The value of analytical queries on Social Networks

Michel de Rougemont
University of Paris II
and LIAFA-CNRS

Guillaume Vimont
University Paris II

***Abstract*—We study analytical queries for two models of Social Networks. Firstly, a datawarehouse which can be analyzed along some OLAP schema and possible dimensions, secondly a model of streaming data which need to be transformed before they are analyzed. In the first case, the linear influence model is generalized for each possible dimensions and provides densities of influence types. In the second case, we need to approximate the analytical queries by sampling the data along some specific distributions. For a model of analytical queries on graphs and their approximations, we give examples of approximable and non approximable queries. We introduce a measure to quantify the information provided by various analyses using both the Entropy of the answer to the query and of the influence types. We illustrate the approach on Facebook and Twitter data.**

I. Introduction

Social Networks such as Facebook or Twitter provide many useful data which can be helpful to predict real-world outcomes, properties of these networks as in [7] or the dynamics of the network as in [9]. In this paper we study analytical queries on a datawarehouse (Facebook model) and on Streaming data (Twitter Model), relative to a Schema. We give a model for the information value of these queries, based on the Entropy. As an example of a datawarehouse, we analyze Facebook movie pages in order to predict movies revenues as in [1], [10] but refine the analysis along various dimensions such as Gender, Location, Types of interactions. As an example of streaming data, we analyze Twitter data on keywords and their Twitter graphs.

For simplicity, we concentrate on OLAP (OnLine Analytical Processing) queries whose answers can be considered as distributions. The support is the domain of the dimensions on which the values are greater than some ε, some percentage value such as 1%. We wish to compare different analyses and associate a value to each of them. The experience shows that some analyses are more informative than others, and we are looking for a formal model which generalizes the classical Information Gain. We don't consider the visualization techniques which play an important role for the quality of the dashboards and only compare the information associated with the distributions.

One classical way to analyze a datawarehouse is to ask for OLAP queries defined on an OLAP schema. A Web site may collect all the users' data and load them in a datawarehouse. It may provide an OLAP schema which lists the possible dimensions and their dependencies to analyse the data.

In [10], the analysis of Facebook movie pages is used to predict the number of ticket sales of a movie, using the Linear Influence Model [12] and then the possible revenues of a movie. We refine this approach by analyzing the *Engagements*, i.e. the possible Likes, Shares, Comments users can generate on a Facebook page, collected in a datawarehouse depending on the profile of the users. An implicit schema provides possible dimensions such as Time, Gender, Location, Types to analyze these data. We can measure the number of Engagements per Gender and Time, or per Location and Time, and many more elaborate queries.

We then define a formal model to quantify the value of these analyses, to infer which analysis gives more information. On the Facebook example, we deduce in section 6 that the analysis per Location and Time is more informative than the analysis per Gender and Time.

A different model, followed by Twitter, is based on streaming data. In this case, we can only approximately answer these queries if we assume that we can't store all the data. A general problem is to bound the number of local queries on XML streams as in [3], [6] to test various properties. For the Streaming Property Testing model introduced in [4], $O(poly(\log n/\varepsilon))$ local samples of the online XML stream with n nodes allow to decide if an XML tree is valid or $\varepsilon-$far from an XML schema. The answers to OLAP queries, viewed as a distribution, can be approximated with L_1 distance with similar techniques.

An ETL (Extract Transform Load) step transforms the Json stream into an enriched structure. In Twitter's case, the Json tree is transformed into a graph. We then enrich the graph with communities and query this new structure along several dimensions. We define a class of analytical queries for these graphs with communities and provide two examples which illustrate the role of the approximation, when we take online samples (nodes of the graphs with their neighborhoods) with the degree distribution or with the uniform distribution. Our main results show that one natural analytical query A_1 is approximable for the degree distribution, but not for the uniform distribution, and conversely the query A_2 is approximable for the uniform distribution, but not for the degree distribution. We argue that the classical ETL procedure has to be extended. An ETCL (Extract, Transform, Compress, Load) includes a *structural compression* which uses specific samplings.

In the general case, we have several sources which we have to integrate and we need to construct the sketches which will be used to approximate analytic queries. We clearly need a tool to compare the value of the different analyses. In this paper we propose a solution for one source and illustrate it for Facebook and Twitter data.

In the second section, we set the framework of OLAP queries and their approximation, and of the linear Influence model. In the third section, we describe a classical datawarehouse with an OLAP schema and the Entropy based model to compare analytical queries. In the fourth section, we describe streaming data and the transformation as a graph with communities. In the fifth section, we define our model of analytical queries on such graphs and our main results. In the sixth section, we describe our experiments on Facebook and Twitter.

II. Preliminaries

We concentrate on simple OLAP queries for analytic queries and their approximation on streaming data. We then recall the Linear Influence Model [12] which defines the Influence profile of users from the observed measures. We want to combine both the information of the analysis and of the profile to define a value of the query.

A. OLAP queries

We follow the functional model associated with an OLAP schema, i.e. the OLAP or star schema is a tree where each node is a set of attributes, the root is the set of all the attributes of the data warehouse relation, and an edge exists if there is a functional dependency between the attributes of the origin node and the attributes of the extremity node, as in Figure 1. This model is usually used for relational data, where

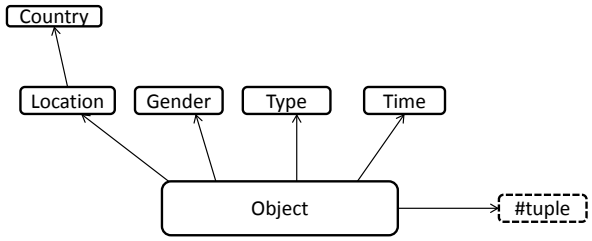

Fig. 1. A simplified OLAP schema for Facebook.

atttributes are the columns and one table is considered as the datawarehouse. It can also be extended to XML trees, where nodes have labels and attributes. The *measure* is a specific node of the schema at depth 1 from the root. The simple OLAP schema of Figure 1, where the functional dependencies follow the edges up, will be used for Facebook data. For each click on a Web page, a tuple with the user's information, the time, the type of *Engagement*, i.e. *Like, Share, Comment*, is recorded in a datawarehouse.

An OLAP query for a schema S is determined by: a filter condition, a *measure*, the selection of dimensions or classifiers, $C_1, ...C_p$ where each C_i is a node of the schema S, and an aggregation operator (COUNT, SUM, AVG, ...). A filter is a condition which selects a subset of the tuples of the datawarehouse (in the relational model), and we assume for simplicity that SUM is the Aggregation Operator. The answer to an OLAP query is a multidimensional array, along the dimensions $C_1, ...C_p$ and the *measure* M. Each tuple $c_1, ..., c_p, m_i$ of the answer is such that

$$m_i = \frac{\sum_{t:t.C_1=c_1,...t.C_p=c_p} t.M}{\sum_{t\in I} t.M}$$

We consider relative *measures* as answers to OLAP queries, i.e. as a distribution and write Q^I_C as the density vector for the answer to Q on dimension C and on datawarehouse I. Each component of Q^I_C is written $Q^I_{C=c}$, the relative density for the dimension $C = c$. Figures 8 and 9 are answers to 1-dimensional OLAP query on Gender and Location respectively, given a filter on a time interval.

B. Approximate answers to queries

Many queries can be efficiently approximated, in particular analytical queries, using randomized algorithms. There are different types of approximation, depending on the queries:

- For boolean queries, we decide if the query is true or ε-far with high probability, as in Property Testing [8],
- For unary queries, we use the Hamming distance between sets,
- For OLAP queries (distributions), we use the L_1 distance.

For streaming data, such as Json or XML streams provided by Twitter, we need to approximate OLAP queries, as we can't store all the data. As in [3], we approximate the distribution by constructing partial sketches. An exact answer to the Gender query of Figure 8 might be $\{female : 80\%, male : 20\%\}$, and an approximate answer might be $\{female : 82\%, male : 18\%\}$ with a 4% error.

C. Linear Influence Model

A Web site records the time of an interaction and the users profiles. The simplest analysis is the number of interactions in a time interval, an analysis over the dimension time with a time interval as a filter, a unit measure and COUNT as an operator. We may want to infer the Influence of the users, as a Social Network diffuses the information over the user's friends. Given the activity measure $x(t)$ at time t, a useful view is to infer an Influence profile $I_u(t)$ for each user u, following the linear model introduced in [12]. We will generalize this approach by introducing Influence vectors for various values of the dimensions, for example we will have $I(t, f)$ for *female* and $I(t, m)$ for *male* and generally $I(t, a_1, ...a_k)$ for k dimensions and values a_i for the i-th dimension. We first review the global model.

Given a network, the influence of a node u, represented by the vector $I_u(t)$, measures the influence of user u at time t. Let i be some information, or some web site on which we observe various Engagements. The simplest form is the *like* of some users. Let $x_i(t)$ be the number of such Engagements for web site i, $A(i)$ be the set of users who interacted with i, and t_u the time when user u made the engagement. We model $x_i(t)$ and $I_u(t)$ as vectors of dimension n and let be m users engaged between t and $t - t_0$ at times $t_1, ...t_m$.

$$x_i(t) = \sum_{u \in A(i)} I_u(t - t_u)$$

We may represent these linear equations as:

$$X_i = M.I$$

where X_i is the observation vector for some finite interval, the number of interactions (Engagements for Facebook) of site i at time t, M the $(n, n.m)$ delay matrix determined by the $t_1...t_m$'s, and I the unknown influence vector $(I_1, ...I_m)^t$. More precisely $M(i, j) = 1$ if $j = t_1 - (i - 1)$ or $j = n + t_2 - (i - 1)$,....or $j = (m - 1).n + t_m - (i - 1)$ and 0 otherwise, for $i = 1, ...n$ and $j = 1...m.n$. We have a classical inverse problem for which many efficient solutions exist [2]. One may look for I such that:

$$Min_I |X - M.I|^2$$

Efficient solutions can be found, using for example the pseudo inverse M^+, computed with an SVD decomposition. Hence we look for the values I such that

$$e = Argmin_I |X - M * I|^2$$

In [10], the model is extended to project revenues, with a simple linear regression. More sophisticated learning models based on inverse problems [2] can be used to infer a model from few measurements.

III. Analysis on datawarehouses

When users interact with a Web page, they transfer part of their identity, as their gender, their locations and potentially much more information to the Web site. In the case of Facebook pages, a user may *Like, Share, Comment* and we will say that any of these actions is an *Engagement*. To analyze these data, OLAP queries on dimensions such as Time, Gender, Location,... can be very useful as they differentiate the various profiles.

For each possible dimensions, we can measure the number of *Engagements* for the different values of the dimensions and stipulate that the influence vector $I(t)$ also depends on these values. Assume a dimension *Location* with possible values $\{a, b, c\}$. We then write $x_i(t + 1, a)$ for the number of *Engagements* from users with Location$= a$, and similarly for $I_u(t, a)$, as the influence of user u with Location$= a$.

A. LIM-based Models

In a linear model, we then write, for each $a, b, c \in L$:

$$x(t + 1, a) = \sum_{u \in A(i)} I_u(t - t_u, a)$$

and similarly for b, c. In a compact form

$$X_a = M_a.I_a$$

and similarly for b and c. For the same time intervals, we have 3 times more variables, and equations.

Let M the matrix, with M_a, M_b, M_c as diagonal sub matrices, and 0 elsewhere. In this case, we have 3 independent optimization problems of the type:

$$e_a = Argmin_{I_a} |X_a - M_a * I_a|^2$$

and similarly for b, c.

The global error is then $e = e_a + e_b + e_c$, and in general $e = \sum_{a \in L} e_a$. In the case of a uniform distribution for the x_a, x_b, x_c, we get similar variables for I_a, I_b, I_c and the error $e = 3.e_a$. When the distributions are asymmetric, the error will decrease, and the model will improve.

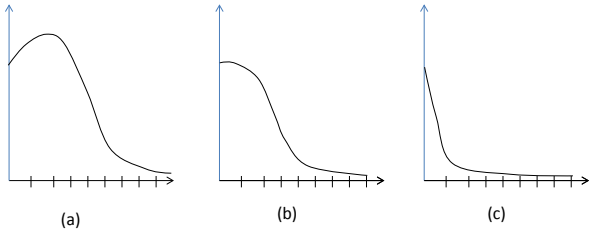

Fig. 2. Three different profiles of Influence: positive (+) in (a), neutral in (b), negative (-) in (c)

B. A simple example

Consider the situation where the global analysis of Engagements is a flat curve, but the analysis for Men is decreasing, the analysis for Women is increasing, as in Figure 3. We suppose 10 users regularly spaced in

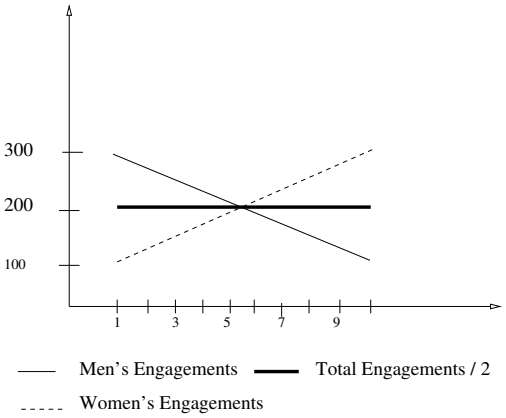

Fig. 3. Analysis of Engagements per Gender over time

time: users 1, 3, 5, 7, 9 are Men, users 2, 4, 6, 8,10 are Women. In this case, the analysis per Gender gives clearly more information than the global analysis which is entirely uniform.

If we solve the inverse linear problem for low-rank vectors $I(t)$, $I(t, m)$ and $I(t, f)$ we will obtain mostly vectors of type neutral for $I(t)$, vectors of negative type for $I(t, m)$ and vectors of positive type for $I(t, f)$.

One way to formalize the intuition that non uniform distributions carry more information than uniform distributions is to use Entropy based measures. We consider both the Entropy of the measure (the curve $x(t)$ or $x(t, m)$) and the Entropy of the $I(t)$ or $I(t, m)$, viewed as a distribution. If the inverse problem gives 3 negative and 2 neutral profiles out of the 5 users, the distribution is $\{+ : 3/5, neutral : 2/5\}$.

C. Entropy and Information gain

There are many measures based on the Entropy to compare distributions, the basic Entropy (Shannon or Renyi), the relative Entropy and the Kullback-Liebler divergence. We will make variations on the relative Entropy.

For a Source S, i.e. a distribution, $Ent(S) = -\sum_i p_i \log p_i$. Given a dimension A of the OLAP Schema, the relative Entropy on the dimension A is

$$Ent(S|A) = \sum_{\nu \in A} \frac{|S_\nu|}{|S|}.Ent(S_\nu)$$

where $\frac{|S_\nu|}{|S|}$ is the relative weight for ν and $Ent(S_\nu)$ is the Entropy of the relative source, i.e. when $A = Ent(S_\nu)$.

We view S, S_m, S_f as the 3 distributions for Figure 2, where S is the global source, i.e. $x(t)$, and S_m (resp. S_f) is the source where *Gender=males* (resp. *Gender=females*), i.e. $x(t, m)$ (resp. $x(t, f)$).

In our simple example, $Ent(S) = 1/10$ as S is the uniform distribution, whereas $Ent(S_m) = Ent(S_f) = \log 5 - 3/10. \log 3$, as the distribution over 5 points for S_m is given by the values $\{2/5, 3/10, 1/5, 1/10, 0\}$. Then $Ent(S|Gender) = \log 5 - 3/10. \log 3 =< \log 10$. The Entropy decreased, and the Information Gain increased.

When we solve the inverse linear problem over low-rank vectors, assume we obtain a distribution of profiles I over the gobal data, and I_m over the male curve and I_f over the female curve.

Let the *Global relative Entropy* sums the factors $Ent(S_\nu)$ and $Ent(I_\nu)$, i.e.

$$GH(S|A) = \sum_{\nu \in A} \frac{|S_\nu|}{|S|}.[Ent(S_\nu) + Ent(I_\nu)]$$

If A is the $Gender$ property there are two ν's distributions, one for $\nu = f$ and one for $\nu = m$. GH averages the Entropy of the measures with the Entropy of the predicted influences I and varies between 0 and some positive value. If there is no attribute, then $GH(S|) = [Ent(S) + Ent(I)]$.

Definition 1: The Gain of Information for the dimension A is the function

$$Gain(S|A) = \frac{1}{1 + GH(S|A)}$$

This new function varies between 0 and 1. The optimal value (maximal gain) is 1 when the Global relative Entropies are 0, i.e. both Dirac distributions. This function generalizes for several dimension $Gain(S|A_1, ...A_k)$ and it is then possible to compare an analysis on one attribute to an analysis on several attributes.

Gain on the simple example. In the previous example let S, S_m, S_f be the 3 distributions corresponding to Figure 3 and assume I, I_m and I_f are Dirac distributions (with a 0 entropy). Then $GH(S|Gender) = \log 5 - 3/10. \log 3 = 1.79$, compared with $GH(S|) = \log 10 = 3.32$. We can then compare the two gains: $Gain(S|Gender) = 0.36 > Gain(S|) = 0.23$, as we would intuitively guess.

IV. Sketches on Streaming data

A fundamental operation on data sources is the ETL (*Extract Transform Load*) paradigm which combines Data Exchange and Transformations into Target Schemas. Classes of analytical queries can then be defined on the Target Schema, which generalizes the OLAP Schema.

We add a crucial step in this paradigm, *Structural Compression*, to enrich the Transformation step of the ETL analysis. We sample the original structure with some distribution and define a new structure which we view as an approximate compression of the stream, often called *a sketch*. We want to guarantee that the analytical queries on the original structure are well approximated by some other analytical queries on the compressed structure. This compression is not the the classical compression as in Lempel-Ziv which leads to .zip files, but a structural compression which leads to a different structure obtained with a randomized process.

In the general situation of multisources with different schema, the construction of combined sketches and of a global schema remains a fundamental problem. We propose an approach on streaming data which may be extended to cope with the Data Integration.

A. Json and XML trees

A Json or XML stream is a sequence of opening tags followed by closing tags in a well parenthesis manner, along a depth first representation of a tree. We wish to approximately decide a property of the tree, without storing all the tree, but with only two complementary operations: taking samples and compressing the tree.

Property Testing is a framework for approximate decision problems with a distance between structures such as words or trees. We use the *Edit distance* on labeled unranked ordered trees. Basic operations are *edges insertions, edge deletions, labels modifications* at a unit cost. The absolute distance between two trees t, t' is the minimum size of a set of basic operations which transform t into t'. The relative distance $0 \leq dist(t, t') \leq 2$ is the absolute distance divided by the largest size (number of nodes) of t, t'. Let a property P of trees be a subset of trees. The distance of t to P is $dist(t, P) = Min_{t' \in P}\ dist(t, t')$.

Given a parameter $0 \leq \varepsilon \leq 1$, an *$\varepsilon$-tester* [8] for a property P decides if a structure satisfies the property P or if it is ε-far from satisfying the property P. A property is *testable* if for all ε, there exists an ε-tester whose time complexity is independent of the size of the structure and only depends on ε. If the time complexity is $O(poly(\log n))$ we say that the property is *polylog testable*.

In [5] we showed that regular tree properties were testable with the edit distance with moves, a distance weaker than the classical edit distance and in [6], we showed that regular tree properties were polylog testable but not testable for the edit distance. It implies that OLAP queries can be approximated with these techniques [3] which can be summarized by two ideas:

- Reservoir sampling [11] maintains a uniform distribution over k nodes on a stream must be first generalized to handle weighted distributions online. Some local peaks are compressed to new letters with a weight proportional to the size of the peak, and samples must follow such distributions. A *dichotomy sampling* on a word consists of suffixes in geometrical progression. We need such a sampling along the partial peaks of the stream, i.e. the paths of the trees. For each suffix, we take samples with a weighted distribution.
- Balanced peaks of the trees must be compressed as pairs of states.

B. Twitter Graph

Given a stream of Json trees, or tweets, we construct the *Twitter Graph* of the stream, i.e. the graph $G = (V, E)$ with multiple edges $E \subseteq V.V$ where V is the set of tags ($\#x$ or $@y$) seen and for each tweet sent by $@y$ which contains tags $\#x$,$@z$ edges $(@y, \#x)$ and $(@y, @z)$ are generated in E. This graph can be extended as a structure $G = (V, E, d)$ where d is the degree function ($d: V \to N$) which gives the number of edges from a node $u \in V$.

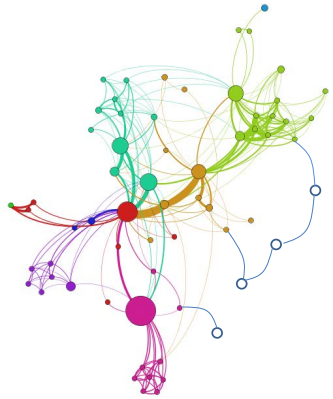

Fig. 4. Twitter graph with communities, colors of the nodes

Social Networks tools such as *Gephi,* transform a Json stream, i.e. a tree, into a graph in an online manner. New nodes and new edges appear online, and a key point is that the graph has a specific structure. It is connected, the degree distribution of the nodes follows a power law and the graph satisfies many other properties.

Samples in graphs are classically of two types. On sparse graphs, we take p random nodes and their neighborhoods at distance k. On dense graphs, we take the subgraph spanned by p random nodes. In Social networks, part of the graph is dense and part is sparse, so we take samples in both sense for some values of k and p. The resulting subgraph is much smaller than the original graph, typically of size $poly(\log n, \log m)$ if the graph has n nodes and m edges. The samples can be taken with at least two specific distributions.

- The degree distribution. Each node $i \in V$ is taken with probability $\frac{d(i)}{\sum_i d(i)}$.
- The uniform distribution. Each node $i \in V$ is taken with probability $\frac{1}{n}$ if $|V| = n$.

The first distribution requires an explicit representation of the degree function d, whereas the the second distribution requires an explicit representation of the domain. Both can be maintained online with a memory $O(n + m)$. In practical applications, we periodically reset the graph and only store the samples.

C. Graph extensions

There are many classical extensions of social graphs, such as *communities* which are subsets of the nodes. A graph with community is a structure $G = (V, E, U_1, ...U_k, d)$ where each $U_i \subseteq V$ represents the i-th community and d is the degree function, as in Figure 4. In many models, the U_i are disjoint and in this case we use the structure $GC = (V, E, C, d)$, *graphs with communities*, where $C : V \to N$ is the Community function, i.e. $C(x) = i$ iff $U_i(x)$, i.e. x is in the community i. We set $C(x) = 0$ if x is in no community.

Community detection is an approximate extension, based on Mincut algorithms or other combinatorial methods. The extended structure is therefore only approximate as many nodes may be incorrectly labeled. The analytical queries will therefore be only approximate. The main difficulty is to maintain coherent communities over long periods of time. If we analyze a new graph every day, we have to identify the communities at time t from the communities at the previous time.

V. Analytical queries on streaming data

We wish to analyze the graphs with communities, defined from streaming Json data, as in the previous sections. Typical analytical queries on these structures are:

- A_1: The most influential users, i.e. the nodes of maximal degrees, in each of the communities.
- A_2: The distribution of sizes of the communities.

In this section we first define a class of analytical queries on graphs and study their approximation. We show that the first query A_1 can be approximated with the degree distribution, whereas the second query A_2 can be approximated with the uniform distribution. On the other hand, the uniform distribution can't approximate A_1 and the degree distribution can't approximate A_2.

A. Analytical queries on graphs

We first need to formalize the notion of an analytical query on *graphs with communities*, i.e. structure $GC = (V, E, C, d)$.

Definition 2: An analytical query on graphs with communities GC has three components:

- A first-order formula $Q(x_1, ...x_k)$ in the language of GC, i.e. with relations $=$, E the edge relation and the graphs of the functions C and d, i.e. the atomic formulas $C(x) = y$ and $d(x) = y$,
- Classifiers, i.e. free variables of Q,
- Aggregation operators: Max, Min, Sum, Count,... applied to some of the free variables of Q.

Let us say that the variables used as Classifiers or by the Aggregation are *bound*. The free variables are the variables of Q which are not bound.

In a typical example $Q(x, y, z)$ is a first-order formula in the language of GC. We may classify on z and apply on operator on y. In this case the *answer to the query* may be a set of x's for each value of z or a set of values y for each value of z. We may obtain a set or a distribution depending on which Aggregation operator we use. Consider the two previous queries.

The *most influential users per community* query A_1 is defined by the following components:

- $Q(x, y, z) : \ d(x) = y \wedge C(x) = z$,
- Classifier: z,
- Aggregation $Max_x\{y\}$.

The answer is a set $\{(u_0, 0), (u_1, 1), ...(u_i, i), ...\}$ such that $GC \models Q(u_i, j, i)$ for $u_i \in V$ and j is the maximal degree of node u_i for the community i.

The *distribution of sizes of communities* query, A_2, is defined by the components:

- $Q(x, z) : \ C(x) = z$,
- Classifier: z,
- Aggregation $Count(x)$.

The answer may be the set of pairs $\{(30, 0), (40, 1), (20, 3), (10, 4)\}$, i.e. $\{(j, i)\}$ such that $GC \models Q(u_i, i)$ and j is the number of such u_i's for a community i. If we take the relative values, we may have the distribution $\{(30\%, 0), (40\%, 1), (20\%, 2), (10\%, 3)\}$ for the 3 communities, and 30% in no community, as in Figure 5.

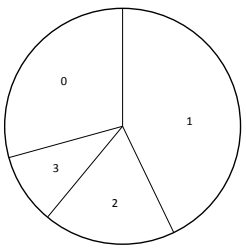

Fig. 5. Distribution of sizes of the communities (1,2,3). Label 0 is for nodes with no community.

In general, the answer to an analytical query is obtained by taking first the possible values on the Classifier z. There are two cases:

- No free variables: the Aggregation function returns for each value of the Classifier, a numerical value. We can represent the answer to the query as a distribution, by taking the relative values. The answer of A_2 is a distribution.
- Free variables: the Aggregation function returns a value which depends on x. We can represent the answer of the query as a function which for each z gives a relation $Q(x, y)$. The answer of A_1 gives for each z an individual x and its degree y.

B. Approximation of Analytical queries on graphs

For the query A_2, the result is a distribution as in Figure 5 and we approximate it with the L_1 distance, as in the case of the OLAP queries. For the query A_1, we obtain a set and the distance clearly depends on the degree function d. Suppose the exact answer is $S = \{(u_0, 0), (u_1, 1), (u_2, 2), (u_3, 3)\}$ and we output $S' = \{(u_0, 0), (u'_1, 1), (u_2, 2), (u_3, 3)\}$, i.e. a different node u'_1 for the 1st community. If the degree of u_1 is 10 and the degree of u'_1 is 8, we make an absolute error of 2 and a relative error of $2/10$.

The *relative weighted error* for A_1 is based on weighted colored sets, where the weight of each element is its degree. The relative distance between two colored sets is the sum of the weights on the differences divided by the total sum of the weights.

We say that an analytical query A is ε-approximated by an algorithm $\mathbb{A}$ if for every large enough graph G with communities with n nodes, the relative distance between $A(G)$ and $\mathbb{A}(G)$ is less than ε. We concentrate on the number of samples used by the algorithm. In the optimal case, it depends on ε only and we say that we us e $O(1)$ samples. In the general case, it depends on ε and on $f(n)$ where n is the size of the structure and f some function. We expect f to be sublinear, as $\log n$ for example. If f is the identity, then we use the whole structure, and the samples are useless.

C. General method for lower bounds

We want to show that some properties P can't be approximated from a sketch, random samples taken from a specific distribution.

Lemma 1: If there are two structures which differ on P, with have ε-close sketches, then P can't be ε-approximated using $O(1)$ samples.

Proof: Consider a graph property P and two graphs G_1 and G_2 such that $G_1 \models P$ and $G_1 \models \neg P$. As the

sketches of G_1 and G_2 are ε-close, any randomized algorithms can't distinguish them, hence the property P can't be ε-approximated using $O(1)$ samples. ■

It suffices to find two graphs which differ on P with close sketches, to conclude that the properties can't be ε-approximated with $O(1)$ samples.

D. Main results

We now present the main results concerning the two queries A_1 and A_2.

Theorem 1: The analytical query A_1 can be ε-approximated with $O(1/\varepsilon^2)$ samples with the degree distribution but requires $O(f(n))$ samples with the uniform distribution.

Samples with the degree distribution will most likely take nodes of high degrees, hence make a small error for the relative weighted error. Uniform samples can be as far as possible from the points of high degree. Consider two classical graphs: a circle C_n, i.e. n nodes with edges linking nodes i with $i+1$ and node n with node 1, and a star S_n with node 1 linked with all the other nodes. A formal argument uses directly lemma 1 applied to these graphs.

Proof: Let G_1 be the union of two circles C_n, one for each community and G_2 be the union of two stars S_n, one for each community. If we sample according to the degree, we will find a corrext approximate answer with few samples (less than $O(1/\varepsilon^2)$). If we sample with the uniform distribution, the answer on G_2 will be incorrect with high probability. The distribution of the samples on G_1 and G_2 will be close and hence by lemma 1, A_1 can't be approximated with $O(1)$ samples. ■

Theorem 2: The analytical query A_2 can be ε-approximated with $O(1/\varepsilon^2)$ samples with the uniform distribution but requires $O(f(n))$ samples with the degree distribution.

On the contrary, uniform samples approximate the size of the communities, using a simple Chernoff bound. Samples with the degree distribution will not correctly estimate the sizes of the communities, using an argument similar to the one used in theorem 1.

E. The value of analytical queries

If the answer of the analytical query is a distribution, we can use the relativized entropy $Ent(S_\nu)$ for some values of the classifiers, i.e.

$$GH(S|A) = \sum_{\nu \in A} \frac{|S_\nu|}{|S|}.[Ent(S_\nu))]$$

Let the Gain of Information for the dimension A be the function

$$Gain(S|A) = \frac{1}{1 + GH(S|A)}$$

so that this new function varies between 0 and 1. The optimal value (maximal gain) is 1 when the relative Entropy is 0, i.e. a Dirac distribution.

In our example, the answer of A_2 is a distribution. We can compute $Gain(A_2) = Gain(Q|A)$ with A as a community attribute, and compare it to the gain of another query. If the case of A_1, whose answer is a set, the model does not apply. It needs to be extended.

VI. Experiments

We conducted two experiments, one for Facebook and one for Twitter. The first analysis follows the classical OLAP queries whereas the second one follows the model of graphs with communities.

A. Facebook

With the *Nodexl* package and its Social Network Importer, we selected the Facebook fan page of the Paddington movie for the French market (*https://www.facebook.com/PaddingtonFrance*) and a specific time window. Nodexl transmits the Json data which can be analyzed by the *Gephi* package as a graph and by the *Qlik* package for the OLAP analysis.

The OLAP schema is given in Figure 1. The first analysis uses both the Time and Type dimensions, given in Figure 7 as different curves.

We want to correlate these data, with the ticket sales on the same time interval.

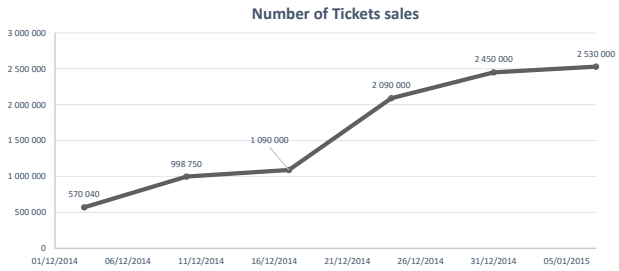

Fig. 6. Total number of tickets, hence revenues on the same time interval.

Figure 6 gives the projected number of tickets, hence the revenues. The linear model can be built with these data and will provide a better prediction tool, adapted to the different dimensions.

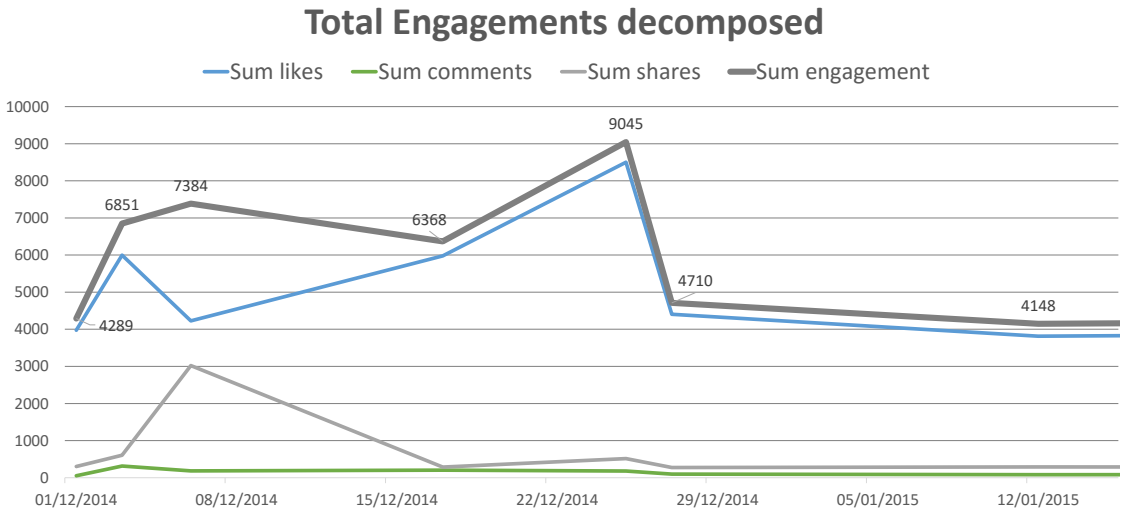

Fig. 7. Total Engagements, decomposed as likes (blue line), comments (green line), shares (grey line) and total (dark grey line), in Facebook.

The main observation is the increase of *shares* and the decrease of *likes* at $1/3$ of the time scale.

Figure 8 gives the analysis per Gender wheras Figure 9 gives the analysis per Country. The modelisation of section 3 using the $Gain$ function concludes that $Gain(S|Gender) < Gain(S|Country)$, i.e. the analysis per country gives *more information* than the analysis per Gender.

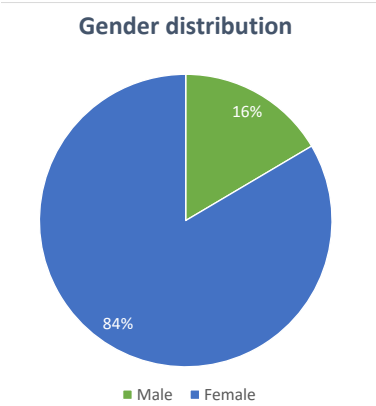

Fig. 8. Analysis per Gender (male, female), in Facebook.

B. Twitter

We analyzed the Json stream with the *Naoyun* connector on keywords such #LVMH (luxury brand) and used *Gephi* to obtain a graph with communities such as the one described in Figure 4. The communities partition the users in different segments and a challenging task is to follow these partitions in time.

A normalized query language is needed to analyze this structure and to implement the correct sampling strategies.

VII. Conclusion

We considered two models of Social Networks, inspired by Facebook and Twitter and proposed a model of value for a class of analytical queries. In this work, we don't take the visualization paradigm into account, in order to concentrate on the value of distributions.

The first model is a classical datawarehouse with an OLAP schema. We used the linear influence model, and analyzed the influence vector of users for each value of the dimensions. We introduced an Entropy based method to quantify the value of these OLAP analyses, combining the Entropy of the answer distribution, with the Entropy of the distribution of the Influence vectors.

Our main contribution is a framework which extends this approach to streaming data. In Twitter, the Json trees are transformed into graphs and then extended into graphs with communities. We extend the classical ETL (Extract Transform Load) step with a *structural compression* used to efficiently approximate analytical queries. We defined a class of analytical queries for these graphs and gave examples of approximable and non-approximable queries. When the answer of the analytical query is a distribution, the Entropy-based model can be applied.

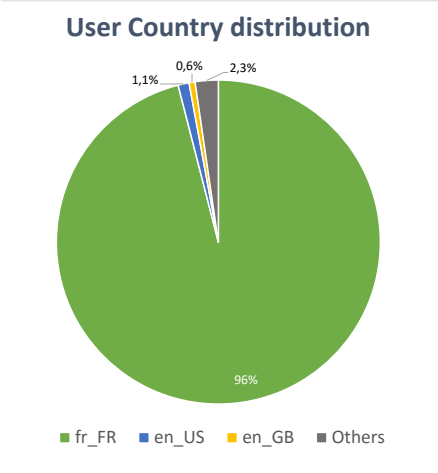

Fig. 9. Analysis per Country, in Facebook.

REFERENCES

[1] S. Asur and B. A. Huberman. Predicting the future with social media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '10, pages 492–499. IEEE Computer Society, 2010.

[2] Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[3] M. de Rougemont. Statsreduce in the cloud for approximate analytics. In *International Conference on Data Science and Advanced Analytics, DSAA*, pages 593–599, 2014.

[4] J. Feigenbaum, S. Kannan, M. Strauss, and M. Viswanathan. Testing and spot-checking of data streams. *Algorithmica*, 34(1):67–80, 2002.

[5] E. Fischer, F. Magniez, and M. de Rougemont. Approximate Satisfiability and Equivalence. *SIAM Journal of Computing*, 39(6):421–430, 2010.

[6] N. François, F. Magniez, M. de Rougemont, and O. Serre. Streaming property testing of visibly pushdown languages. *CoRR*, http://arxiv.org/abs/1505.03334, 2015.

[7] Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 211–220. ACM, 2009.

[8] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

[9] Manuel Gomez Rodriguez, Jure Leskovec, and Bernhard Schölkopf. Structure and dynamics of information pathways in online media. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 23–32. ACM, 2013.

[10] Wan-Hsin Tang, Mi-Yen Yeh, and Anthony J. T. Lee. Information diffusion among users on facebook fan pages over time: Its impact on movie box office. In *International Conference on Data Science and Advanced Analytics, DSAA 2014*, pages 340–346, 2014.

[11] Jeffrey S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, March 1985.

[12] Jaewon Yang and Jure Leskovec. Modeling information diffusion in implicit networks. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 599–608. IEEE Computer Society, 2010.