An Incremental Algorithm for Granular Counting with Possibility Theory

Corrado Mencar
Department of Informatics, University of Bari "Aldo Moro", 70125 Bari, Italy
Email: corrado.mencar@uniba.it

***Abstract*—Data counting is non-trivial when data are uncertain. In the case of uncertainty due to incompleteness, possibility theory can be used to define a granular counting model. Two algorithms were proposed in literature to compute granular counting: exact granular counting, with quadratic time complexity, and approximate granular counting, with linear time complexity. However, both algorithms require that all data are available before counting. This paper presents an incremental granular counting algorithm which provides an efficient and exact computation of the granular count without the need of having all data available, thus opening the door to applications involving data streams.**

***Index Terms*—Possibility Theory, Fuzzy Arithmetic, Uncertain Data**

I. Introduction

Data counting is the process of finding the number of data samples having a specific value, hence it is often a preliminary step for several types of analysis, such as descriptive statistics, comparative analysis, etc. It is quite a simple operation when data are precise, but it becomes non-trivial when data are uncertain. In fact, uncertainty in data should propagate in counting, so that results are *granular* rather than precise.

There are at least four strategies to deal with uncertain data: understand, minimize, exploit and ignore uncertainty [1]. Most often than not, uncertainty is ignored in order to simplify the subsequent processes, by assigning a value to an uncertain data sample according to some arbitrary criteria. However ignoring uncertainty may introduce bias in the subsequent processing stages, which is hard to recognize. On the other hand, uncertainty can be exploited by propagating it through information processing: in this way, the results of data analysis show their uncertainty, which can be assessed in order to judge their final utility. But, in order to be exploited, uncertainty must be understood and modeled by a proper theoretical framework.

In fact, several frameworks are available to deal with uncertainty, first of all probabilistic [2], which is however a particular case falling in the Granular Computing paradigm that also includes classical sets [3], rough sets [4], evidence theory [5] and possibility theory [6]. In particular, possibility theory deals with uncertainty due to incomplete information, e.g. when the value of an observation cannot be precisely determined: in this case we speak of *uncertain data*. (This approach contrasts with a probabilistic framework, where precise data are collected from a random phenomenon.) We adopt the possibilistic framework in this paper.

Mencar & Pedrycz proposed a definition of granular count through possibility theory [7]. It was shown that the resulting counts are fuzzy intervals in the domain of natural numbers. Based on this result, two algorithms for granular counting were defined: an exact granular counting algorithm with quadratic-time complexity and an approximate counting algorithm with linear-time complexity. Approximate granular counting is appealing in applications dealing with large amounts of data due to its low complexity, but a compromise must be accepted in terms of accuracy of the resulting fuzzy interval. Furthermore, both algorithms require all data to be available before counting, which is ineffective in data stream applications where data (usually in large amounts) are collected progressively.

To overcome such limitation, a different approach is undertaken in this paper to count data, which is based on fuzzy arithmetic. The result is a new algorithm which carries out exact counting but more efficiently than the original exact counting algorithm. Most importantly, the new algorithm is capable of computing granular count in an incremental fashion, thus opening the door to applications involving data streams.

The concept of granular count and related algorithms are briefly described in Sec. II, while the proposal of incremental granular count is introduced in Sec. III. Sec. IV reports some numerical experiments to assess the efficiency of the proposed algorithm, as well as the outline of an application in Bioinformatics.

II. Granular count

A brief summary of Granular Counting is reported in this Section. Further details can be found in the original papers [7], [8].

We assume that data are manifested through *observations*, which refer to some objects or *referents*. The relation between observations and referents—which is called *reference*—is binary (an observation either refers or not to a reference) but may be uncertain in the sense that an unequivocal reference of the observation to one of the referents is not possible. For example, a RNA fragment can be considered as an observation that may refer to a gene; the fragment has been generated by one gene only but its observation may not allow to identify the gene with certainty. We model such uncertainty with possibility theory [6] as we assume that uncertainty is due to the imprecision of the observation, i.e. the observation is not complete enough to make reference unequivocal.

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

Given a set R of referents and an observation o in a set $O_{(m)}$ of m observations,[1] a possibility distribution is a mapping

$$\pi_o : R \mapsto [0, 1]$$

such that $\exists r \in R : \pi_o(r) = 1$. The value $\pi_o(r) = 0$ means that it is impossible that the referent r is referred by the observation, while $\pi_o(r) = 1$ means that the referent r is absolutely possible (though not certain). Intermediate values of $\pi_o(r)$ stand for gradual values of possibility, which are related to the completeness of information resulting from an observation. The possibility distributions of all observations can be arranged in a *possibilistic assignment table*, as exemplified in Table I. In the each row of the table, the possibility distribution that an observation $o \in O_{(m)}$ refers to each considered reference $r \in R$ is reported.

TABLE I
EXAMPLE OF POSSIBILISTIC ASSIGNMENT TABLE. EACH ROW IS A POSSIBILITY DISTRIBUTION π_{o_j}.

	r_1	r_2	r_3
o_1	1	0.3	0.54
o_2	0.8	1	0.6
o_3	1	0	0
o_4	0.864	0.91	1
o_5	1	0	0
o_6	0.5	1	0.64
o_7	1	0.8	1
o_8	0.2	0.5	1
o_9	1	0	0
o_{10}	0.6	1	0.78

A. Definition of granular count

By using the operators of possibility theory, as well as the assumption that observations are non-interactive (i.e. they do not influence each other), the possibility degree that a subset $O_x \subseteq O_{(m)}$ of $x \in \mathbb{N}$ observations is *exactly*[2] the set of observations referring to an reference $r_i \in R$, is defined as:

$$\pi_{O_x}(r_i) = \min \left\{ \min_{o \in O_x} \pi_o(r_i), \min_{o \notin O_x} \max_{r \neq r_i} \pi_o(r) \right\} \quad (1)$$

with the convention that $\min \emptyset = 1$. Informally speaking, Eq. (1) defines the possibility degree that O_x is the subset of all and only the observations referring to r_i by computing the least possibility degree of two simultaneous events: (i) all observations of O_x refer to r_i, and (ii) all the other observations refer to a different referent.

In order to compute the possibility degree that the number of observations referring to a referent r_i is $N_{(m)}$, we are not interested in a specific set O_x, but in *any set* of x elements. We can therefore define the possibility value that the number of observations for a referent r_i is x as:

$$N_{(m)}(x) = \max_{O_x \subseteq O_{(m)}} \pi_{O_x}(r_i) \quad (2)$$

[1]Since in our analysis we will deal with a varying number of observations, their number m is highlighted in the notation.

[2]In the sense that any observation non belonging to O_x does not refer to r.

for $x \leq m$ and $N_{(m)}(x) = 0$ for $x > m$. Eq. (2) provides a granular definition of count. Counting is imprecise because observations are uncertain.

It is possible to prove that a granular count as in Eq. (2) is a fuzzy interval in the domain of natural numbers. A fuzzy interval is a convex and normal fuzzy set on a numerical domain (in our case, it is the set of natural numbers, $\mathbb{N}$). Convexity of a fuzzy set can be established by proving that all α-cuts are intervals, while normality of the granular count is guaranteed because of the normality of the possibility distributions π_o for all $o \in O_{(m)}$. Fig. 1 depicts an example of possibility distribution representing the granular count of referent r_1 in Table I.

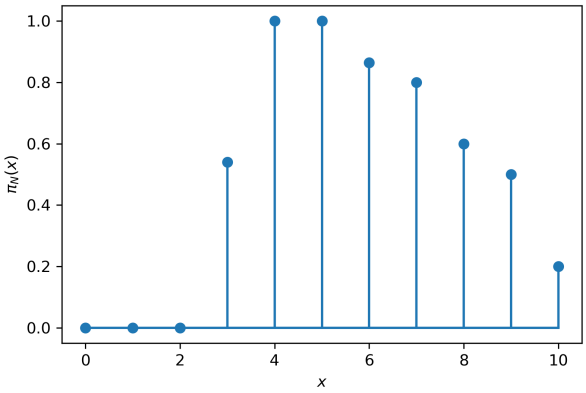

Fig. 1. Exact granular count of referent r_1 as in Table I

B. Algorithms for granular counting

The direct application of Eq. (2) leads to an intractable counting procedure as all possible subsets of $O_{(m)}$ must be considered. On the other hand, a polynomial-time algorithm can be devised by making profit of the representation of a granular count as a fuzzy interval. In particular, a granular counting algorithm builds the fuzzy interval by considering the α-cut representation of fuzzy sets. On such basis, two variants of granular counting algorithms can be devised:

- *Exact* granular counting uses all the values of α that correspond to some possibility degree in the possibilistic assignment table;
- *Approximate* granular counting uses the values of α taken from a finite set of evenly spaced numbers over $]0, 1]$. The number of such values depend on a user-defined parameter n_α.

The approximate granular counting is more efficient than the exact version because it does not require to scan the possibilistic assignment table, though at the price of a new required parameter.

Exact granular counting (Algorithm 1) and approximate granular counting (Algorithm 2) share the same core algorithm (Algorithm 3) and only differ by how the set of α-values are computed. In essence, the core algorithm computes the

Algorithm 1: EXACTGRANULARCOUNT

Data: $\mathbf{R}, i$
/* $\mathbf{R}$: possibil. assignment table */
/* i: index of referent to count */
Result: $N \in [0,1]^m$
1 $A \leftarrow \{\alpha \in \mathbf{R} : \alpha \neq 0\}$;
2 **return** GRANULARCOUNT(R, i, A);

Algorithm 2: APPROXIMATEGRANULARCOUNT

Data: $\mathbf{R}, i, n_\alpha$
/* $\mathbf{R}$: possibil. assignment table */
/* i: index of referent to count */
/* n_α: number of α-levels */
Result: $N \in [0,1]^m$
1 $\varepsilon \leftarrow 10^{-12}$;
2 $A \leftarrow \{\varepsilon + k \cdot \frac{1-\varepsilon}{n_\alpha - 1} : k = 0, 1, \ldots, n_\alpha - 1\}$;
3 **return** GRANULARCOUNT$(\mathbf{R}, i, A)$;

granular count by reckoning the α-cuts of the fuzzy interval for each α value provided in input.

In brief, the core algorithm works as follows. Given the possibilistic assignment table $\mathbf{R}$, the index i of the referent and the set A of α-cuts, the array $\mathbf{r}$ represents the possibility degrees that an observation refers to r_i, i.e. $\mathbf{r}_j = \pi_{o_j}(r_i)$ (line 1), while $\bar{\mathbf{r}}$ represents the possibility degrees that an observation refers to any other referent different from r_i (line 2). N is the array representing the granular count (line 3). The main cycle (lines 4–17) loops over each $\alpha \in A$ and computes the bounds $\underline{x}$ and $\bar{x}$ of the corresponding α-cut (line 5). These bounds are calculated by looping over all observations (lines 6–13), so that $\bar{x}$ is incremented if the possibility degree that the current observation refers to r_i is greater than or equal to α (lines 7–8), while $\underline{x}$ further requires that the possibility degree that the observation refers to any other referent is less than α (lines 9–10). When both $\underline{x}$ and $\bar{x}$ are computed, the degrees of membership of the granular count are updated accordingly (lines 14–16).

For a fixed referent, the time-complexity of exact granular count is $\mathcal{O}\left(nm^2\right)$ (being n the number of referents and m the number of observations), while the time-complexity of approximate granular count drops to $\mathcal{O}\left(m\left(n + n_\alpha\right)\right)$. In consideration that, in typical scenarios, the number of observations is very large (i.e., $m \gg n$), especially in comparison with the number of referents, it is deduced that approximate granular count is the preferred choice in the case of very large amounts of uncertain data.

III. INCREMENTAL GRANULAR COUNT

A. α-cut computation

Algorithm 3 computes the granular count for the i-th referent given a possibilistic assignment table $\mathbf{R}$ and a set A of α-values. The main cycle within the algorithm computes the α-cut of the granular count, which is represented by the

Algorithm 3: GRANULARCOUNT

Data: $\mathbf{R}, i, A$
Result: $N \in [0,1]^m$
/* m is the number of observations */
1 $\mathbf{r} \leftarrow [\mathbf{R}_{ji}]$ for $j = 1, 2, \ldots, m$;
2 $\bar{\mathbf{r}} \leftarrow [\max_{k \neq i} \mathbf{R}_{jk}]$ for $j = 1, 2, \ldots, m$;
3 $N \leftarrow [0, 0, \ldots, 0]$ ($m+1$ times);
4 **for** $\alpha \in A$ **do**
5 $\quad \underline{x} \leftarrow 0$; $\overline{x} \leftarrow 0$;
$\quad$ /* Compute α-cut */
6 $\quad$ **for** $k = 1, 2, \ldots, m$ **do**
7 $\quad\quad$ **if** $\mathbf{R}_{ki} \geq \alpha$ **then**
8 $\quad\quad\quad \overline{x} \leftarrow \overline{x} + 1$;
9 $\quad\quad\quad$ **if** $\bar{\mathbf{r}}_{ki} < \alpha$ **then**
10 $\quad\quad\quad\quad \underline{x} \leftarrow \underline{x} + 1$;
11 $\quad\quad\quad$ **end**
12 $\quad\quad$ **end**
13 $\quad$ **end**
$\quad$ /* Update granular count */
14 $\quad$ **for** $x \in \underline{x}, \ldots, \overline{x}$ **do**
15 $\quad\quad N[x] \leftarrow \max\{N[x], \alpha\}$;
16 $\quad$ **end**
17 **end**
18 **return** N

array N and corresponds to the possibility distribution $N_{(m)}$ defined in (2). For a given value of α, the variable $\overline{x}$ counts the number of observations that refer to r_i with a possibility degree $\geq \alpha$; on the other hand, the variable $\underline{x}$ counts the number of observations that refer to r_i with a possibility degree $\geq \alpha$ *and* refer to any other referent with possibility degree $< \alpha$. As a consequence, $\underline{x} \leq \overline{x}$. For the sake of our analysis, since we will consider different values of m, we shall denote the two variables as $\underline{x}_{(m)}$ and $\overline{x}_{(m)}$ respectively.

By construction, the value $\overline{x}_{(m)}$ corresponds to the cardinality of the set

$$\overline{O}_{(m)} = \left\{o \in O_{(m)} | \pi_o(r_i) \geq \alpha\right\} \tag{3}$$

while the value $\underline{x}_{(m)}$ is the cardinality of the set

$$\underline{O}_{(m)} = \left\{o \in O_{(m)} | \pi_o(r_i) \geq \alpha \wedge \max_{r \neq r_i} \pi_o(r) < \alpha\right\} \tag{4}$$

with the obvious relation that $\underline{O}_{(m)} \subseteq \overline{O}_{(m)}$. In summary, the α-cut of the granular count $N_{(m)}$ is

$$\left[N_{(m)}\right]_\alpha = \left[\underline{x}_{(m)}, \overline{x}_{(m)}\right]$$

B. Fuzzy increment

For an observation $o_j \in O_{(m)}$, we define a *fuzzy increment* as the discrete fuzzy set over $\{0,1\}$ such that[3]

$$I_j = \left\{ \frac{\beta_0}{0} + \frac{\beta_1}{1} \right\} = \left\{ \frac{\max_{r \neq r_i} \pi_{o_j}(r)}{0} + \frac{\pi_{o_j}(r_i)}{1} \right\} \quad (5)$$

A fuzzy increment represents the possibility that a count is incremented by 0 (i.e. *not* incremented) or 1 when observation o_j is considered. The possibility that the increment is 0 is given by the possibility that the observation does not refer to the referent under consideration, while the increment is non-null if it is possible that the observation refer to the referent. Because of the normality of the possibility distribution, $\max\{\beta_0, \beta_1\} = 1$; therefore, I_j is a fuzzy number.

Intuitively, a granular count can be identified with the accumulation of fuzzy increments for all observations. Indeed, the intuition is true as proved by the following theorem.

Theorem 1. *For any assignment table of m observations:*

$$N_{(m)} = \sum_{j=1}^{m} I_j$$

Proof: We prove the theorem by induction on m.

a) Base case $m = 1$: In this case, the set of observations is a singleton, i.e. $O_{(1)} = \{o_1\}$, therefore, by definition:

$$N_{(1)}(0) = \pi_{\emptyset}(r_i) = \max_{r \neq r_i} \pi_{o_1}(r) = \beta_0$$

and

$$N_{(1)}(1) = \pi_{O_{(1)}}(r_i) = \pi_{o_1}(r_i) = \beta_1$$

therefore, $N_{(1)} = I_1$.

b) Case $m > 1$: By inductive hypothesis, if we assume the theorem valid for m, we must show that:

$$N_{(m+1)} = \sum_{j=1}^{m+1} I_j$$

that is, we must prove the relation

$$N_{(m+1)} = N_{(m)} + I_{m+1}$$

To this end, we will prove that every α-cut of $N_{(m+1)}$ coincides with the α-cut of $N_{(m)} + I_{m+1}$.

Let $\alpha \in]0,1]$. By the extension principle, the fuzzy set $N_{(m)} + I_{m+1}$ has the following membership function:

$$\left[N_{(m)} + I_{m+1}\right](x) = \max_{a,b:a+b=x} \min\left\{N_{(m)}(a), I_{m+1}(b)\right\} \quad (6)$$

Since $b \in \{0,1\}$, eq. (6) can be simplified as:

$$\left[N_{(m)} + I_{m+1}\right](x) = \max\{ \min\left\{N_{(m)}(x), \beta_0\right\}, \min\left\{N_{(m)}(x-1), \beta_1\right\}\} \quad (7)$$

[3]Zadeh's notation for fuzzy sets is used, i.e. a fuzzy set A defined on the universe of discourse $U = \{x_1, \ldots, x_n\}$ is represented as $A = \left\{ \frac{\mu_A(x_1)}{x_1} + \cdots + \frac{\mu_A(x_n)}{x_n} \right\}$. In particular, fuzzy set I_j is defined on $\{0,1\}$.

where

$$I_{m+1} = \left\{ \frac{\beta_0}{0} + \frac{\beta_1}{1} \right\}$$

defined as in (5).

Let $x \in \mathbb{N}$. Then

$$x \in \left[N_{(m)} + I_{m+1}\right]_\alpha \Leftrightarrow \min\left\{N_{(m)}(x), \beta_0\right\} \geq \alpha \text{ or } \min\left\{N_{(m)}(x-1), \beta_1\right\} \geq \alpha$$

or, equivalently, $x \in \left[N_{(m)} + I_{m+1}\right]_\alpha$ if and only if:

$$x \in \left[N_{(m)}\right]_\alpha \wedge \alpha \leq \beta_0 \quad (8)$$

$$\vee$$

$$x - 1 \in \left[N_{(m)}\right]_\alpha \wedge \alpha \leq \beta_1 \quad (9)$$

where $\left[N_{(m)}\right]_\alpha = \left[\underline{x}_{(m)}, \overline{x}_{(m)}\right]$. Three cases can be considered.

- $\alpha \leq \beta_1$ and $\alpha > \beta_0$: in this case, only condition (9) applies, therefore:

$$\left[N_{(m)} + I_{m+1}\right]_\alpha = \left[\underline{x}_{(m)} + 1, \overline{x}_{(m)} + 1\right] \quad (10)$$

  i.e., the α cut of the sum $N_{(m)} + I_{m+1}$ coincides with the α-cut of $N_{(m)}$ shifted by 1.
- $\alpha \leq \beta_1$ and $\alpha \leq \beta_0$: in this case both conditions (8) and (9) apply, therefore:

$$\left[N_{(m)} + I_{m+1}\right]_\alpha = \left[\underline{x}_{(m)} + 1, \overline{x}_{(m)} + 1\right] \cup \left[\underline{x}_{(m)}, \overline{x}_{(m)}\right] = \left[\underline{x}_{(m)}, \overline{x}_{(m)} + 1\right] \quad (11)$$

  i.e., the α cut of the sum $N_{(m)} + I_{m+1}$ expands the α-cut of $N_{(m)}$ by 1 on the right.
- $\alpha > \beta_1$ and $\alpha \leq \beta_0$: in this case, only condition (8) applies, therefore:

$$\left[N_{(m)} + I_{m+1}\right]_\alpha = \left[\underline{x}_{(m)}, \overline{x}_{(m)}\right] \quad (12)$$

  i.e. the α cut of the sum $N_{(m)} + I_{m+1}$ coincides exactly with the α-cut of $N_{(m)}$.

Notice that the fourth case, i.e. $\alpha > \beta_1$ and $\alpha > \beta_0$ is impossible since we would have $\alpha > \max\{\beta_0, \beta_1\} = 1$.

We now turn our attention to $N_{(m+1)}$. By definition, each α-cut is an interval

$$\left[N_{(m+1)}\right]_\alpha = \left[\underline{x}_{(m+1)}, \overline{x}_{(m+1)}\right]$$

where $\underline{x}_{(m+1)}$ and and $\overline{x}_{(m+1)}$ are the cardinalities of sets $\underline{O}_{(m+1)}$ and $\overline{O}_{(m+1)}$ as defined in (3) and (4). Both sets are subsets of $O_{(m+1)} = O_{(m)} \cup \{o_{m+1}\}$, being $o_{m+1} \notin O_{(m)}$ where $\pi_{o_{m+1}}(r_i) = \beta_0$ and $\max_{r \neq r_i} \pi_{o_{m+1}}(r) = \beta_1$.

Again, three cases can be analyzed:

- $\alpha \leq \beta_1$ and $\alpha > \beta_0$: in this case, by definition both

$$\underline{O}_{(m+1)} = \underline{O}_{(m)} \cup \{o_{m+1}\}$$

and

$$\overline{O}_{(m+1)} = \overline{O}_{(m)} \cup \{o_{m+1}\}$$

therefore $\underline{x}_{(m+1)} = \underline{x}_{(m)} + 1$ and $\overline{x}_{(m+1)} = \overline{x}_{(m)} + 1$. The α-cut of $N_{(m+1)}$ coincides with the α-cut of $N_{(m)} + I_{m+1}$ as in (10).

- $\alpha \leq \beta_1$ and $\alpha \leq \beta_0$: in this case, $o_{m+1} \notin \underline{O}_{(m+1)}$ but $o_{m+1} \in \overline{O}_{(m+1)}$, therefore $\underline{x}_{(m+1)} = \underline{x}_{(m)}$ and $\overline{x}_{(m+1)} = \overline{x}_{(m)} + 1$. In this case too, the α-cut of $N_{(m+1)}$ coincides with the α-cut of $N_{(m)} + I_{m+1}$ as in (11).
- $\alpha > \beta_1$ and $\alpha \leq \beta_0$: in this case, o_{m+1} does not belong neither to $\underline{O}_{(m+1)}$ nor to $\overline{O}_{(m+1)}$, therefore $\underline{x}_{(m+1)} = \underline{x}_{(m)}$ and $\overline{x}_{(m+1)} = \overline{x}_{(m)}$. The α-cut of $N_{(m+1)}$ coincides with the α-cut of $N_{(m)} + I_{m+1}$ as in (12).

In all cases, for all $\alpha \in]0, 1]$: $\left[N_{(m)} + I_{m+1}\right]_\alpha = \left[N_{(m+1)}\right]_\alpha$, therefore $N_{(m)} + I_{m+1} = N_{(m+1)}$. ■

C. Algorithm

Theorem 1 suggests a new algorithm for incremental granular count, which is reported in Algorithm 4. Some observations on the algorithm are noteworthy:

1) The main cycle of the algorithm (lines 4–9) loops over the index j, which scans all the observations once; also, the last statement in the loop (line 8) updates the granular count represented by the array N. In this way partial counts are always available. This makes the algorithm applicable in data stream contexts, where data are only progressively available during processing and can be processed only once.
2) The granular count is represented by an array N whose value in index k represents the membership degree of the fuzzy interval for the number $k - 1$. Initially, N represents the number 0 by a singleton array with membership 1 in correspondence of index 1 (line 3). (It is assumed that the degree of membership for numbers not corresponding to any index is 0.)
3) The inner cycle updates the granular count according to the fuzzy sum (7). The cycle runs on the length of N which increases by one on each cycle. Notice that more efficient implementation can avoid to increase the length of N if no updates are necessary.

The time complexity of INCREMENTALGRANULARCOUNT is $\mathcal{O}\left(m^2 + mn\right)$, being the first addend determined by the nested cycles and the second addend determined by the computation of $\bar{\mathbf{r}}$ (line 2). Complexity is therefore lower than exact granular count, though still super-linear on the number of observations. In practice, it should be observed that the update of the granular count is performed about $m^2/2$ times, because the size of the array N dynamically increases;[4] also, a more efficient implementation could avoid to increase N when not necessary. In summary, incremental granular count is more efficient than exact granular count although being functionally equivalent. Approximate granular count is more efficient (being it linear on m) but it is not functionally equivalent to exact granular count.

[4]However, this requires some extra memory management.

Algorithm 4: INCREMENTALGRANULARCOUNT

```
Data: R, i
Result: N ∈ [0,1]^m
/* m is the number of observations */
1  r ← [R_ji] for j = 1, 2, ..., m;
2  r̄ ← [max_{k≠i} R_jk] for j = 1, 2, ..., m;
3  N ← [1];
4  for j ∈ 1, 2, ..., m do
5      for k ∈ 1, 2, ..., j + 1 do
6          N'_k ← max{min{N_k, r̄_j}, min{N_{k−1}, r_j}};
           /* It is assumed N_0 = 0                    */
7      end
8      N ← N';
9  end
10 return N
```

IV. EXPERIMENTAL RESULTS

A. Efficiency evaluation

The evaluation of efficiency has been performed on synthetically generated data. In particular, a number of random possibilistic assignment tables have been generated by varying the number of observations on a geometrical progression with common ratio 10 but keeping the number of referents fixed to three.[5]

For each possibilistic assignment table, both exact and incremental granular counting algorithms have been applied on the first referent, and the time required to complete operations has been recorded.[6] Each experiment has been repeated 7 times and average time has been recorded. For each repetition, the experiment has been looped for 10 times and the best timing has been retained. The average execution time is reported in Table II and depicted in Fig. 2.

TABLE II
AVERAGE EXECUTION TIME ON SYNTHETIC DATA (TIME IN SECS.)

m	exact g.c.	incremental g.c.
10	206 µs ± 3.26 µs	176 µs ± 6.51 µs
100	12.5 ms ± 995 µs	1.67 ms ± 57 µs
1,000	1.14 s ± 27.6 ms	18.7 ms ± 355 µs
10,000	1min 55s ± 639 ms	364 ms ± 13.9 ms
100,000	3h 22min	18.8 s ± 301 ms

The gain in efficiency of incremental granular count compared to exact granular count is impressive: incremental granular count is still practically feasible even when the number of observations scales to 100k. The linear regression

[5]Each possibilistc assignment table has been generated by taking care that each row corresponds to a normal possibility distribution.

[6]Experiments have been executed on a machine equipped by an Intel i7 CPU, 16GiB RAM, Linux SO. Scripts were written in Python 3.7 and executed in Jupyter Notebook. The NumPy library has been used for fast numerical computations, but the scripts were not implemented with the objective of maximizing performance.

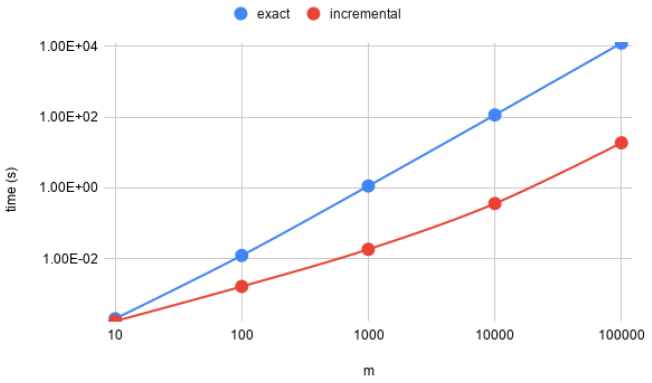

Fig. 2. Average execution time on synthetic data

in the log-log domain shows that the time required for exact granular counting grows exactly with m^2, while the growth for incremental granular counting is $\approx m^{1.24}$. The latter value summarizes the efficiency that is achieved by the implementation of incremental granular count, which is more apparent for $m < 10k$; however, for larger numbers of obervations, the quadratic time-complexity of the algorithm is expected to emerge. This phenomenon can be observed in the chart, where the slope of the line corresponding to incremental granular count slightly changes for high values of m, with a tendency of becoming parallel to the line corresponding to exact granular count.

B. Application: gene expression estimation

In Bioinformatics, RNA-Seq is a protocol that allows to examine the gene expression in a cell by sampling fragments of RNA called "reads". When RNA-Seq output is mapped against a reference database of known genes, a high percentage of reads—called *multireads*—map to more than one gene [9]. Multireads are a source of uncertainty in the quantification of gene expression, which should be managed in order to provide significant results. To this end, the mapping procedure provides a quality index that is a biologically plausible estimate of the possibility that a read can be associated to a gene [10]. However, a high quality index does not mean certainty in association: two or more genes can be candidate for mapping a read because they can be mapped with similar high quality.

Granular counting finds a natural application in the specific problem of counting the number of reads that are possibly associated to a gene. (Reads are considered as observations, while genes are referents.) However, the amount of data involved in such process may be overwhelming. For example, the public dataset SRP014005 downloaded from NCBI-SRA archive[7], contains a case-control study of the Asthma disease with 55,579 reads mapped on 14,802 genes (16% are multireads). Nonetheless, accurate granular counting can be achieved by the use of the proposed algorithm. As an example, in Fig. 3 the incremental granular count has been computed for gene OTTHUMG00000189570|HELLPAR. (The time required to compute the count was below 1s.) It is noteworthy observing how imprecise is the count of this gene, which is due to a large number of multireads (with different quality levels).

[7] ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByStudy/sra/SRP/SRP014/SRP014005

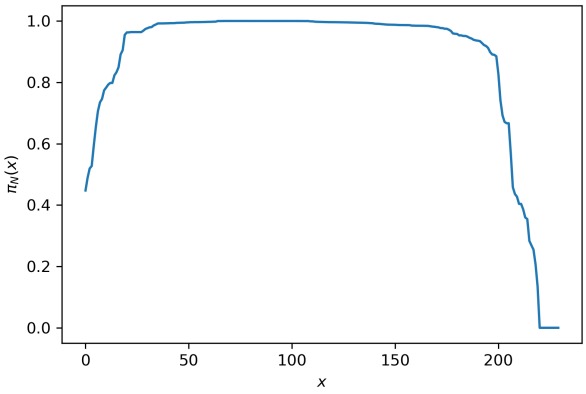

Fig. 3. Incremental granular counting of reads mapping to a sample gene

Incremental evaluation of the granular count can be observed in Fig. 4. Incremental evaluation makes possible partial analyses without the need to wait that all data are available. For example, counting the first 2000 items is enough to deduce that the number of mapped reads for the gene is highly uncertain.

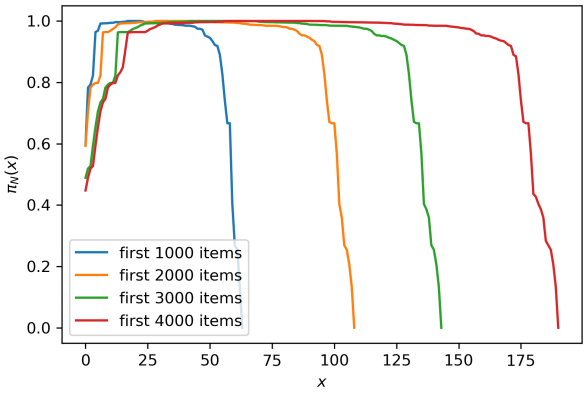

Fig. 4. Four steps for incremental granular counting of reads mapping to a sample gene (each step adds 1000 observations to the count)

V. Conclusions

The proposed incremental granular counting algorithm is a new version of exact granular counting where efficient computation is combined with the ability of managing partial sets of data, thanks to the idea of counting by summing the possibility distributions representing the reference relations. Although the computational complexity of incremental granular counting is still quadratic with respect to the number of observations, in practice the time required for computation

on large amounts of data is significantly smaller than original exact count. Most importantly, partial counts are possible with incremental granular counting, which enables to perform data analysis using granular counts even when not all data are available.

From the point of view of memory requirements, it must be noticed that incremental granular count is exact, therefore the cardinality of the set of all distinct membership degrees compares with the set of all distinct possibility degrees of all observations to the referent under consideration. Therefore, the number of membership degrees scales with m. For very large amounts of data, this might cause some problems in terms of memory resources. In such case, approximate granular counting is a solution as the set of membership degrees is fixed beforehand. The development of an incremental approximate granular counting is matter of current research.

ACKNOWLEDGMENTS

This work was partially funded by the INdAM - GNCS Project 2019 "Metodi per il trattamento di incertezza ed imprecisione nella rappresentazione e revisione di conoscenza". The author is with the *Istituto Nazionale di Alta Matematica "Francesco Severi"*, GNCS group.

REFERENCES

[1] N. Boukhelifa, M.-E. Perrin, S. Huron, and J. Eagan, "How Data Workers Cope with Uncertainty: A Task Characterisation Study," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 3645–3656.

[2] C. C. Aggarwal and P. S. Yu, "A Survey of Uncertain Data Algorithms and Applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 5, pp. 609–623, 2009.

[3] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, "Interval analysis," in *Applied interval analysis*. Springer, 2001, pp. 11–43.

[4] T. Y. Lin and N. Cercone, *Rough Sets and Data Mining*. Boston, MA: Springer US, 1996.

[5] R. R. Yager, "On the Dempster-Shafer framework and new combination rules," *Information sciences*, vol. 41, no. 2, pp. 93–137, 1987.

[6] D. Dubois and H. Prade, "Possibility Theory," in *Computational Complexity*. New York, NY: Springer New York, 2012, pp. 2240–2252. [Online]. Available: http://link.springer.com/10.1007/978-1-4614-1800-9_139

[7] C. Mencar and W. Pedrycz, "Granular counting of uncertain data," *Fuzzy Sets and Systems*, vol. 387, pp. 108–126, may 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0165011419302192

[8] ——, "GrCount: Counting method for uncertain data," *MethodsX*, vol. 6, pp. 2455–2459, 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2215016119302663

[9] A. Consiglio, C. Mencar, G. Grillo, and S. Liuni, "Managing NGS Differential Expression Uncertainty with Fuzzy Sets," in *Computational Intelligence Methods for Bioinformatics and Biostatistics. CIBB 2015 (Revised Selected Papers)*, ser. Lecture Notes in Bioinformatics, C. Angelini, S. Rovetta, and P. M. V. Rancoita, Eds. Naples, Italy: Springer, 2016, vol. 9874, pp. 42–53. [Online]. Available: http://link.springer.com/10.1007/978-3-319-44332-4_4

[10] A. Consiglio, C. Mencar, G. Grillo, F. Marzano, M. F. Caratozzolo, and S. Liuni, "A fuzzy method for RNA-Seq differential expression analysis in presence of multireads," *BMC Bioinformatics*, vol. 17, no. S12:345, pp. 167–182, oct 2016. [Online]. Available: http://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1195-2