k-variates++: more pluses in the k-means++

Richard Nock[†] RICHARD.NOCK@DATA61.CSIRO.AU
Raphaël Canyasse[¶] RAPHAEL.CANYASSE@POLYTECHNIQUE.EDU
Roksana Boreli[‡] R.BORELI@UNSW.EDU.AU
Frank Nielsen[§] NIELSEN@LIX.POLYTECHNIQUE.FR
Data61, Sydney, Australia & {[†]The Australian National University; [‡]The University of New South Wales}
Ecole Polytechnique, Palaiseau, France & {[¶]The Technion, Haifa, Israel; [§]Sony CS Labs, Inc., Tokyo, Japan}

Abstract

k-means++ seeding has become a de facto standard for hard clustering algorithms. In this paper, our first contribution is a two-way generalisation of this seeding, k-variates++, that includes the sampling of general densities rather than just a discrete set of Dirac densities anchored at the point locations, *and* a generalisation of the well known Arthur-Vassilvitskii (AV) approximation guarantee, in the form of a *bias+variance* approximation bound of the *global* optimum. This approximation exhibits a reduced dependency on the "noise" component with respect to the optimal potential — actually approaching the statistical lower bound. We show that k-variates++ *reduces* to efficient (biased seeding) clustering algorithms tailored to specific frameworks; these include distributed, streaming and on-line clustering, with *direct* approximation results for these algorithms. Finally, we present a novel application of k-variates++ to differential privacy. For either the specific frameworks considered here, or for the differential privacy setting, there is little to no prior results on the direct application of k-means++ and its approximation bounds — state of the art contenders appear to be significantly more complex and / or display less favorable (approximation) properties. We stress that our algorithms can still be run in cases where there is *no* closed form solution for the population minimizer. We demonstrate the applicability of our analysis via experimental evaluation on several domains and settings, displaying competitive performances vs state of the art.

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

1. Introduction

Arthur-Vassilvitskii's (AV) k-means++ algorithm has been extensively used to address the hard membership clustering problem, due to its simplicity, experimental performance and guaranteed approximation of the *global* optimum; the goal being the k-partitioning of a dataset so as to minimize the sum of within-cluster squared distances to the cluster center (Arthur & Vassilvitskii, 2007), *i.e.*, a centroid or a *population minimizer* (Nock et al., 2016b).

The k-means++ non-uniform seeding approach has also been utilized in more complex settings, including tensor clustering, distributed, data stream, on-line and parallel clustering, clustering with non-metric distortions and even clustering with distortions not allowing population minimizers in closed form (Ailon et al., 2009; Balcan et al., 2013; Jegelka et al., 2009; Liberty et al., 2014; Nielsen & Nock, 2014; Nielsen et al., 2014; Nielsen & Nock, 2015; Nock et al., 2008). However, apart from the non-uniform seeding, all these algorithms are distinct and (seemingly) do not share many common properties.

Finally, the application of k-means++ in some scenarios is still an open research topic, due to the related constraints – e.g., there is limited prior work in a differentially private setting (Nissim et al., 2007; Wang et al., 2015).

Our contribution — In a nutshell, we describe a generalisation of the k-means++ seeding process, k-variates++, which still delivers an efficient approximation of the global optimum, and can be used to obtain *and* analyze efficient algorithms for a wide range of settings, including: distributed, streamed, on-line clustering, (differentially) private clustering, etc. . We proceed in two steps.

First, we describe k-variates++ and analyze its approximation properties. We leverage two major components of k-means++: (i) data-dependent *probes* (specialized to observed data in the k-means++) are used to compute the weights for selecting centers, and (ii) selection of centers

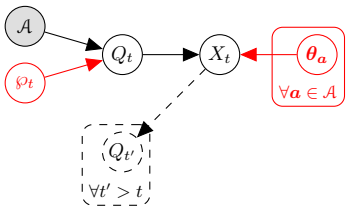

Figure 1. Graphical model for the k-means++ seeding process (black) and our generalisation (black + red, best viewed in color).

is based on an *arbitrary* family of densities (specialized to Diracs in the k-means++). Informally, the approximation properties (when only (ii) is considered), can be shown as:

$$\text{expected\_cost}(k\text{-variates++}) \quad \leq \quad (2 + \log k) \cdot \Phi \ , \ \text{with}$$

$\Phi \doteq 6 \cdot$optimal_**noise-free**_cost$+2 \cdot$noise_(bias + variance), where "noise" refers to the family of densities (note that constants are explicit in the bound). The dependence on these densities is arguably smaller than expected (factor 2 for noise vs 6 for global optimum). There is also not much room for improvement: we show that the guarantee approaches the Fréchet-Cramér-Rao-Darmois lower bound.

Second, we use this general algorithm in two ways. We use it directly in a differential privacy setting, addressing a conjecture of (Nissim et al., 2007) with weaker assumptions. We also demonstrate the use of this algorithm for a *reduction* to other biased seeding algorithms for distributed, streamed or on-line clustering, and obtain the approximation bounds for these algorithms. This simple reduction technique allows us to analyze lightweight algorithms that compare favorably to the state of the art in the related domains (Ailon et al., 2009; Balcan et al., 2013; Liberty et al., 2014), from the approximation, assumptions and / or complexity aspects. Experiments against state of the art for the distributed and differentially private settings display that solid performance improvement can be obtained.

The rest of this paper is organised as follows: Section 2 presents k-variates++. Section 3 presents approximation properties for distributed, streamed and on-line clustering that use a reduction from k-variates++. Section 4 presents direct applications of k-variates++ to differential privacy. Section 5 presents experimental results. Last Section discusses extensions (to more distortion measures) and concludes. In order not to laden the paper's body, a Supplementary Information (SI) provides all proofs and extensive experiments not shown here (Nock et al., 2016a).

2. k-variates++

We consider the hard clustering problem (Banerjee et al., 2005; Nock et al., 2016b): given set $\mathcal{A} \subset \mathbb{R}^d$ and integer $k > 0$, find centers $\mathcal{C} \subset \mathbb{R}^d$ which minimizes the L_2^2 potential to the centers (here, $\boldsymbol{c}(\boldsymbol{a}) \doteq \arg\min_{\boldsymbol{c} \in \mathcal{C}} \|\boldsymbol{a} - \boldsymbol{c}\|_2^2$):

$$\phi(\mathcal{A}; \mathcal{C}) \quad \doteq \quad \sum_{\boldsymbol{a} \in \mathcal{A}} \|\boldsymbol{a} - \boldsymbol{c}(\boldsymbol{a})\|_2^2 \ , \qquad (2)$$

Algorithm 0 describes k-variates++. u_m denotes the uniform distribution over $\mathcal{A}$ ($|\mathcal{A}| = m$). The parenthood with k-means++ seeding, which we name "k-means++" for short[1] (Arthur & Vassilvitskii, 2007) can be best understood using Figure 1 (the red parts in Figure 1 are pinpointed in Algorithm 0). k-means++ is a random process that generates cluster centers from observed data $\mathcal{A}$. It can be modelled using a two-stage generative process for a mixture of Dirac distributions: the first stage involves random variable $Q_t \sim \text{Mult}(m, \boldsymbol{\pi}_t)$ whose parameters $\boldsymbol{\pi}_t \in \triangle_m$ (m-dim probability simplex) are computed from the data and previous centers; sampling Q_t chooses the Dirac distribution, which is then "sampled" for one center (and the process iterates). All the crux of the technique is the design of $\boldsymbol{\pi}_t$, which, under *no* assumption of the data, yield in expectation a k-means potential for the centers chosen that is within $8(2 + \log k)$ of the global optimum (Arthur & Vassilvitskii, 2007).

k-variates++ generalize the process in two ways: first, the update of $\boldsymbol{\pi}_t$ depends on data and previous *probes*, using a sequence of *probe functions* $\wp_t : \mathcal{A} \to \mathbb{R}^d$ ($\wp = \text{Id}$, the identity function, $\forall t$, in k-means++). Second, Diracs are replaced by arbitrary but fixed *local* distributions (sometimes called *noisy*) with parameters[2] $(\boldsymbol{\mu}_{\boldsymbol{a}}, \boldsymbol{\theta}_{\boldsymbol{a}})$ depending on $\mathcal{A}$. Let $\mathcal{C}_{\text{opt}} \subset \mathbb{R}^d$ denote the set of k centers minimizing (2) on $\mathcal{A}$. Let $\boldsymbol{c}_{\text{opt}}(\boldsymbol{a}) \doteq \arg\min_{\boldsymbol{c} \in \mathcal{C}_{\text{opt}}} \|\boldsymbol{a} - \boldsymbol{c}\|_2^2$ ($\boldsymbol{a} \in \mathcal{A}$), and

$$\phi_{\text{opt}} \quad \doteq \quad \sum_{\boldsymbol{a} \in \mathcal{A}} \|\boldsymbol{a} - \boldsymbol{c}_{\text{opt}}(\boldsymbol{a})\|_2^2 \ , \qquad (3)$$

$$\phi_{\text{bias}} \quad \doteq \quad \sum_{\boldsymbol{a} \in \mathcal{A}} \|\boldsymbol{\mu}_{\boldsymbol{a}} - \boldsymbol{c}_{\text{opt}}(\boldsymbol{a})\|_2^2 \ , \qquad (4)$$

$$\phi_{\text{var}} \quad \doteq \quad \sum_{a \in \mathcal{A}} \text{tr}\left(\Sigma_{\boldsymbol{a}}\right) \ . \qquad (5)$$

ϕ_{opt} is the optimal **noise-free** potential, ϕ_{bias} is the bias of the noise[3], and ϕ_{var} its variance, with $\Sigma_{\boldsymbol{a}} \doteq \mathbb{E}_{\boldsymbol{x} \sim p_{\boldsymbol{a}}}[(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{a}})(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{a}})^\top]$ the covariance matrix of $p_{\boldsymbol{a}}$. Notice that when $\boldsymbol{\mu}_{\boldsymbol{a}} = \boldsymbol{a}$, $\phi_{\text{bias}} = \phi_{\text{opt}}$. Otherwise, it *may* hold that $\phi_{\text{bias}} < \phi_{\text{opt}}$, and even $\phi_{\text{bias}} = 0$ if expectations coincide

[1] Both approaches can be completed with the same further local monotonous optimization steps like Lloyd or Hartigan iterations; furthermore, it is the biased seeding which holds the approximation properties of k-means++.

[2] Because expectations are the major parameter for clustering, we split the parameters in the form of $\boldsymbol{\mu}_{\boldsymbol{a}}$ (expectation) and $\boldsymbol{\theta}_{\boldsymbol{a}}$ (other parameters, *e.g.* covariance matrix).

[3] We term it *bias* by analogy with supervised classification, considering that the expectations of the densities could be used as models for the cluster centers (Kohavi & Wolpert, 1996).

Algorithm 0 k-variates++

Input: data $\mathcal{A} \subset \mathbb{R}^d$ with $|\mathcal{A}| = m$, $k \in \mathbb{N}_*$, densities $\left\{p_{(\boldsymbol{\mu_a},\boldsymbol{\theta_a})}, \boldsymbol{a} \in \mathcal{A}\right\}$, probe functions $\wp_t : \mathcal{A} \to \mathbb{R}^d$ $(t \geq 1)$;
Step 1: Initialise centers $\mathcal{C} \leftarrow \emptyset$;
Step 2: **for** $t = 1, 2, ..., k$
2.1: randomly sample $\boldsymbol{a} \sim_{q_t} \mathcal{A}$, with $q_1 \doteq u_m$ and, for $t > 1$,

$$q_t(\boldsymbol{a}) \quad \doteq \quad D_t(\boldsymbol{a}) \left(\sum_{\boldsymbol{a}' \in \mathcal{A}} D_t(\boldsymbol{a}') \right)^{-1} \quad , \text{ where } D_t(\boldsymbol{a}) \doteq \min_{\boldsymbol{x} \in \mathcal{C}} \|\wp_t(\boldsymbol{a}) - \boldsymbol{x}\|_2^2 \ ; \tag{1}$$

2.2: randomly sample $\boldsymbol{x} \sim p_{(\boldsymbol{\mu_a},\boldsymbol{\theta_a})}$;
2.3: $\mathcal{C} \leftarrow \mathcal{C} \cup \{\boldsymbol{x}\}$;
Output: $\mathcal{C}$;

with $\mathcal{C}_{\text{opt}}$. Let C_{opt} denote the partition of $\mathcal{A}$ according to the centers in $\mathcal{C}_{\text{opt}}$. We say that probe function $\wp_t$ is η-stretching if, informally, replacing points by their probes does not distort significantly the observed potential of an optimal cluster, with respect to its actual optimal potential. The formal definition follows.

Definition 1 *Probe functions $\wp_t$ are said η-stretching on $\mathcal{A}$, for some $\eta \geq 0$, iff the following holds: for any cluster $A \in C_{\text{opt}}$ with $|A| > 1$, any $\boldsymbol{a}_0 \in A$ such that $\phi(\wp_t(A); \{\wp_t(\boldsymbol{a}_0)\}) \neq 0$, any set of $\leq k$ centers $\mathcal{C} \subset \mathbb{R}^d$,*

$$\frac{\phi(A; \mathcal{C})}{\phi(A; \{\boldsymbol{a}_0\})} \quad \leq \quad (1+\eta) \cdot \frac{\phi(\wp_t(A); \mathcal{C})}{\phi(\wp_t(A); \{\wp_t(\boldsymbol{a}_0)\})}, \forall t \tag{6}$$

Since $\phi(A; \mathcal{C}_{\text{opt}}) = \sum_{\boldsymbol{a}_0 \in A} \phi(A; \{\boldsymbol{a}_0\})$ (Arthur & Vassilvitskii, 2007) (Lemma 3.2), Definition 1 roughly states that the potential of an optimal cluster with respect to a set of cluster centers, relatively to its potential with respect to the optimal set of centers, does not blow up through probe function $\wp_t$. The identity function is trivially 0-stretching, for any $\mathcal{A}$. Many local transformations would be eligible for η-stretching probe functions with η small, including local translations, mappings to core-sets (Har-Peled & Mazumdar, 2004), mappings to Voronoi diagram cell centers (Boissonnat et al., 2010), etc. Notice that ineq. (6) has to hold only for optimal clusters and *not* any clustering of $\mathcal{A}$. Let $\mathbb{E}[\phi(\mathcal{A}; \mathcal{C})] \doteq \int \phi(\mathcal{A}|\mathcal{C}) \mathrm{d}p(\mathcal{C})$ denote the expected potential over the random sampling of $\mathcal{C}$ in k-variates++.

Theorem 2 *For any dataset $\mathcal{A}$, any sequence of η-stretching probe functions $\wp_t$ and any density $\{p_{\boldsymbol{a}}, \boldsymbol{a} \in \mathcal{A}\}$, the expected potential of k-variates++ satisfies:*

$$\mathbb{E}[\phi(\mathcal{A}; \mathcal{C})] \quad \leq \quad (2 + \log k) \cdot \Phi \ , \tag{7}$$

with $\Phi \doteq (6 + 4\eta)\phi_{\text{opt}} + 2\phi_{\text{bias}} + 2\phi_{\text{var}}$.

(Proof in SI, Subsection 2.1) Five remarks are in order. First, we retrieve AV's bound in their setting ($\eta = \phi_{\text{var}} = 0$, $\phi_{\text{bias}} = \phi_{\text{opt}}$) (Arthur & Vassilvitskii, 2007). Second, we *may* beat AV's bound when $\phi_{\text{bias}} < \phi_{\text{opt}}$, which is essentially domain or setting dependent. Third, apart from being η-stretching, there is no constraint on the choice of probe functions $\wp_t$: it can be randomized, iteration dependent, etc. Fourth, the algorithm can easily be generalized to the case where points are weighted. Last, as we show in the following Lemma, the dependence in noise in ineq. (7) can hardly be improved in our framework.

Lemma 3 *Suppose each point in $\mathcal{A}$ is replaced (i.i.d.) by a point sampled in $p_{\boldsymbol{a}}$ with $\Sigma_{\boldsymbol{a}} = \Sigma$. Then any clustering algorithm suffers: $\mathbb{E}[\phi(\mathcal{A}; \mathcal{C})] = \Omega(|\mathcal{A}| \text{tr}\,(\Sigma))$.*

(Proof in SI, Subsection 2.2) We make use of k-variates++ in two different ways. First, we show that it can be used to prove approximation properties for algorithms operating in different clustering settings: distributed, streamed and on-line clustering. The proof involves a *reduction* (explained in SI, Section 2.4) from k-variates++ to each of these algorithms. By reduction, we mean there exists distributions and probe functions (even non poly-time computable) for which k-variates++ yields the same result in expectation as the other algorithm, thus directly yielding an approximability ratio of the global optimum for this latter algorithm via Theorem 2. A key is the choice of the probe functions, which, if not the identity function, models the approximation of the seeding process in the whole set by a *subset* of it, or by an alternative set. Second, we show how k-variates++ can *directly* be specialized for settings for which no efficient application of k-means++ was known.

3. Reductions from k-variates++

Despite tremendous advantages, k-means++ has a serious downside: it is difficult to parallelize, distribute or stream it under relevant communication, space, privacy and/or time resource constraints (Bahmani et al., 2012). Although extending k-means clustering to these settings has been a major research area in recent years, there has been no obvious solution to tailoring k-means++ (Ackermann et al., 2010;

	Ref.	Property	Them	Us
(I)	(Balcan et al., 2013)	Communication complexity	$\Omega((nkd/\varepsilon^4) + n^2 k \ln(nk))$	$O(n^2 k)$
(II)	(Balcan et al., 2013)	Data points shared	$\Omega((kd/\varepsilon^4) + nk \ln(nk))$	k
(III)	(Balcan et al., 2013)	Approximation bound	$(2 + \log k)(1 + \varepsilon) \cdot 8\phi_{\rm opt}$	$(2 + \log k) \cdot \left(10\phi_{\rm opt} + 6\phi_s^F\right)$
(i)	(Ailon et al., 2009)	Time complexity (outer loop)	— identical —	
(ii)	(Ailon et al., 2009)	Approximation bound	$(2 + \log k)(1 + \eta) \cdot 32\phi_{\rm opt}$	$(2 + \log k) \cdot ((8 + 4\eta)\phi_{\rm opt} + 2\phi_s^\wp)$
(a)	(Liberty et al., 2014)	Knowledge required	Lower bound $\phi^* \leq \phi_{\rm opt}$	None
(b)	(Liberty et al., 2014)	Approximation bound	$O(\log m \cdot \phi_{\rm opt})$	$(2 + \log k) \cdot \left(4 + (32/\varsigma^2)\right) \phi_{\rm opt}$
(A)	(Nissim et al., 2007)	Knowledge required	$\lambda(\phi_{\rm opt})$	None
(B)	(Nissim et al., 2007)	Noise variance (σ)	$O(\lambda k R/\epsilon)$	$O(R/(\epsilon + \log m))$
(C)	(Nissim et al., 2007)	Approximation bound	$O^*(\phi_{\rm opt} + m\lambda^2 k R^2/\epsilon^2)$	$O(\log k(\phi_{\rm opt} + mR^2/(\epsilon + \log m)^2))$

Table 1. Comparison with some state of the art approaches for distributed (I-III), streamed (i, ii) and on-line clustering (a, b), and differential privacy (A-C). We refer to related sections for notations. SI (Nock et al., 2016a) provides a more extensive table.

Algorithm 1 Dk-means++ (// PDk-means++)

Input: Forgy nodes $(\mathsf{F}_i, \mathcal{A}_i), i \in [n]$,
for $t = 1, 2, ..., k$
Round 1 : N^* picks $i^* \sim_{q_t^D} [n]$ and asks F_{i^*} for a center;
Round 2 : F_{i^*} picks $\boldsymbol{a} \sim_{u_{i^*}} \mathcal{A}_{i^*}$ and sends $\boldsymbol{a}$ to $\mathsf{F}_i, \forall i$;
// PDk-means++: F_{i^*} sends $\boldsymbol{x} \sim p_{(\boldsymbol{\mu_a}, \boldsymbol{\theta_a})}$ to $\mathsf{F}_i, \forall i$;
Round 3 : $\forall i, \mathsf{F}_i$ updates $D_t(\mathcal{A}_i)$ and sends it to N^*;
Output: $\mathcal{C}$ = set of broadcasted $\boldsymbol{a}$s (or $\boldsymbol{x}$s);

Ailon et al., 2009; Bahmani et al., 2012; Balcan et al., 2013; Liberty et al., 2014; Shindler et al., 2011) (and others).

Distributed clustering We consider horizontally partitioned data among *peers*. Distributed clustering performs a synchronous computation of k-means++, as shown in Algorithm 1. There are two readings of Algorithm 1. Without more assumption about the distributed setting, node N^* denotes one of the peers that randomly selects one of the n peers that is going to sample a centroid with Forgy clustering. We have n such *Forgy nodes*, $(\mathsf{F}_i, \mathcal{A}_i), i \in [n]$, where $\mathcal{A}_i$ is the dataset held by F_i. Our result works regardless of the way N^* is elicited: it can be a fixed peer, a peer that varies through time, or a special node (eventually not being a peer). The notion of having a "special" node in collaborative services is common in a number of distributed domains, *e.g.* in hybrid or server assisted peer-to-peer networks (Yang & Garcia-Molina, 2001). One could also imagine that Forgy nodes are non-computationally intensive and just able to perform uniform sampling in their data, so that N^* is a different node that performs non-uniform sampling. This setting complies with privacy constraints in which data sharing between nodes is limited. In particular, we can enforce that N^* is not allowed to handle *any* data (points) from the Forgy nodes. We therefore split the location of the computational power from the location of the data. We can also prevent the Forgy nodes from exchanging *any* data between themselves, with the sole exception of cluster centers. We note that none of the algorithms of (Ailon et al., 2009; Balcan et al., 2013; Bahmani et al., 2012) would be applicable to this setting without non-trivial modifications affecting their properties.

Algorithm 1 includes two variants: a protected version Dk-means++ where Forgy nodes directly share local centers and a private version PDk-means++ where the nodes share noisy centers, such as to ensure a differentially private release of centers (with relevant noise calibration). Notations used in Algorithm 1 are as follows. Let $D_t(\mathcal{A}_i) \doteq \sum_{\boldsymbol{a} \in \mathcal{A}_i} D_t(\boldsymbol{a})$ and $q_{ti}^D \doteq D_t(\mathcal{A}_i) \cdot (\sum_j D_t(\mathcal{A}_j))^{-1}$ if $t > 1$ and $q_{ti}^D \doteq 1/n$ otherwise. Also, u_i is uniform distribution.

Theorem 4 *Let $\phi_s^F \doteq \sum_{i \in [n]} \sum_{\boldsymbol{a} \in \mathcal{A}_i} \|\boldsymbol{c}(\mathcal{A}_i) - \boldsymbol{a}\|_2^2$ be the total spread of the Forgy nodes ($\boldsymbol{c}(\mathcal{A}_i) \doteq (1/m_i) \cdot \sum_{\mathcal{A}_i} \boldsymbol{a}$). At iteration k, the expected potential on the **total** data $\mathcal{A} \doteq \cup_i \mathcal{A}_i$ satisfies ineq. (7) with*

$$\Phi \doteq \begin{cases} 10\phi_{\rm opt} + 6\phi_s^F & (\text{D}k\textit{-means++}) \\ 10\phi_{\rm opt} + 4\phi_s^F + 2\phi_{\rm var} & (\text{PD}k\textit{-means++}) \end{cases} \quad (8)$$

*Here, $\phi_{\rm opt}$ is the optimal potential on **total** data $\mathcal{A}$.*

(Proof in SI, Subsection 2.4) We note that the optimal potential is defined on the total data. The dependence on ϕ_s^F, which is just the peer-wise variance of data, is thus rather intuitive. A positive point is that ϕ_s^F is weighted by a factor smaller than the factor that weights the optimal potential. Another positive point is that this parameter *can* be computed from data, and among peers, without disclosing more data. Hence, it may be possible to estimate the loss against the centralized, k-means++ setting, taking as reference eq. (8). To gain insight in the leverage that Theorem 4 provides, Table 1 compares Dk-means++ to (Balcan et al., 2013)'s (ε is the coreset approximation parameter), even though the latter approach would not be applicable to our restricted framework. To be fair, we assume that the algorithm used to cluster the coreset in (Balcan et al., 2013) is k-means++. We note that, considering the communication complexity and the number of data points shared, Algorithm 1 is a clear winner. In fact, Algorithm 1 can also win from the approximability standpoint. The dependence in ε

Algorithm 2 Sk-means++

Input: Stream S
Step 1: $\mathcal{S} \doteq \{(\boldsymbol{s}_j, m_j), i \in [n]\} \leftarrow \textsc{Synopsis}(\mathsf{S}, n)$;
Step 2: **for** $t = 1, 2, ..., k$
2.1: **if** $t = 1$ then let $\boldsymbol{s}_j \sim_{u_n} \mathcal{S}$ **else** $\boldsymbol{s}_j \sim_{q_t^S} \mathcal{S}$ s.t.

$$q_t^S(\boldsymbol{s}_j) \doteq m_j D_t(\boldsymbol{s}_j) \left(\sum_{j' \in [n]} m_{j'} D_t(\boldsymbol{s}_{j'}) \right)^{-1} \quad (9)$$

// $D_t(\boldsymbol{s}_j) \doteq \min_{\boldsymbol{c} \in \mathcal{C}} \|\boldsymbol{s}_j - \boldsymbol{c}\|_2^2$;
2.2: $\mathcal{C} \leftarrow \mathcal{C} \cup \{\boldsymbol{s}_j\}$;
Output: Cluster centers $\mathcal{C}$;

prevents to fix it too small in (Balcan et al., 2013). Comparing the bounds in row (III) shows that if $\varepsilon > 1/4$, then we can also be better from the approximability standpoint if the spread satisfies $\phi_s^F = O(\phi_{\rm opt})$. While this may not be feasible over arbitrary data, it becomes more realistic on several real-world scenarii, when Forgy nodes aggregate "local" data with respect to features, *e.g.*, state-wise insurance data, city-wise financial data, etc. When n increases, this also becomes more realistic.

Streaming clustering We have access to a stream S, with an assumed finite size: S is a sequence of points $\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_m$. We authorise the computation / output of the clustering at the end of the stream, *but* the memory n allowed for all operations satisfies $n < m$, such as $n = m^\alpha$ with $\alpha < 1$ in (Ailon et al., 2009). We assume for simplicity that each point can be stored in one storage memory unit. Algorithm 2 (Sk-means++) presents our approach. It relies on the standard "trick" of summarizing massive datasets via compact representations (synopses) before processing them (Indyk et al., 2014). The approximation properties of Sk-means++, proven using a reduction from k-variates++, hold regardless of the way synopses are built. They show that two key parameters may guide its choice: the spread of the synopses, analogous to the spread of Forgy nodes for distributed clustering, and the stretching properties of the synopses used as centers.

Theorem 5 *Let $\wp(\boldsymbol{a}) \doteq \arg\min_{\boldsymbol{s}' \in \mathcal{S}} \|\boldsymbol{a} - \boldsymbol{s}'\|_2^2, \forall \boldsymbol{a} \in \mathsf{S}$. Let $\phi_s^\wp \doteq \sum_{\boldsymbol{a} \in \mathsf{S}} \|\wp(\boldsymbol{a}) - \boldsymbol{a}\|_2^2$ be the spread of $\wp$ on synopses set $\mathcal{S}$. Let $\eta > 0$ such that $\wp$ is η-stretching on S. Then the expected potential of* Sk*-means++ on stream S satisfies ineq. (7) with $\Phi \doteq (8 + 4\eta)\phi_{\rm opt} + 2\phi_s^\wp$. Here, $\phi_{\rm opt}$ is the optimal potential on* ***stream*** *S.*

(Proof in SI, Subsection 2.4) It is not surprising to see that Sk-means++ looks like a generalization of (Ailon et al., 2009) and almost matches it (up to the number of centers delivered) when $k' \gg k$ synopses are learned from

Algorithm 3 OLk-means++

Input: Minibatch S_j, current weighted centers $\mathcal{C}$;
Step 1: **if** $j = 1$ then let $\boldsymbol{s} \sim_{u_1} \mathsf{S}_1$ **else** $\boldsymbol{s} \sim_{q_j^O} \mathsf{S}_j$ s.t.

$$q_j^O(\boldsymbol{s}) \doteq D_t(\boldsymbol{s}) \left(\sum_{\boldsymbol{s}' \in \mathsf{S}_j} D_t(\boldsymbol{s}') \right)^{-1} ; \quad (10)$$

// $D_t(\boldsymbol{s}) \doteq \min_{\boldsymbol{c} \in \mathcal{C}} \|\boldsymbol{s} - \boldsymbol{c}\|_2^2$;
Step 2: $\mathcal{C} \leftarrow \mathcal{C} \cup \{\boldsymbol{s}\}$;

k'-means$_\#$. Yet, we rely on a different — and more general — analysis of its approximation properties. Table 1 compares properties of Sk-means++ to (Ailon et al., 2009) (η relates to approximation of the k-means objective in their inner loop).

On-line clustering This setting is probably the farthest from the original setting of the k-means++ algorithm. Here, points arrive in a sequence, finite, but of unknown size and too large to fit in memory (Liberty et al., 2014). We make no other assumptions – the sequence can be random, or chosen by an adversary. Therefore, the expected analysis we make is only with respect to the internal randomisation of the algorithm, *i.e.*, for the fixed stream sequence as it is observed. We do not assume a feedback for learning (common for supervised learning); so, we do not assume that the algorithm has to predict a cluster for each point that arrives, yet it has to be easily modifiable to do so.

Our approach is summarized in Algorithm 3 (OLk-means++), a variation of k-means++ which consists of splitting the stream S into minibatches S_j for $j = 1, 2, ...$, each of which is used to sample one center. u_1 denotes the uniform distribution with support S_1. Let $R \doteq \max_{\boldsymbol{a}, \boldsymbol{a}' \in \mathsf{S}} \|\boldsymbol{a} - \boldsymbol{a}'\|_2 (\ll \infty)$ be the diameter of S.

Theorem 6 *Let $\varsigma > 0$ be the largest real such that the following conditions are met (for any $A \in C_{\rm opt}, j \geq 1$): for any set of at most k centers $\mathcal{C}$, $\sum_{\boldsymbol{a}, \boldsymbol{a}' \in A} \|\boldsymbol{a} - \boldsymbol{a}'\|_2^2 \geq \varsigma \cdot \binom{|A|}{2} R^2$ and $\sum_{\boldsymbol{a} \in A \cap \mathsf{S}_j} \|\boldsymbol{a} - \boldsymbol{c}(\boldsymbol{a})\|_2^2 \geq \varsigma \cdot \sum_{\boldsymbol{a} \in A} \|\boldsymbol{a} - \boldsymbol{c}(\boldsymbol{a})\|_2^2$ (with $\boldsymbol{c}(\boldsymbol{a})$ defined in eq. (2)). Then the expected potential of* OLk*-means++on stream S satisfies ineq. (7) with $\Phi \doteq (4 + (32/\varsigma^2))\phi_{\rm opt}$, where $\phi_{\rm opt}$ is the optimal potential on* ***stream*** *S.*

(Proof in SI, Subsection 2.4) Notice that loss function $\phi(\mathsf{S}, \mathcal{C})$ in eq. (2) implies the finiteness of S, and the existence of $\varsigma > 0$; also, the second condition implies $\varsigma \leq 1$. In (Liberty et al., 2014), the clustering algorithm is required to have space and time at most polylog in the length of the stream. Hence, each minibatch can be

reasonably large with respect to the stream — the larger they are, the larger ς. The knowledge of ς is not necessary to run OLk-means++; it is just a part of the approximation bound which quantifies the loss in approximation due to the fact that centers are computed from the *partial* knowledge of the stream. Table 1 compares properties of OLk-means++ to (Liberty et al., 2014) (we picked the fully on-line, non-heuristic algorithm). To compare the bounds, suppose that batches have the same size, b, so that $\log k = \log(m/b)$. If batches are at least polylog size, up to what is hidden in the big-Oh notation, our approximation can be quite competitive when ς is large, *e.g.*, if d is large and optimal clusters are not too small.

4. Direct use of k-variates++

The most direct application domain of k-variates++ is differential privacy. Several algorithms have independently emphasised the idea that powerful mechanisms may be amended via a carefully designed noise mechanism to broaden their scope with new capabilities, without overly challenging their original properties. Examples abound (Hardt & Price, 2014; Kalai & Vempala, 2005; Chaudhuri et al., 2011; Chichignoud & Lousteau, 2014), etc. Few approaches are related to clustering, yet noise injected is big — the existence of a smaller, sufficient noise, was conjectured in (Nissim et al., 2007) — and approaches rely on a variety of assumptions or knowledge about the optimum (See Table 1) (Nissim et al., 2007; Wang et al., 2015). To apply k-variates++, we consider that $\wp_t = \mathrm{Id}, \forall t$, and assume $0 < R \ll \infty$ s.t. $\max_{\boldsymbol{a},\boldsymbol{a}' \in \mathcal{A}} \|\boldsymbol{a} - \boldsymbol{a}'\|_2 \leq R$ (a current assumption in the field (Dwork & Roth, 2014)).

A general likelihood ratio bound for k-variates++ We show that the likelihood ratio of the same clustering for two "close" instances is governed by two quantities that rely on the neighborhood function. Most importantly for differential privacy, when densities $p_{(\boldsymbol{\mu}_{\boldsymbol{a}},\boldsymbol{\theta}_{\boldsymbol{a}})}$ are carefully chosen, this ratio *always* $\to 1$ as a function of m, which is highly desirable for differential privacy. We let $\mathrm{NN}_{\mathcal{N}}(\boldsymbol{a}) \doteq \arg\min_{\boldsymbol{a}' \in \mathcal{N}} \|\boldsymbol{a} - \boldsymbol{a}'\|_2$ denote the nearest neighbour of $\boldsymbol{a}$ in $\mathcal{N}$, and let $\boldsymbol{c}(A) \doteq (1/|A|) \cdot \sum_{\boldsymbol{a} \in A} \boldsymbol{a}$.

Definition 7 *We say that neighborhood in $\mathcal{A}$ is δ_w-spread for some $\delta_w > 0$ iff for any $\mathcal{N} \subseteq \mathcal{A}$ with $|\mathcal{N}| = k-1$, and any $\mathcal{B} \subseteq \mathcal{A}$ with $|\mathcal{B}| = |\mathcal{A}| - 1$,*

$$\sum_{\boldsymbol{a} \in \mathcal{B}} \|\boldsymbol{a} - \mathrm{NN}_{\mathcal{N}}(\boldsymbol{a})\|_2^2 \quad \geq \quad \frac{R^2}{\delta_w} \ . \tag{11}$$

Definition 8 *We say that neighborhood in $\mathcal{A}$ is δ_s-monotonic for some $\delta_s > 0$ iff the following holds. $\forall \mathcal{N} \subseteq \mathcal{A}$ with $|\mathcal{N}| \in \{1, 2, ..., k-1\}$, for any $A \subseteq \mathcal{A} \backslash \mathcal{N}$ which is $\mathcal{N}$-packed, we have:*

$$\sum_{\boldsymbol{a} \in \mathcal{A}} \|\boldsymbol{a} - \mathrm{NN}_{\mathcal{N}}(\boldsymbol{a})\|_2^2 \quad \leq \quad (1+\delta_s) \cdot \sum_{\boldsymbol{a} \in \mathcal{A}} \|\boldsymbol{a} - \mathrm{NN}_{\mathcal{N} \cup \{\boldsymbol{c}(A)\}}(\boldsymbol{a})\|_2^2 \ . \tag{12}$$

Set A is said $\mathcal{N}$-packed iff there exists $\boldsymbol{x} \in \mathbb{R}^d$ satisfying $\boldsymbol{x} = \arg\min_{\boldsymbol{c} \in \mathcal{N} \cup \{\boldsymbol{x}\}} \|\boldsymbol{a} - \boldsymbol{c}\|_2^2$, $\forall \boldsymbol{a} \in A$.

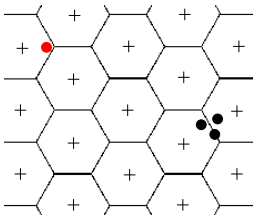

Figure 2. Checking that δ_s is small, for $\mathcal{N}$ the set of crosses (+). Any set A of points close to each other, such as the black dots (•), would be $\mathcal{N}$-packed (pick $\boldsymbol{x} = \boldsymbol{c}(A)$ in this case), but would fail to be $\mathcal{N}$-packed if too spread (*e.g.*, red dot (•) plus black dots). Segments depict the Voronoi diagram of $\mathcal{N}$. Best viewed in color.

It is worthwhile remarking that as long as $k < |\mathcal{A}| \ll \infty$, both $0 < \delta_w \ll \infty$ and $0 < \delta_s \ll \infty$ always exist. Informally, δ_w brings that the sum of squared distances to any subset of $k-1$ centers in $\mathcal{A}$ must not be negligible against the diameter R. δ_s yields a statement a bit more technical, but it roughly reduces to stating that adding one center to any set of at most $k-1$ points that are already close to each other should not decrease significantly the overall potential to the set of centers. Figure 2 provides a schematic view of the property, showing that the modifications of the potential can be very local, thus yielding small δ_s in ineq. (12). The following Theorem uses the definition of neighbouring samples: samples $\mathcal{A}$ and $\mathcal{A}'$ are neighbours, written $\mathcal{A} \approx \mathcal{A}'$, iff they differ by one point. We also define $\mathbb{P}[\mathcal{C}|\mathcal{A}]$ to be the density of output $\mathcal{C}$ given input data $\mathcal{A}$.

Theorem 9 *Fix $\wp_t = \mathrm{Id}$ ($\forall t$) and densities $p_{(\boldsymbol{\mu}_.,\boldsymbol{\theta}_.)}$ having the same support Ω in k-variates++. Suppose there exists $\varrho(R) > 0$ such that densities $p_{(\boldsymbol{\mu}_.,\boldsymbol{\theta}_.)}$ satisfy the following pointwise likelihood ratio constraint:*

$$\frac{p_{(\boldsymbol{\mu}_{\boldsymbol{a}'},\boldsymbol{\theta}_{\boldsymbol{a}'})}(\boldsymbol{x})}{p_{(\boldsymbol{\mu}_{\boldsymbol{a}},\boldsymbol{\theta}_{\boldsymbol{a}})}(\boldsymbol{x})} \quad \leq \quad \varrho(R) \ , \forall \boldsymbol{a}, \boldsymbol{a}' \in \mathcal{A}, \forall \boldsymbol{x} \in \Omega \ . \tag{13}$$

Then, there exists a function $f(.)$ such that, for any $\delta_w, \delta_s > 0$ such that $\mathcal{A}$ is δ_w-spread and δ_s-monotonic, for any $\mathcal{A}' \approx \mathcal{A}$, for any $k > 0$ and any $\mathcal{C} \subset \Omega$ of size k output by Algorithm k-variates++ on whichever of $\mathcal{A}$ or $\mathcal{A}'$, the likelihood ratio of $\mathcal{C}$ given $\mathcal{A}$ and $\mathcal{A}'$ is upperbounded as:

$$\frac{\mathbb{P}[\mathcal{C}|\mathcal{A}']}{\mathbb{P}[\mathcal{C}|\mathcal{A}]} \leq (1+\delta_w)^{k-1} + f(k) \cdot \delta_w \cdot (1+\delta_s)^{k-1} \varrho(R) \tag{14}$$

(Proof in SI, Subsection 2.5) Notice that Theorem 9 makes just one assumption (13) about the densities, so it can be applied in fairly general settings, such as for regular exponential families (Banerjee et al., 2005). These are a key choice because they extensively cover the domain of distortions for which the average is the population minimiser.

An (almost) distribution-free $1 + o(1)$ likelihood ratio We now show that if $\mathcal{A}$ is sampled i.i.d. from any distribution $\mathcal{D}$ which satisfies the mild assumption that it is locally bounded everywhere (or almost surely) in a ball, then with high probability the right-hand side of ineq. (14) is $1+o(1)$ where the little-oh vanishes with m. The proof, of independent interest, involves an explicit bound on δ_w and δ_s.

Theorem 10 *Suppose $\mathcal{A}$ with $|\mathcal{A}| = m > 1$ sampled i.i.d. from distribution $\mathcal{D}$ whose support contains a L_2 ball $\mathscr{B}_2(\mathbf{0}, R)$ with density inside in between $\epsilon_m > 0$ and $\epsilon_M \geq \epsilon_m$. Let $\rho_{\mathcal{D}} \doteq \epsilon_M/\epsilon_m\ (\geq 1)$. For any $0 < \delta < 1/2$, if (i) $\mathcal{A} \subset \mathscr{B}(\mathbf{0}, R)$ and (ii) the number of clusters k meets:*

$$k \quad \leq \quad \frac{\delta^2}{4\rho_{\mathcal{D}}} \cdot \sqrt{m} \ , \qquad (15)$$

then there is probability $1 - \delta$ over the sampling of $\mathcal{A}$ that k-variates++, instantiated as in Theorem 9, satisfies $\mathbb{P}[\mathcal{C}|\mathcal{A}']/\mathbb{P}[\mathcal{C}|\mathcal{A}] \leq 1 + \rho_{\mathcal{D}}^k \cdot g(m, k, d, R)$, $\forall \mathcal{A}' \approx \mathcal{A}$, with

$$g(m, k, d, R) \quad \doteq \quad \frac{4}{m^{\frac{1}{4}+\frac{1}{d+1}}} + \left(\frac{64}{k^{\frac{2}{d}}}\right)^k \cdot \frac{\varrho(2R)}{m} \qquad (16)$$

(Proof in SI, Subsection 2.6) The key informal statement of Theorem 10 is that one may obtain with high probability some "good" datasets $\mathcal{A}$, *i.e.*, for which δ_w, δ_s are small, under very weak assumptions about the domain at hand. The key point is that if one has access to the sampling, then one can resample datasets $\mathcal{A}$ until a good one comes.

Applications to differential privacy Let $\mathscr{M}$ be any algorithm which takes as input $\mathcal{A}$ and k, and returns a set of k centers $\mathcal{C}$. Let $\mathbb{P}_{\mathscr{M}}[\mathcal{C}|\mathcal{A}]$ denote the probability, over the internal randomisation of $\mathscr{M}$, that $\mathscr{M}$ returns $\mathcal{C}$ given $\mathcal{A}$ and k (k, fixed, is omitted in notations). Following is the definition of differential privacy (Dwork et al., 2006), tailored for conciseness to our clustering problem.

Definition 11 *$\mathscr{M}$ is ϵ-differentially private (DP) for k clusters iff for any neighbors $\mathcal{A} \approx \mathcal{A}'$, set $\mathcal{C}$ of k centers,*

$$\mathbb{P}_{\mathscr{M}}[\mathcal{C}|\mathcal{A}']/\mathbb{P}_{\mathscr{M}}[\mathcal{C}|\mathcal{A}] \quad \leq \quad \exp\epsilon \ . \qquad (17)$$

A relaxed version of ϵ-DP is (ϵ, δ)-DP, in which we require $\mathbb{P}_{\mathscr{M}}[\mathcal{C}|\mathcal{A}'] \leq \mathbb{P}_{\mathscr{M}}[\mathcal{C}|\mathcal{A}] \cdot \exp\epsilon + \delta$; thus, ϵ-DP $= (\epsilon, 0)$-DP (Dwork & Roth, 2014). We show that low noise may be affordable to satisfy ineq. (17) using Laplace distribution, $Lap(\sigma/\sqrt{2})$. We refer to the *Laplace mechanism* as a popular mechanism which adds to the output of an algorithm a sufficiently large amount of Laplace noise to be ϵ-DP. We refer to (Dwork et al., 2006) for details, and assume from now on that data belong to a L_1 ball $\mathscr{B}_1(\mathbf{0}, R)$.

Theorem 12 *Using notations and setting of Theorem 9, let*

$$\tilde{\epsilon} \quad \doteq \quad \log\left(\frac{\exp(\epsilon) - (1+\delta_w)^{k-1}}{f(k) \cdot \delta_w \cdot (1+\delta_s)^{k-1}}\right) \ . \qquad (18)$$

*Then, k-variates++ with $p_{(\boldsymbol{\mu}_\cdot, \boldsymbol{\theta}_\cdot)}$ a product of $Lap(\sigma_1/\sqrt{2})$, for $\sigma_1 \doteq 2\sqrt{2}R/\tilde{\epsilon}$, both meets ineq. (17) **and** its expected potential satisfies ineq. (7) with*

$$\Phi \quad = \quad \Phi_1 \doteq 8 \cdot \left(\phi_{\text{opt}} + \frac{mR^2}{\tilde{\epsilon}^2}\right) \ . \qquad (19)$$

*On the other hand, if we opt for $\sigma_2 \doteq 2\sqrt{2}kR/\epsilon$, then k-variates++ is an instance of the Laplace mechanism **and** its expected potential satisfies ineq. (7) with*

$$\Phi \quad = \quad \Phi_2 \doteq 8 \cdot \left(\phi_{\text{opt}} + \frac{mk^2R^2}{\epsilon^2}\right) \ . \qquad (20)$$

(Proof in SI, Subsection 2.7) A question is how do σ_1 (resp. Φ_1) and σ_2 (resp. Φ_2) compare with each other, and how do they compare to the state of the art (Nissim et al., 2007; Wang et al., 2015) (we only consider methods with provable approximation bounds of the global optimum). The key fact is that, if m is sufficiently large, then it happens that we can fix $\delta_w = O(1/m)$ and $\delta_s = O(1)$. The proof of Theorem 10 in SI formalizes this intuition (SI, Subsection 2.6) and the experiments (SI, Section 3) display that such regimes *are* indeed observed. In this case, it is not hard to show that $\tilde{\epsilon} = \Omega(\epsilon + \log m)$, granting $\sigma_1 = o(\sigma_2)$ since

$$\sigma_1 \quad = \quad O\left(\frac{R}{\epsilon + \log(m)}\right) \ , \qquad (21)$$

i.e. the noise guaranteeing ineq. (17) *vanishes* at $1/\log(m)$ rate. Consequently, in this regime, Φ_1 in eq. (19) becomes:

$$\Phi_1 \quad = \quad \tilde{O}\left(\phi_{\text{opt}} + \frac{mR^2}{(\epsilon + \log m)^2}\right) \ , \qquad (22)$$

ignoring all factors other than those noted. Thus, the noise dependence grows *sublinearly* in m. Since in this setting, unless all datapoints are the same, δ_w and δ_s for $\mathcal{A}$ and *any* possible neighbor $\mathcal{A}'$ are within $1 + o(1)$, it is also possible to overestimate δ_w and δ_s to still have $\delta_w = O(1/m)$ and $\delta_s = O(1)$ **and** grant ϵ-DP for k-variates++. Otherwise, the setting of Theorem 10 can be used to grant (ϵ, δ)-DP without any tweak. Table 1 compares k-variates++ to (Nissim et al., 2007) in this large sample regime, which is actually a prerequisite for (Nissim et al., 2007).

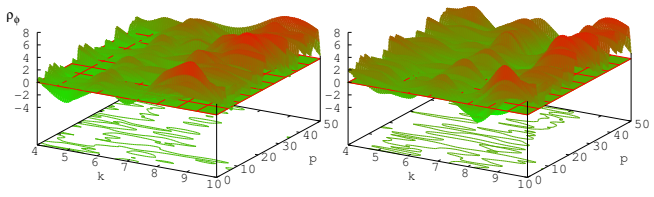

Figure 3. Plot of $\rho_\phi(\mathcal{H}) = f(k,p)$ (points below $z = 0$ — isocontour shown — correspond to superior performances for Dk-means++). Left: $\mathcal{H}$=k-means++; right: $\mathcal{H}$=k-means$_\|$ (best viewed in color).

Dataset	m	d	$\overline{k}$	$\overline{(\tilde{\epsilon}/\epsilon)}$	$\overline{\rho'_\phi\,(\text{F-DP})}$	$\overline{\rho'_\phi\,(\text{GUPT})}$
LifeSci	26 733	10	3	4.5	163.0	0.7
Image	34 112	3	2.5	7.9	188.5	2.9
EuropeDiff	169 308	2	5	13.0	2857.1	40.4

Table 2. k-variates++ vs F-DP and GUPT (see text).

5. Experiments

Due to the large number of experiments carried out, the overview we provide appears *in extenso* in SI (Section 3).

Dk-means++ vs k-means++ and k-means$_\|$ (Bahmani et al., 2012) To address algorithms that can be reduced from k-variates++ (Section 3), we have tested Dk-means++ vs state of the art approach k-means$_\|$; to be fair with Dk-means++, we use k-means++ seeding as the reclustering algorithm in k-means$_\|$. Parameters are in line with (Bahmani et al., 2012). To control the spread of Forgy nodes ϕ_s^F (Theorem 4), each peer's initial data consists of points uniformly sampled in a random hyperrectangle in a space of $d = 50$ (expected number of peers points $m_i = 500, \forall i$). We sample peers until a total of $m \approx 20000$ point is sampled. Then, each point moves with $p\%$ chances to a uniformly sampled peer. We checked that ϕ_s^F blows up with p, *i.e.*, >20 times for $p = 50\%$ with respect to $p = 0$. A remarkable phenomenon was the fact that, even when the number of peers n is quite large (dozens on average), Dk-means++ is able to *beat* both k-means++ and k-means$_\|$, even for large values of p, as computed by ratio $\rho_\phi(\mathcal{H}) \doteq 100 \cdot (\phi(\text{D}k\text{-means++}) - \phi(\mathcal{H}))/\phi(\mathcal{H})$ for $\mathcal{H} \in \{k\text{-means++}, k\text{-means}_\|\}$ (Figure 3). Another positive point is that the amount of data to compute a center for Dk-means++ is in average $\approx n$ times smaller than k-means$_\|$.

k-variates++ vs Forgy-DP and GUPT To address algorithms that can be obtained via a direct use of k-variates++ (Section 4), we have tested it in a differential privacy framework vs state of the art approach GUPT (Mohan et al., 2012). We let $\epsilon = 1$ in our experiments. We also compare it to Forgy DP (F-DP), which is just Forgy initialisation in the Laplace mechanism, with noise rate (standard dev.) $\propto kR/\epsilon$. In comparison, the noise rate for GUPT is $\propto kR/(\ell\epsilon)$ at the end of its aggregation process, where ℓ is the number of blocks. Table 2 gives results for the average (over the choices of k) parameters used, k, $\overline{\tilde{\epsilon}}$, and ratio $\overline{\rho'_\phi}$ where $\rho'_\phi(\mathcal{H}) \doteq \phi(\mathcal{H})/\phi(k\text{-variates++})$ — values above 1 indicate better results for k-variates++. We use $\tilde{\epsilon}$ as the equivalent ϵ for k-variates++, *i.e.* the value that guarantees ineq. (17). From Theorem 12, when $\tilde{\epsilon} > \epsilon$, this brings a smaller noise magnitude, desirable for clustering. The obtained results show that k-variates++ becomes more of a contender with increasing m, but its relative performance tends to decrease with increasing k. This is in accordance with the "good" regime of Theorem 12. Results on synthetic domains display the same patterns, along with the fact that relative performances of k-variates++ improves with d, making it a relevant choice for "big" domains.

In fact, extensive experiments on synthetic data (Nock et al., 2016a) show that intuitions regarding the sublinear noise regime in eq. (22) are experimentally observed, and furthermore they may happen for quite small values of m.

6. Discussion and Conclusion

We first show in this paper that the k-means++ analysis of Arthur and Vassilvitskii can be carried out on a significantly more general scale, aggregating various clustering frameworks of interest and for which no trivial adaptation of k-means++ was previously known. Our contributions stand at two levels: (i) we provide the "meta" algorithm, k-variates++, and two key results, one on its approximation abilities of the *global* optimum, and one on the *likelihood ratio* of the centers it delivers. We do expect further applications of these results, in particular to address several other key clustering problems: stability, generalisation and smoothed analysis (Arthur et al., 2011; von Luxburg, 2010); (ii) we provide two examples of application. The first is a reduction technique from k-variates++, which shows a way to obtain straight approximabilty results for other clustering algorithms, some being efficient proxies for the generalisation of existing approaches (Ailon et al., 2009). The second is a direct application of k-variates++ to differential privacy, exhibiting a noise component significantly better than existing approaches (Nissim et al., 2007; Wang et al., 2015).

Due to the lack of space, we refer to the SI (Nock et al., 2016a) for the extension of our results to "extreme cases" of distortions (other than L_2^2) that do not even admit population minimizers in closed form (Nielsen & Nock, 2015) — clustering being a huge practical problem, it is indeed reasonable to tailor the distortion to the application at hand.

Acknowledgments

Thanks are due to Stephen Hardy, Guillaume Smith, Wilko Henecka and Max Ott for stimulating discussions and feedback on the subject. Work carried out in NICTA which was supported by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Center of Excellence Program.

References

Ackermann, M.-R., Lammersen, C., Märtens, M., Raupach, C., Sohler, C., and Swierkot, K. Streamkm++: A clustering algorithms for data streams. In *12th ALENEX*, pp. 173–187, 2010.

Ailon, N., Jaiswal, R., and Monteleoni, C. Streaming k-means approximation. In *NIPS*22*, pp. 10–18, 2009.

Arthur, D. and Vassilvitskii, S. k-means++ : the advantages of careful seeding. In *19th SODA*, pp. 1027 – 1035, 2007.

Arthur, D., Manthey, B., and Röglin, H. Smoothed analysis of the k-means method. *JACM*, 58:19, 2011.

Bahmani, B., Moseley, B., Vattani, A., Kumar, R., and Vassilvitskii, S. Scalable k-means++. In *38th VLDB*, pp. 622–633, 2012.

Balcan, M.-F., Ehrlich, S., and Liang, Y. Distributed k-means and k-median clustering on general communication topologies. In *NIPS*26*, pp. 1995–2003, 2013.

Banerjee, A., Merugu, S., Dhillon, I., and Ghosh, J. Clustering with Bregman divergences. *JMLR*, 6:1705–1749, 2005.

Boissonnat, J.-D., Nielsen, F., and Nock, R. Bregman voronoi diagrams. *DCG*, 44(2):281–307, 2010.

Chaudhuri, K., Monteleoni, C., and Sarwate, A.-D. Differentially private empirical risk minimization. *JMLR*, 12: 1069–1109, 2011.

Chichignoud, M. and Lousteau, S. Adaptive noisy clustering. *IEEE Trans. IT*, 60:7279–7292, 2014.

Dwork, C. and Roth, A. The algorithmic foudations of differential privacy. *Found. & Trends in TCS*, 9:211–407, 2014.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *3rd TCC*, pp. 265–284, 2006.

Har-Peled, S. and Mazumdar, S. On coresets for k-means and k-median clustering. In *37th ACM STOC*, pp. 291–300, 2004.

Hardt, M. and Price, E. The noisy power method: a meta algorithm with applications. In *NIPS*27*, pp. 2861–2869, 2014.

Indyk, P., Mahabadi, S., Mahdian, M., and Mirrokni, V.-S. Composable core-sets for diversity and coverage maximization. In *33rd ACM PODS*, pp. 100–108, 2014.

Jegelka, S., Sra, S., and Banerjee, A. Approximation algorithms for tensor clustering. In *20th ALT*, pp. 368–383, 2009.

Kalai, A. and Vempala, S. Efficient algorithms for online decision problems. *J. Comp. Syst. Sc.*, pp. 291–307, 2005.

Kohavi, R. and Wolpert, D. Bias plus variance decomposition for zero-one loss functions. In *13th ICML*, pp. 275–283, 1996.

Liberty, E., Sriharsha, R., and Sviridenko, M. An algorithm for online k-means clustering. *CoRR*, abs/1412.5721, 2014.

Mohan, P., Thakurta, A., Shi, E., Song, D., and Culler, D.-E. GUPT: privacy preserving data analysis made easy. In *38th ACM SIGMOD*, pp. 349–360, 2012.

Nielsen, F. and Nock, R. Optimal interval clustering: Application to bregman clustering and statistical mixture learning. *IEEE Signal Processing Letters*, 21:1289–1292, 2014.

Nielsen, F. and Nock, R. Total Jensen divergences: definition, properties and clustering. In *40th IEEE ICASSP*, pp. 2016–2020, 2015.

Nielsen, F., Nock, R., and S.-I.Amari. On clustering histograms with k-means by using mixed α-divergences. *Entropy*, 16, 2014.

Nissim, K., Raskhodnikova, S., and Smith, A. Smooth sensitivity and sampling in private data analysis. In *40th ACM STOC*, pp. 75–84, 2007.

Nock, R., Luosto, P., and Kivinen, J. Mixed Bregman clustering with approximation guarantees. In *19th ECML*, pp. 154–169, 2008.

Nock, R., Canyasse, R., Boreli, R., and Nielsen, F. k-variates++: more pluses in the k-means++ (supplementary information to this paper). In *33rd ICML*, 2016a.

Nock, R., Nielsen, F., and Amari, S.-I. On conformal divergences and their population minimizers. *IEEE Trans. IT*, 62:1–12, 2016b.

Shindler, M., Wong, A., and Meyerson, A. Fast and accurate k-means for large datasets. In *NIPS*24*, pp. 2375–2383, 2011.

von Luxburg, U. Clustering stability: an overview. *Found. & Trends in ML*, 2(3):235–274, 2010.

Wang, Y., Wang, Y.-X., and Singh, A. Differentially private subspace clustering. In *NIPS*28*, 2015.

Yang, B. and Garcia-Molina, H. Comparing hybrid peer-to-peer systems. In *27^{th} VLDB*, pp. 561–570, 2001.