

where $X^i_{\mathcal{S}}$ is a random variable denoting the extrapolated frequency of w^i in $\mathcal{D}$ from a sample $\mathcal{S}$ generated using the stratified-sampling strategy $[S_1, S_2]$ and $\hat{x}^i \times |\mathcal{D}|$ denotes the actual frequency in the whole dataset. Preserving frequencies by a multiple of $\pm\epsilon$ is exactly the same as preserving occurrence rates by a multiple of $\pm\epsilon$, since occurrence rates is simply the frequency scaled down by the dataset size (on both sides of the inequality). Informally, we want to ensure that for any sample generated according to the strategy $[S_1, S_2]$, the probability of the estimated frequency of *any* word w^i (i.e., $X^i_{\mathcal{S}}$) deviating by more than $\pm\epsilon$ times the actual frequency be bounded by h. In particular, even one word not satisfying its condition would be a failure event. This can be generalized to k strata by changing the format of the strategy of interest from a pair to a k-length array $[S_1, \ldots, S_k]$.

5 Our Method

We will first outline our method for two-strata stratified sampling, and later show how that could be generalized to more number of strata. Consider a stratified sampling strategy $[S_1, S_2]$ and a word w^i. Let x^i_j be the random variable corresponding to finding the word w^i in stratum $\mathcal{D}_j$ ($j \in \{1, 2\}$). $X^i_{\mathcal{S}}$ is then a function of x^i_js as the following:

$$X^i_{\mathcal{S}} = \frac{|\mathcal{D}_1|}{S_1} \times \sum_{k=1}^{S_1} x^i_1 + \frac{|\mathcal{D}_2|}{S_2} \times \sum_{k=1}^{S_2} x^i_2 \quad (2)$$

$X^i_{\mathcal{S}}$ is the conventional stratified sampling variable denoting frequency of w^i in $\mathcal{D}$ under the stratified sampling strategy $[S_1, S_2]$. The expected value of $X^i_{\mathcal{S}}$, which we will denote as μ^i, is independent of $\mathcal{S}$, and may be written as:

$$\mathbb{E}[X^i] = \mu^i = |\mathcal{D}_1| \times \hat{x}^i_1 + |\mathcal{D}_2| \times \hat{x}^i_2 = |\mathcal{D}| \times \hat{x}^i \quad (3)$$

The last condition holds since the extrapolation in $X^i_{\mathcal{S}}$ is in proportion to the strata sizes. We use μ^i and the union bound on Eq. 1 to write as:

[2] referred to generically as words hereon.

$$\mathbb{P}\{\cup_i \big((X^i_{\mathcal{S}} \leq (1-\epsilon)\mu^i) \cup (X^i_{\mathcal{S}} \geq (1+\epsilon)\mu^i)\big)\} < \big(\sum_i \mathbb{P}\{X^i_{\mathcal{S}} \leq (1-\epsilon)\mu^i\}\big) + \big(\sum_i \mathbb{P}\{X^i_{\mathcal{S}} \geq (1+\epsilon)\mu^i\}\big)$$

We will consider bounding the RHS of the above equation, to be lower than h. Going by conventions, we will refer to the first term in the RHS as the left-tail, and the second term as the right-tail. We first illustrate simplifying the left-tail expression for a particular w^i.

5.1 Left-Tail

We will now use a positive quantity t and multiply each side of the internal expression by $-t$ and exponentiate, with a corresponding inversion of the inequality. It may be noted that this is inspired from the classical derivation for Chernoff bounds; however, unlike Chernoff bounds, our random variable is not a Binomial random variable.

$$\mathbb{P}\{X^i_{\mathcal{S}} \leq (1-\epsilon)\mu^i\} = \mathbb{P}\{\exp(-t(1-\epsilon)\mu^i) \geq \exp(-tX^i_{\mathcal{S}})\}$$

Using the Markov inequality, i.e., $\mathbb{P}\{A \geq a\} \leq \frac{\mathbb{E}[A]}{a}$, the above expression is upper bounded by:

$$\frac{\mathbb{E}[\exp(-tX^i_{\mathcal{S}})]}{\exp(-t(1-\epsilon)\mu^i)} \tag{4}$$

Let us now focus on the numerator, which we expand using the expression from Eq. 2 and re-write using $\exp(a+b) = \exp(a) \times \exp(b)$.

$$\mathbb{E}[\exp(-tX^i_{\mathcal{S}})] = \mathbb{E}[\exp\big(-t \times (\frac{|\mathcal{D}|}{S_1} \times \sum_{k=1}^{S_1} x^i_1)\big) \times \exp\big(-t \times (\frac{|\mathcal{D}|}{S_2} \times \sum_{k=1}^{S_2} x^i_2)\big)] \tag{5}$$

x^i_1 and x^i_2 within the summation in the equation above are random variables. We can take the $\mathbb{E}[.]$ and $\exp(.)$ inward, assuming independence among the inner random variables.

$$= \Big(\prod_{k=1}^{S_1} \mathbb{E}[\exp(-tx^i_1 \frac{|\mathcal{D}_1|}{S_1})]\Big)\Big(\prod_{k=1}^{S_2} \mathbb{E}[\exp(-tx^i_2 \frac{|\mathcal{D}_2|}{S_2})]\Big)] \tag{6}$$

Consider the internal expression $\mathbb{E}[\exp(-tx^i_j \frac{|\mathcal{D}_j|}{S_j})]$ (sub/superscripts generalized). The random variable x^i_j will be 1 with a probability of $\hat{x}^i_j$ and 0 with a probability $(1-\hat{x}^i_j)$. We can write the expectation as the sum of these two cases:

$$\mathbb{E}[\exp(-tx^i_j \frac{|\mathcal{D}_j|}{S_j})] = \hat{x}^i_j \times \exp(-t\frac{|\mathcal{D}_j|}{S_j}) + (1-\hat{x}^i_j) \times \exp(0)$$

$$= 1 - \hat{x}^i_j\Big(1 - \exp(-t\frac{|\mathcal{D}_j|}{S_j})\Big)$$

We now use the inequality $1 - x < \exp(-x)$ to further upper bound the above expression as:

$$\mathbb{E}[\exp(-tx^i_j \frac{|\mathcal{D}_j|}{S_j})] < \exp\Big(-\hat{x}^i_j(1 - \exp(-t\frac{|\mathcal{D}_j|}{S_j})\Big) \tag{7}$$

Re-writing and putting this back into Eq. 6,

$$\mathbb{E}[\exp(-tX^i_{\mathcal{S}})] < \Big(\prod_{k=1}^{S_1} \exp\big(\hat{x}^i_1(\exp(-t\frac{|\mathcal{D}_1|}{S_1}) - 1)\big)\Big) \times \Big(\prod_{k=1}^{S_2} \exp\big(\hat{x}^i_2(\exp(-t\frac{|\mathcal{D}_2|}{S_2}) - 1)\big)\Big) \tag{8}$$

Since $\exp(a) \times \exp(b) = \exp(a+b)$:

$$< \exp\Big(\sum_{k=1}^{S_1} \big(\hat{x}^i_1(\exp(-t\frac{|\mathcal{D}_1|}{S_1}) - 1)\big) + \sum_{k=1}^{S_2} \big(\hat{x}^i_2(\exp(-t\frac{|\mathcal{D}_2|}{S_2}) - 1)\big)\Big) \tag{9}$$

Since the expression does not have random variables:

$$< \exp\Big(\sum_{j\in\{1,2\}} S_j \hat{x}^i_j(\exp(-t\frac{|\mathcal{D}_j|}{S_j}) - 1)\Big)$$

Replacing this upper bound in Eq. 4 and re-writing μ^i,

$$\mathbb{P}\{X^i_{\mathcal{S}} \leq (1-\epsilon)\mu^i\} < \exp\Big(t(1-\epsilon)(|\mathcal{D}| \times \hat{x}^i) + \sum_{j\in\{1,2\}} S_j \hat{x}^i_j(\exp(-t\frac{|\mathcal{D}_j|}{S_j}) - 1)\Big) \tag{10}$$

Using a similar sequence of steps for right-tail:

$$\mathbb{P}\{X^i_{\mathcal{S}} \geq (1+\epsilon)\mu^i\} < \exp\Big(-t(1+\epsilon)(|\mathcal{D}| \times \hat{x}^i) + \sum_{j\in\{1,2\}} S_j \hat{x}^i_j(\exp(t\frac{|\mathcal{D}_j|}{S_j}) - 1)\Big) \tag{11}$$

We will refer to the expressions in the RHS of Eq. 10 and Eq. 11 as $LU(t,i,S_1,S_2,\epsilon)$ and $RU(t,i,S_1,S_2,\epsilon)$ respectively[3]. These upper bounds hold for any positive value of t; the preferred value of t would be that which gives the tightest bound. Further, the expressions above can be easily generalized to any stratification of the dataset into k strata by letting the j variable iterate over as many values as there are strata.

5.2 Full Expression and Optimization

The full expression for upper bound would thus be:

$$\mathbb{P}\{\bigcup_i \big((X^i_{\mathcal{S}} \leq (1-\epsilon)\mu^i) \cup (X^i_{\mathcal{S}} \geq (1+\epsilon)\mu^i)\big)\} < \sum_i \Big(LU(t^i_L,i,S_1,S_2,\epsilon) + RU(t^i_R,i,S_1,S_2,\epsilon)\Big) \tag{12}$$

If the RHS of the above expression evaluates to less than h, then the LHS would too (since LHS<RHS as above), and our task in Eq. 1 will be satisfied. We have added subscripts and superscripts to t within the expressions to indicate that the ts need not necessarily

[3] Short for Left-tail Upper bound and Right-tail Upper bound

take the same value across expressions and are internal to the expression; the t used in the $LU(.)$ for one word w_i could be different from that used in the $LU(.)$ or $RU(.)$ for the same or different words. It may be noted that the flexibility that we have is to alter S_1, S_2 and the t's (ϵ is part of the problem specification), since the data stratification is given and $\hat{x}^i_j$ is deterministic in the sense that it is calculated from the stratified dataset. To re-iterate, if we can find values of S_1, S_2 and the ts such that the following holds

$$\sum_i \Big(LU(t^i_L, i, S_1, S_2, \epsilon) + RU(t^i_R, i, S_1, S_2, \epsilon) \Big) < h \quad (13)$$

we can then claim to have a sampling strategy $[S_1, S_2]$ that addresses our task. However, simply addressing the task is not sufficient; for example, a sample size for uniform random sampling that addresses our task is easily available from Chernoff bounds. Our interest is in achieving the task using fewer samples than uniform random sampling by leveraging strata level occurrence rates (i.e., $\hat{x}^i_j$s), and thus the measure of interest is the total sample size, $S_1 + S_2$, which we will look to minimize. Thus, ideally, we look for values of S_1, S_2 and the ts such that the above condition is satisfied and $S_1 + S_2$ is minimized.

Due to the complexity of the expression, a search in the possible values of S_1, S_2 and the ts is a possibility to identify feasible sampling strategies. From an optimization perspective, it is useful to localize the search to a small region of the parameter space, in the interest of reducing computational expense. Since S_1 and S_2 are sizes of samples from $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively, their ranges would respectively be $[1, |\mathcal{D}_1|]$ and $[1, |\mathcal{D}_2|]$. Though the extent of the search space for values of S_1 and S_2 are finite (due to bounds), the ts can take any positive value; we will now see how to localize the optimal t to limit the search.

5.3 Localizing the Optimal t

Consider the RHS in Eq. 10 which we are interested in minimizing (Ref. Eq. 13); we will focus on optimizing for t for chosen values of S_1 and S_2. Since $\exp(x)$ increases with x, we can focus on minimizing the expression within the exp(.):

$$fLU_i(t) = t(1-\epsilon)(|\mathcal{D}| \times \hat{x}^i) + \sum_{j \in \{1,2\}} S_j \hat{x}^i_j (\exp(-t\frac{|\mathcal{D}_j|}{S_j}) - 1) \quad (14)$$

We outline some analytical observations about the behavior of $fLU_i(t)$ with varying t; we omit detailed derivations for brevity. First, $fLU_i(t = 0) = 0$. This is evident from setting $t = 0$ in Equation 14. Secondly, there exists a positive value t' such that the following hold:

$$\frac{\partial fLU_i(0 < t < t')}{\partial t} < 0$$

$$\frac{\partial fLU_i(t = t')}{\partial t} = 0$$

$$\frac{\partial fLU_i(t > t')}{\partial t} > 0$$

In other words, $fLU_i(t)$ is a convex function in t in our region of interest (i.e., positive t or $t \in (0, \infty]$) with an optima at t' where $fLU_i(t')$ would evaluate to a negative value. Thus, if we can find values t_l and t_u such that $\frac{\partial fLU_i(t=t_l)}{\partial t} < 0$ and $\frac{\partial fLU_i(t=t_u)}{\partial t} > 0$, we can localize the search for the optimal t to the range (t_l, t_u) since the optimal t is bound to be in that range, given the above observations. We will show that $\left[\frac{\log(\frac{1}{1-\epsilon})}{\max\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}}, \frac{\log(\frac{1}{1-\epsilon})}{\min\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}}\right]$ is one such range.

Consider the slope of $fLU_i(t)$:

$$\frac{\partial fLU_i(t)}{\partial t} = (1-\epsilon)(|\mathcal{D}| \times \hat{x}^i) + \sum_{j \in \{1,2\}} |\mathcal{D}_j| \hat{x}^i_j \exp(-t\frac{|\mathcal{D}_j|}{S_j})$$

Setting $t = \log(\frac{1}{1-\epsilon}) / \max\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}$ in the above expression and using Eq. 3 with some re-arrangements yields:

$$\sum_{j \in \{1,2\}} |\mathcal{D}_j| \hat{x}^i_j \Big((1-\epsilon) - (1-\epsilon)^{\frac{\frac{|\mathcal{D}_j|}{S_j}}{\max\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}}} \Big)$$

The exponent of the second $(1-\epsilon)$ is evidently less than 1.0 since its denominator is least as big as its numerator (if not bigger). Also, given that $(1-\epsilon) < 1.0$ and due to the obvious result that $x^y > x$ when $x < 1.0$ and $y < 1.0$, the multiplier of each $|\mathcal{D}_j| \hat{x}^i_j$ term would be negative, leading to a negative value for the whole expression. Analogously, we now consider the slope at $t = \log(\frac{1}{1-\epsilon}) / \min\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}$:

$$\sum_{j \in \{1,2\}} |\mathcal{D}_j| \hat{x}^i_j \Big((1-\epsilon) - (1-\epsilon)^{\frac{\frac{|\mathcal{D}_j|}{S_j}}{\min\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}}} \Big)$$

In this case, the exponent of the second $(1-\epsilon)$ turns out to be greater than one. Thus, adapting the earlier argument, the multiplier of each $|\mathcal{D}_j| \hat{x}^i_j$ would be positive, leading to an overall positive value. Thus:

$$\arg\min_t fLU_i(t) \in \left[\frac{\log(\frac{1}{1-\epsilon})}{\max\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}}, \frac{\log(\frac{1}{1-\epsilon})}{\min\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}}\right]$$

The analogous result for the right-tail expression is:

$$\arg\min_t fRU_i(t) \in \left[\frac{\log(1+\epsilon)}{\max\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}}, \frac{\log(1+\epsilon)}{\min\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}}\right]$$

Though the optimal value of t would be different for expressions corresponding to different words, the bounds above are attractive in that they do not depend on any $\hat{x}^i_j$s and thus can be used across words. These bounds can be easily extended from two strata to multiple strata by changing the max and min to iterate over k entries instead of two.

It is computationally intensive to find a separate optimal value of t for each term in Eq. 13. Thus, one might fall back to search for a single value of t to be used across all expressions in Eq. 13; this single value could be chosen as that which minimizes the value of the whole expression in Eq. 13. For such a case, the search may be directed to within the union of the left-tail and right-tail bounds above, which would be:

$$\left[\frac{\log(1+\epsilon)}{\max\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}}, \frac{\log(\frac{1}{1-\epsilon})}{\min\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}}\right] \quad (15)$$

Optimal value of t can be either obtained by searching through the values in the range given by Eq. 15, or by using a gradient descent approach where the update equation would be:

$$t^{new} = t^{old} - \eta \frac{\partial fLU_i(t)}{\partial t} \tag{16}$$

where η is the learning rate. t can be initialized to any value in the range given by Eq. 15. In our experiment, we have used grid-search approach to search for the optimal value of t in view of its simplicity for the optimization of a single dimensional variable.

5.4 Grid-Search

Algorithm 1 outlines an intuitive grid-search approach, *StratSam*, to discover a sampling strategy $[S_1, S_2]$; we resort to choosing a single value of t across terms in Eq. 13 for computational convenience as outlined earlier. The algorithm is largely self-explanatory with lines 6 and 7 checking for satisfaction of the task condition (Eq. 13). Line 4 avoids checking for strategies that are already worse on total sample size than the best strategy seen so far. It may be noted that *StratSam* allows for exploring the trade-off between computational expense and accuracy by tuning the step-size hyperparameters. Smaller step-sizes would allow to discover a better sampling strategy (i.e., smaller $(S_1 + S_2)$) whereas larger step-sizes lead to fast search completion. It is also worthy to point out that the condition may never be reached when the chosen h and ϵ values are very small for the dataset size; in such cases, we will choose the entire dataset as the sample. In large datasets such as those with Twitter, such cases are very rare.

Alg. 1 ***Grid-Search: StratSam***

Input. 2-Strata Dataset Specs: $|\mathcal{D}_1|$, $|\mathcal{D}_2|$, $\forall\, w_i, (\hat{x}_1^i, \hat{x}_2^i)$ pairs
Problem Specs: ϵ, h
Hyper-Parameters. Step-sizes $s_1, s_2, \delta t$
Output. Sampling Strategy, i.e., a vector $[S_1, S_2]$.

1. Best Strategy, BS = ϕ, Best Strategy Size, BSS = ∞
2. For $S_1 = 1 \rightarrow |\mathcal{D}_1|$ in steps of s_1
3. For $S_2 = 1 \rightarrow |\mathcal{D}_2|$, s_2
4. If $(S_1 + S_2) > BSS$ continue;
5. For $t = \frac{\log\left(1+\epsilon\right)}{\max\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}} \rightarrow \frac{\log\left(\frac{1}{1-\epsilon}\right)}{\min\{\frac{|\mathcal{D}_1|}{S_1}, \frac{|\mathcal{D}_2|}{S_2}\}}, \delta t$
6. Evaluate $v = \sum_i LU(i) + RU(i)$ with the choices of S_1, S_2 and t
7. If $(v < h) \wedge (S_1 + S_2 < BSS)$
8. $BS = [S_1, S_2], BSS = (S_1 + S_2)$
9. Output BS as sampling strategy of choice.

5.5 Remarks

Better Sample Sizes: The total sample size from the above stratified approach would *always be equal or smaller* than that from a similar uniform random sampling approach (or that from the looser Chernoff bounds). This is so since the latter's sample size would also be a valid solution for the former, when split in proportion to strata sizes.

Speeding up the Search: Our proposed grid-search approach is quite feasible for a small number of strata and is very fast. It can be further speed-ed up by replacing the grid-search for t (Lines 5-8 in the algorithm) by a gradient descent approach, given the convexity observation from Section 5.3. In resource constrained scenarios or to estimate sample sizes for fine-grained data stratification, conventional optimization methods may be employed over the entire search space of S_is and t. For purposes of optimization, the objective function is simply $(\sum_j S_j)$ with the generalization of Eq. 13 to the required number of strata serving as an inequality constraint.

5.6 Uptake of Our Work

We now discuss considerations relating to uptake of our work. Analogous to the assumption of global occurrence rate availability for the uniform random sampling setting, we assume the availability of stratum-level occurrence rates. We will now outline why stratum-level occurrence rate availability is a feasible assumption in practical scenarios. Our target ecosystem is the emerging data economy that encompasses data providers and data consumers. Data providers maintain the entire dataset and provide various kinds of APIs for usage by data consumers with a pricing scheme. These APIs would include sampled streams, as well as search functionalities and various analytics features such as geo-trends, all of which require content indexing at the level of different granularities such as strata. The source of the data (e.g., the region), the type of the tweets (e.g., Twitter activity streams[4]) etc. provide straightforward stratifications that would be maintained at the data provider. Typical search functionalities are supported by inverted lists at the level of each word/tag; occurrence rates are then simply normalized inverted list sizes. Our method leverages the skew in occurrence rates of topical hashtags across strata to reduce required sample sizes as against those for uniform random sampling. Our results are generalizable to cover domains such as market-basket data mining where frequencies of specific items within transactions (as opposed to frequencies of words in tweets) are the measure of interest; in such cases, we can leverage existing stratification of customers such as *silver, gold* and *platinum* and/or stratification of stores such as *small* and *large* to collect stratum-level information. Uptake of our technology necessitates a few simple changes at the data user as well as the data provider.

Data User/Application: The sampled data request issued by the data user remains the same, i.e., a set of words and the specification of desired probabilistic bound. However, the data sample received from the provider would now be a stratified sample. Analogous to usage of uniform random samples where the results (e.g., sentiment frequencies) derived from the sample needs to be extrapolated to get to corpus-level estimates, results from the stratified samples need to be extrapolated in accordance with the sampling rates (as in Equation 2), to arrive at corpus-level estimates. This is the only difference required at the data user's side.

Data Provider: As outlined earlier, the data provider maintains multiple stratifications of Twitter data; while some of these may be maintained for purposes such as providing trends estimation and faceted search, some stratifications could be specifically targeted at supporting the new stratified sampling API. The data provider kicks off processing upon receiving a sampled data request from the user comprising of a set of words/tags, tolerance, and probability threshold. Next, the data provider runs our method against each stratification separately using respective occurrence rate statistics, each of which provide a different sample size estimate. The smallest sample size estimate is expected to be achieved for the stratification where the skew of occurrence rates for the provided set of words/tags is maximum. The data provider would then return the best sample, and charge the data user accordingly. In a competitive marketplace, it is in the inter-

[4] http://support.gnip.com/articles/activity-streams-intro.html

est of the data provider to maintain a rich library of different stratifications. This would ensure that low sample sizes may be provided for data requests on a variety of topics, enhancing chances of repeat business.

6 Simulation and Experiments

We first describe the setup for our simulation and experimental studies followed by results and discussion.

6.1 Experimental Setup

We compare our method, *StratSam*, against uniform random sampling (US), the baseline method. Instead of using the final Chernoff bounds result that involves many approximations leading to looser (i.e., larger) sample size estimates, we do a similar derivation as in our case and use a grid search for fairness in comparison. For clarity, the US expression corresponding to RHS in Eq. 10 is:

$$\exp\left(t(1-\epsilon)(|\mathcal{D}| \times \hat{x}^i) + S\hat{x}^i(\exp(-t\frac{|\mathcal{D}|}{S}) - 1)\right) \quad (17)$$

The comparison of interest would be that between the US sample size ($US.Size$) and the total sample size $S_1 + S_2$; we use the sample size ratio, $SSR = \frac{S_1+S_2}{US.Size}$, as the primary evaluation measure; $SSR \leq 1$ always holds (Sec. 5.5), and lower values of SSR are desirable. We perform extensive simulation analysis as well as experiments on real-world data to illustrate the savings achieved by *StratSam* over US. In the case of analysis on real-world datasets, we analyze another measure, the actual empirical failure rate (*StratSam* and US guarantee that to be bounded by h) as well. For the real-world dataset, we use a set of tweets crawled around the time of the Indian General Election, 2014[5]. In our $StratSam$ implementation, we use 100 equal steps in each of the three parameters.

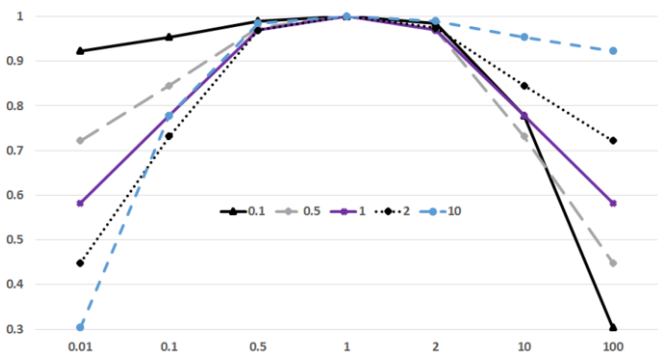

Figure 1. SSR ($\frac{S_1+S_2}{US.Size}$) on Y-axis vs. Occurrence Rate Ratio ($\frac{\hat{x}_1^1}{\hat{x}_2^1}$) plots for varying stratum size ratios ($\frac{|\mathcal{D}_1|}{|\mathcal{D}_2|}$)

6.2 Simulation Studies

We now use simulation studies to analyze the behavior of our approach. Two cases are considered: first, where there is only one core word for the topic of interest, and a second case involving two words.

[5] https://en.wikipedia.org/wiki/Indian_general_election,_2014

6.2.1 Single Word Simulation

Figure 1 plots the SSR trends when the occurrence rate ratio of a word across the two strata ($\hat{x}_1^1/\hat{x}_2^1$) is varied keeping the dataset-level occurrence rate (i.e., $\hat{x}^1$) constant at 0.2. We use $\epsilon = 0.1$ and $h = 0.1$ for the plot in the figure; the trends were similar for other choices of ϵ and h too. Such trendlines are plotted for varying values of relative strata sizes ($\frac{|\mathcal{D}_1|}{|\mathcal{D}_2|}$) from 0.1 to 10. When each trendline is analyzed, it may be seen that *StratSam* is able to achieve smaller sample sizes as $\hat{x}_1^1/\hat{x}_2^1$ deviates from 1 on either side. When occurrence rates are equal, the strata are practically indistinguishable wrt w^1 and *StratSam* defaults to the US sample size, as is expected. It may be noted that *StratSam* is able to leverage the skew in occurrence rates under equal strata sizes to achieve $> 40\%$ reductions in sample sizes over US. On analyzing across trendlines (i.e., across stratum size ratios), it is evident that *StratSam* performs best when the occurrence rate is very high in a very small stratum; for example, the bottom-right point in the chart corresponds to the word being 100 times more frequent in the first stratum when it is $1/10^{th}$ of the second stratum in size. Thus, the chosen keywords being denser in the smaller stratum is favorable to *StratSam*.

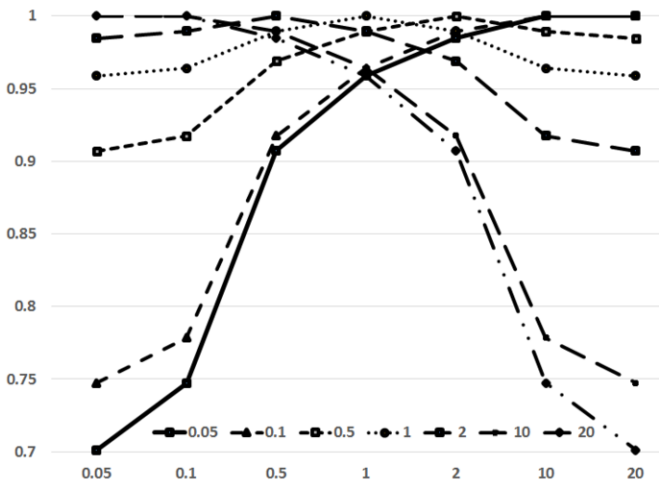

Figure 2. SSR ($\frac{S_1+S_2}{US.Size}$) on Y-axis vs. w^1 Occurrence Rate Ratio ($\frac{\hat{x}_1^1}{\hat{x}_2^1}$) plots for varying w^2 ratios ($\frac{\hat{x}_1^2}{\hat{x}_2^2}$)

6.2.2 Two Words Simulation

Figure 2 analyzes SSR trends for two words with equal sized strata (i.e., $|\mathcal{D}_1| = |\mathcal{D}_2|$). For the trendline where the second word is equally dense on either strata, deviations of $\hat{x}_1^1/\hat{x}_2^1$ shows similar trends as for the single-word case, though the quantum of savings is lower. Across trendlines, it may be seen that both the words being more skewed towards the same stratum (i.e., both occurrence rate ratios being low, or both being high) leads to maximum savings, with up to 30% savings recorded when occurrence rate ratios are both 0.05 (or equivalently, 20). Since typical sampling scenarios would be task focused (e.g., guaging sentiment on *US Elections*), it is intuitive to expect that words of interest are skewed towards the same stratum. The SSR trends were consistent with varying values of ϵ and h.

6.3 Experiments on Real-World Data

We use the tweet set from the Indian General Elections, 2014, and consider how *StratSam* performs on SSR under sets of words related

Words or Phrases	Universe size	Strata size	Strata Size Ratio	Word Skew (#N,#S)	Sample size (Strat-Sam)	Sample size (US)	SSR	Empirical Failure rate Strat-Sam	US
arvind, kejriwal, contribution	54501	north: 46808 south: 7693	6.08	(2,1)	14076 north:8993 south:5083	18638	**0.75**	0.01	0.008 (0.031)
bjp, modi, latestnew	87550	north: 31721 south: 55829	0.57	(2,1)	11997 north:1584 south:10413	16072	**0.74**	0.011	0.015 (0.028)
latestnew, sonia, varanasi, win, aap kejriwal, firstpost, narendra, namo, exit, gandhi, arvind, bjp, vote, modi, poll,	265060	north: 98726 south: 166334	0.59	(11,6)	56199 north:15149 south:41050	58643	**0.95**	0.011	0.012 (0.018)

Table 1. Results on Real Data (North-South Stratification)

Words or Phrases	Universe size	Strata size	Strata Size Ratio	Word Skew (#E,#W)	Sample size (Strat-Sam)	Sample size (US)	SSR	Empirical Failure rate Strat-Sam	US
arvind, kejriwal, contribution	54501	east: 48959 west: 5542,	9.09	(2,1)	15641 west:3388 east:12252	18638	**0.83**	0.015	0.018 (0.028)
bjp, modi, latestnew	87550	east: 44589 west: 42961	1.03	(2,1)	10752 east:2603 west:8149	16072	**0.66**	0.01	0.014 (0.058)
latestnew, sonia, varanasi, win, aap kejriwal, firstpost, narendra, namo, exit, gandhi, arvind, bjp, vote, modi, poll,	265060	east: 140378 west: 124682	1.12	(10,7)	44898 east:17379 west:27518	50691	**0.88**	0.008	0.011 (0.017)

Table 2. Results on Real Data (East-West Stratification)

to the election. We use the geo-stratification of tweets as *North* and *South*; *East* and *West* India using the location and time zone in the user profile. Twitter's API was used to crawl tweets from May 12 to May 19, 2014, using topical keywords related to the election event. Table 1 and 2 summarize some representative results. Instead of using the entire set of tweets as the dataset, we wanted to experiment with varying dataset sizes too. Towards this, for every set of keywords, we filter out all tweets not containing even one of those keywords, to create the dataset for that keyword set. Thus, universe size represents the number of tweets obtained after such filtering. Strata size shows the number of tweets in the respective strata, with the strata size ratio indicating the ratio of the sizes of the strata. To provide a sense of the word skew, we look at each word in the set of interest, and assign it to the stratum in which it has a higher occurrence rate; thus, a word skew of $(2, 1)$ indicates that 2 words have higher occurrence rates in the first stratum and the third word in the set occurs at a higher rate in the second stratum. While this does not capture the quantum of stratum-skew for each word, it is an indicator of the occurrence rate skew in the set of words of interest. We also report the sample size for *StratSam* and US, the SSR, and the empirical error rates obtained by repeatedly sampling (with 1000 Monte Carlo rounds) according to the respective strategy and measuring the fractional failure rate (which is analytically bounded above by $h = 0.1$). Results are obtained for $\epsilon = 0.1$ and $h = 0.1$. Empirical error rate within brackets in US column is obtained by uniform sampling with *StratSam* sample size. The trends are similar to that from the simulation and the experiments record gains up to 34% with significantly lower empirical error rates as well. The dataset was collected for the general elections, a pan-India event, thus mitigating the skew between various geographic strata within India; while this setting helps us observe that StratSam can achieve significant gains even in not-so-favorable scenarios, StratSam is expected to achieve much better gains when the stratification is more 'aligned' to the keyword set. It may be noted that in practical scenarios where millions of tweets need to be sampled on a paid-basis, even $\approx$5% gains are expected to result in large cost savings. Another noteworthy point is that most empirical failure rates are $\approx$ 10 times smaller than $h(= 0.1)$; this indicates potential for more empirical and/or theoretical work.

7 Conclusions and Future Work

In this paper, we considered the problem of using stratification in estimating sufficient sample sizes to reliably estimate the occurrence rates of specific words of interest, in sampling for Twitter. We exploit differential word occurrence rates across strata in a grid-search approach to significantly improve upon analogous estimates for uniform random sampling. We analyze our estimates through simulation studies as well as experiments on real-world data and illustrate that significant savings can be achieved over corresponding sample size estimates for uniform random sampling. We also outlined the context of big data applications that warrant superior sampling strategies for cost and computation reasons, and described how our method could be easily used by data providers and data users.

Translating the probabilistic bounds used in our approach to task-level bounds (e.g., bounds on deviation in sentiment analysis) would be an interesting direction for future research. Another direction is to see whether the sufficient sample sizes may be tightened in the context of our results in Section 6.3. Adapting the probabilistic bounds to time varying word occurrence rates and its application to online sampling streams and dynamic stratification derived from text clustering [2] could be considered in future. There are interesting engineering issues that are pertinent to the uptake of our method. For example, a data provider maintaining a library of different stratifications of data would benefit from heuristically choosing a subset of stratifications to run StratSam against; a heuristic that can choose geo-stratification when the set of keywords are to do with highly geo-focused events such as the UK EU Referendum would enable the data provider to achieve computational cost savings.

REFERENCES

[1] Jisun An and Ingmar Weber, 'Whom should we sense in social sensing - analyzing which users work best for social media now-casting', *EPJ Data Science*, (2015).

[2] Vipin Balachandran, P Deepak, and Deepak Khemani, 'Interpretable and reconfigurable clustering of document datasets by deriving word-based rules', *Knowledge and information systems*, **32**(3), 475–503, (2012).

[3] David Blei, Andrew Ng, and Michael Jordan, 'Latent dirichlet allocation', in *Journal of Machine Learning Research*, (2003).

[4] Venkatesan T. Chakaravarthy, Vinayaka Pandit, and Yogish Sabharwal, 'Analysis of sampling techniques for association rule mining', in *Database Theory - ICDT 2009, 12th International Conference, St. Petersburg, Russia, March 23-25, 2009, Proceedings*, pp. 276–283, (2009).

[5] Herman Chernoff, 'A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations', *The Annals of Mathematical Statistics*, 493–507, (1952).

[6] William G. COCHRAN, *Sampling Techniques*, Wiley, 1977.

[7] Saptarshi Ghosh, Muhammad Bilal Zafar, Parantapa Bhattacharya, Naveen Sharma, Niloy Ganguly, and Krishna Gummadi, 'On sampling the wisdom of crowds: Random vs. expert sampling of the twitter stream', in *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pp. 1739–1744, New York, NY, USA, (2013). ACM.

[8] R.O. Gilbert, *Statistical Methods For Environmental Pollution Monitoring,*, Van Nostrand, New York., 1987.

[9] Sandra Gonzalez-Bailon, Ning Wang, Alejandro Riveroc, Javier Borge-Holthoeferd, and Yamir Moreno, 'Assessing the bias in samples of large online networks', *Social Networks*, **38**, 16–27, (2014).

[10] Carol Huang, 'Facebook and twitter key to arab spring uprisings: report', in *The National*, volume 6, (2011).

[11] David Inouye and Jugal K Kalita, 'Comparing twitter summarization algorithms for multiple post summaries', in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pp. 298–306. IEEE, (2011).

[12] Kenneth Joseph, Peter M Landwehr, and Kathleen M Carley, 'Two 1% s dont make a whole: Comparing simultaneous samples from twitters streaming api', in *Social Computing, Behavioral-Cultural Modeling and Prediction*, 75–83, Springer, (2014).

[13] Olga Kolchyna, Tharsis TP Souza, Philip Treleaven, and Tomaso Aste, 'Twitter sentiment analysis', *arXiv preprint arXiv:1507.00955*, (2015).

[14] Fred Morstatter, Jürgen Pfeffer, and Huan Liu, 'When is it biased?: Assessing the representativeness of twitter's streaming api', in *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, pp. 555–556, (2014).

[15] Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen M. Carley, 'Is the sample good enough? comparing data from twitter's streaming API with twitter's firehose', in *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8-11, 2013.*, (2013).

[16] Deepan Subrahmanian Palguna, Vikas Joshi, Venkatesan T. Chakaravarthy, Ravi Kothari, and L. Venkata Subramaniam, 'Analysis of sampling algorithms for twitter', in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pp. 967–973, (2015).

[17] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo, 'Earthquake shakes twitter users: real-time event detection by social sensors', in *Proceedings of the 19th international conference on World wide web*, pp. 851–860. ACM, (2010).

[18] Beaux Sharifi, Mark-Anthony Hutton, and Jugal K. Kalita, 'Experiments in microblog summarization', in *Proceedings of the 2010 IEEE Second International Conference on Social Computing*, SOCIALCOM '10, pp. 49–56, Washington, DC, USA, (2010). IEEE Computer Society.

[19] S. K. Thompson, *Sampling*, Wiley, 2012.

[20] Yu Wang, Eugene Agichtein, and Michele Benzi, 'Tm-lda: Efficient online modeling of latent topic transitions in social media', in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pp. 123–131, New York, NY, USA, (2012). ACM.