Operation properties: a representation and their role in the propagation of meta-data.

J. Amiguet-Vercher
Data base Group
University of Twente
5, Drienerlolaan
Enschede, The Netherlans 7500AE
Email: j.amiguet@utwente.nl

P. Apers
Data base Group
University of Twente
5, Drienerlolaan
Enschede, The Netherlans 7500AE
Email: p.m.g.apers@utwente.nl

A. Wombacher
Data base Group
University of Twente
5, Drienerlolaan
Enschede, The Netherlans 7500AE
Email: a.wombacher@utwente.nl

Abstract—**To facilitate the sharing and re-use of data in scientific studies we propose an automated technique for annotating operation results. The annotated output has to preserve, as much as possible, the properties of the input annotations. The preservation of properties is achieved by taking into account operation properties. Property preservation is evaluated with information theory metrics.**

I. Motivating introduction

In many disciplines producing data is expensive and therefore data is used for multiple purposes. We have experienced this at various research and governmental organizations in the field of hydrology and many other areas [**3TU**]. However, reusing other peoples data and interpreting them correctly requires more information than the data itself. In Particular, the metadata or context of the source data is required to correctly re-use and interpret the data. However, the associated context, although vital, is easily lost when communicating results. Further, if a result serves as input to further calculations it is close to impossible to see the implications of the source data context in the output.

Techniques like data provenance [**Moreau:2008:PED:1330311.1330323**] have been developed to solve this problem. However, the interpretation of the provenance data requires an active involvement from the data user to understand the effect of the context on the resulting data. This is a time consuming task. Therefore, data provenance inference is only called upon when a result does not match the expectations of the scientist interpreting it.

An alternative approach is augmenting the processed data with information derived from the available context by using knowledge about the underlying data processing. This technique has been successfully applied in data quality propagation [**1547706**]. While the data quality approach was limited to a particular aspect of the context for a limited set of operations, we propose an approach which is more generic with regard to processing operations and context information with the risk of being less meaningful to the user. In particular, we propose a lightweight technique for propagating the data context regardless of its origin. The preservation of the context is measured in terms of information theoretic metrics.

A. *Areas of applicability*

Two areas where communicating the data context can be of crucial importance are when re-using data from previous studies, and as part of automated decision making.

Data curation efforts [**wallis2007know**, **SwissExperiment2010**] allow for data to be re used in several studies. In [**artioli2005defining**] an estimation of the coastal area of the river Po in Italy is done by combining data from 3 different studies. The original data would have been gathered and cleaned with different purposes in mind. The original purpose having influenced the data capture process by defining capture parameters such as locations and sampling ratios. Though these technical decisions, influencing the data, are part of the basic meta-data more subtle parameters may not be communicated.

Consider briefly that one of the studies contributing data to the river Po scenario, decided to normalise the water salinity measurement based on a seasonal average. Such data correction may interfere with an outlier detection technique based on value ranges. If the corrected data are not identified as corrected any further processing of the data may be based on erroneous data.

In another hydrology application subterranean water conductivity is used to determine if the water pumped from a nearby well is suitable for human consumption [**hessd-8-2503-2011**]. As part of the data processing an automated outlier detection and correction mechanism is implemented [**loureiro2004outlier**, **liu2004line**]. The data is used by a complex algorithm detecting if there is a rapid change in conductivity. Rapid changes in water conductivity are indicative of river water flowing too rapidly in the ground. Rendering ground water unsuitable for human consumption. Based on the results of the algorithm a decision is made to stop the pumping. How can we be sure that the decision is made on the basis of correct data, and not corrected outliers?

Both scenarios rely on data properties: i) Has the data been normalised? ii) Is the datum an outlier? to be available in subsequent data processing steps.

B. *Application example*

In the hydrology department of a large university staff members and PhD candidates are encouraged to share their data. Together with the data scientists annotate the values, that is, supply datum level meta-data. The annotations usually correlate with events occurring during the data acquisition (i.e. data was acquired during sensor maintenance, exceptionally high water discharge values as barrage doors where open, etc.) Other annotations indicate that the datum is a corrected outlier. Implying extra care needs to be taken since the datum represents a computed and not a measured value. Hence meta-data represents *data properties* which are of interest to the scientists.

Scientists then share their data both inside and outside the institution.

A part of the daily routine of most scientists is spent preparing the data for analysis. That is, computing aggregations, usually of the same time-spans, and removing outliers. Since most of the studies require similar data preparation it is decided to ask scientists to share their clean data. That is, data without outliers, and aggregated in common time frames. The most common aggregations are hourly, daily, and monthly averages. Further for the daily averages the values around midday and midnight are considered more representative, as such two weighted daily aggregations are made. Further data may consist of different data types. Water conductivity values may be a single scalar but wind speed and direction are usually stored in one same vector.

Scientist then notice that most of the pre-aggregated data lacks meta-data. Further the authors of the few annotated data sets complain of the time spent manually annotating the results. A lightweight, generic, preferably automated technique is required to allow the sharing of the meta-data.

Since the annotations are unconstrained, i.e. the scientist are free to specify them, interpreting the annotations is not possible. Hence, only the presence or absence of the annotation can be relied on.

Further care has to be taken that the annotations preserve their relationship to the data. That is, it would be unwise to annotate a daily average if only a datum representing a 2 minute time span is annotated. However on an hourly average it may make sense. This difference relies on the operation. That is, the hourly and the daily average operate on the same data with the same annotations, but only one of the results is annotated. The *operation properties*, i.e. daily versus hourly aggregation, contribute to the annotation of the result.

C. Contributions

This article proposes a simple automated lightweight mechanism for propagating annotations. That is, to annotate the output of aggregations with datum level meta-data present in the input of the operation. We deal in particular with operations where the result is a linear combination of the inputs. $f(x_1, \cdots, x_n) = \sum_{i=1}^{n} w_i x_i$. And do so whilst preserving the properties of the data, that is the annotation, and the properties of the operation. Our technique is optimal with regards of two preservation criteria, information preservation, and annotation distribution preservation.

The mechanism is evaluated by comparing the data property presence before and after the propagation, with the help of information theoretic measures based on the Jensen-Shannon divergence.

II. Related work

Previous work differs on the representation and manipulation of data and operation properties. Data properties can be classified according to the strictness of their definition and their origin: value, semantic or process. We outline here some of the related work organised in terms of the definition of data properties and provide a description of how operation properties are considered in each case.

When data properties are value dependent and strictly defined like in the case of numerical accuracy, completeness and variability. Klein [**1547706**] proposes the re-calculation of the data quality metric on the output of the operation. The data properties are, in this case, a function of the data. This precludes the operation properties from participating in the annotation of the output.

Another algorithmic definition of annotations is that of outlier [**loureiro2004outlier**] [**liu2004line**]. However detecting them in the output of an operation is not always possible. Leaving as only possibility to propagate the data property, from the input, instead of recognising it in the output values.

When dealing with collections of multimedia [**hare2005saliency**] or textual [**springerlink:10.1007/s11042-008-0249-**] documents annotated with semantic meta-data. Properties are propagated by annotating new documents based on their similarity. The propagation is assisted by an ontology organising the data properties. The data properties are hence constrained to a pre-defined set of related values and need not be value dependent. Operation properties do not play a role in this kind of system since the documents are not transformed.

When the data properties are un-restricted and/or generated by the computation process, as for provenance data [**journals/sigmod/SimmhanPG05**], no ontology can describe the relations between an unbound number of data properties. Further the properties not being data related inference from the output values is impossible. In this scenario operation properties have to be considered.

Previous work considering the operation in the propagation of annotations is restricted to a full description of the operation. Relational algebra is used as the sole way of describing the operation. For example [**bowers2006calculus**] offers a generic algebra enabling the propagation and inference of semantic meta-data over data transforms. When re-computing data quality metrics [**1635146**] the operations are also specified using relational algebra. Other systems in the biological domain [**conf/cidr/ArefEO07**] enable the derivation of data properties for relational query results. Mondrian [**mondrian2006**], another of such systems, focuses on supplying an interface to generate and populate the annotations.

Our approach ignores the meaning of the data property, and relies only on relevant properties of the operation.

In previous work by the same authors annotation propagation is described as a clustering under constraints problem in [**6274029**, **Amiguet:2010:ADS:1871902.1871904**] for both aggregating and interpolating operations. The present work focuses on the representation and manipulation of data and operation properties enabling the propagation on data reducing operations, such as aggregations.

III. Method outline

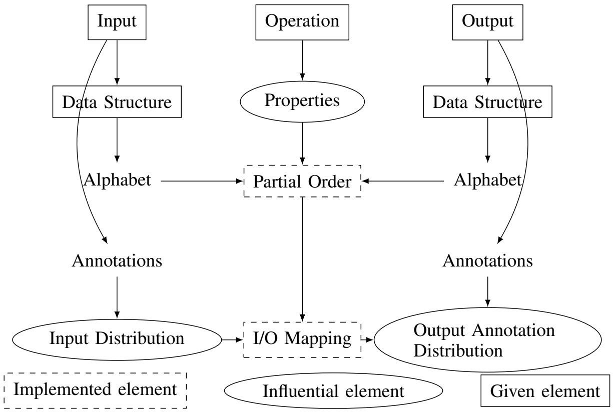

Fig. 1: Method participating elements

Propagating data properties across operations can be achieved with the help of a mapping between input and output data property representations. Further the mapping has to preserve both data and operation properties by taking into account both types of properties during mapping construction.

Data properties, represented as annotations, need not be value related. That is the property may also depend on attributes of the measure or the processing environment. Hence the properties may not be directly detectable in the output data. The representation of all possible annotations on the input or output data structure is called an *alphabet* (See Fig. 1 (Alphabet)).

Further *operation properties* (See Fig. 1 (Properties)) help identify which of the inputs participate in the computation of the result and if all inputs contribute equally. Operation properties are represented as a partial order amongst elements of the input and output alphabets (See Fig. 1 (Partial Order)). We will describe a technique to construct a partial order preserving operation properties. We illustrate the operation properties with the help of weighted and un-weighted average calculations on different data structures (See Fig. 3) and (See Fig. 4) respectively.

Besides operation properties the mapping (See Fig. 1 (I/O Mapping)) has to preserve also data properties. This is ensured by maximising the data property information as part of the propagation. This maximisation is measured with the help of the *output annotation distribution* (See Fig. 1 (Output Annotation Distribution)).

We shall now illustrate each element of the method in more depth and explain the techniques involved. Starting with data properties and their representation.

IV. Data properties

In related work (See Section II) we provide a wealth of examples of what are possible sources of data properties. Besides value related properties like outlier, there are process properties, common in provenance data, and semantic properties.

The only common element between the three kinds of properties is their presence.

We are then left, only, with the possibility of representing the presence or absence of a property. Hence a representation can be built by attaching a 1 or a 0, respectively, at the corresponding data structure position. An *annotation* is hence the name for the representation of a property.

We will now in turn introduce how the data properties are represented and subsequently quantified.

A. Alphabet construction

The *alphabet* is constituted of all different configurations of annotations on top of a data structure. Regardless of it being the input or the output of the operation. For example, when averaging four values (See Fig. 3 $(x_1; x_2; x_3; x_4)$) we have four elements which can be independently annotated. Hence all permutations of four zeros and ones in the operation input denote its input alphabet. Similarly all the combinations of one 0 or 1 construct the output alphabet (See Fig. 3 (x_o)).

B. Annotation distribution

Given an alphabet we count how many times each element occurs in a data set. This count, once normalised, constitutes the *annotation distribution*. The count can be made in the input or, after propagation, in the output. Giving the *input distribution* for the input. And the *output annotation distribution* for the output.

To evaluate changes in the annotation we perform a third measure. We count the number of times the input alphabet elements occur in the output of the operation. That is after the input data set has been propagated, this measurement is called the *output stream distribution*.

The different annotation distributions defined here play diverse roles in the propagation and evaluation of annotations.

V. Operation properties

We have seen that operation properties play a role in the propagation of annotations (See Section III). Further only the presence or absence of a data property is relevant to the propagation of the annotation (See Section IV). Hence out of all possible properties only ones helping determine the presence or absence of the property in the output are of interest. For example, if an operation is *weighted* or *un-weighted* determines if all the inputs contribute equally or not to the output. And by extension how much an annotated input, relative to the others, contributes to the output. Further an operation in which not all inputs contribute to an output, such as a vector average (See Fig. 4), is said to have the property of *localisation*.

The present work deals only with linear operations, leading to the following principle: The more annotated an input, the more likely is the output to be annotated. Owing to the additive nature of the linear combination of inputs in the

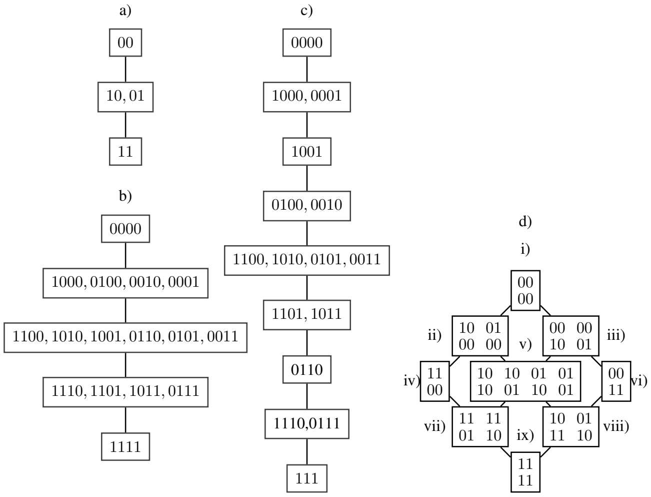

Fig. 2: Partial orders for un-weighted averages for 2 (a) and 4 (b) input values; weighted average (c) and localised average (d)

operation. This enables the construction of a *partial order* [**davey2002introduction**] between the alphabet elements, annotations, of the operation input. We represent operation properties as partial orders.

A. Un-weighted operations

$$\text{a) } x_o = \frac{1}{4}\sum_{i=1}^{4} x_i$$

$$\text{b) } x_o = \frac{1}{4}\sum_{i=1}^{4} w_i x_i \text{ With } W = [1, 5, 5, 1]$$

$$\text{c) } x_o = \frac{1}{4}\sum_{i=1}^{4} w_i x_i \text{ With } W = [1, 7, 7, 1]$$

x_1 x_2 x_3 x_4
Average
x_o

Fig. 3: Weighted and un-weighted average operations

In Un-weighted operations all inputs participate equally in the computation of the result. Hence only the number of annotations present is important, not their respective locations in the input. Inputs with the same number of annotations are equivalent. This creates equivalence classes amongst the alphabet elements, arranged in the partial order (See Fig. 2 (b))). Further the more annotations in the input the more likely is the output to be annotated. Giving an order in which the elements with a higher number of 1s are placed closer to the lower extreme of the partial order (See Fig. 2 (b)).

B. Weighted operations

In weighted operations the weight associated with each input indicates its importance in the computation. We can then use the weights associated to each input for the construction of the partial order. The partial order may contain empty classes, that is it does not need to preserve the distance between alphabet elements. As such only the relative weights define the partial order of the operation. This can be seen in Table I where two order associations (Order 1 and Order 2), yielding the same partial order (See Fig. 2 (c)), are calculated from two different weight sets (See Fig. 3 (b;c)). The order association is calculated by summing the weights of the annotated elements in the input alphabet.

TABLE I: Order association for the input alphabet elements, weighted average operation (See Fig. 3 (b;c))

Input	Order 1	Order 2
0000	0	0
0001	1	1
0010	5	7
0100	5	7
1000	1	1
0011	6	8
0101	6	8
0110	10	14
1001	2	2
1010	6	8
1100	6	8
0111	11	15
1011	7	9
1101	7	9
1110	11	15
1111	12	16

Further it can be noted that the partial order for an un-weighted operation can be constructed by assigning the same weight to all the inputs. Un-weighted operations are then a specific case of weighted operations in which all elements contribute equally.

C. Localised operations

In localised operations not all inputs contribute to all outputs. This is the case in a vector average operation (See Fig. 4). The two outputs (See Fig. 4 $(x_o; y_o)$) only depend on x_i and y_i respectively. This allows to compute independently both outputs. The property is reflected in the partial order (See Fig. 2 (d)).

$$(x_0, y_0) = (\frac{1}{2}\sum_{i=1}^{2} x_i, \frac{1}{2}\sum_{i=1}^{2} y_i)$$

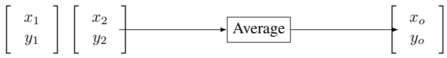

Fig. 4: Localised operation example: Vector average

In localised operations the position of the annotation in the input plays a role. An annotation in the x values only contributes to the annotation of the output x_o. This property makes the representation of the partial order a *diamond* (See Fig. 2 (d)).

Localisation enables the construction of the partial order with the help of the cross product. That is the partial order for the input of the vector average operation (See Fig. 2 (d)) is the cross product of the partial orders for two independent un-weighted averages (See Fig. 2 (a)).

VI. Mapping construction

With the data properties represented as alphabet elements (See Section IV), their information quantified with the help of the annotation distribution (See Section IV-B), and the operation properties represented as partial orders (See Section

V) we can now focus on the construction of the mapping enabling the propagation of annotations.

For annotations to retain their information, both data and operation properties have to be preserved. This implies the information contained in the input annotations needs to be preserved in the output.

We present here our method for propagating annotations whilst maximising the information preserved in the output. A technique for combining mappings enabling the propagation over operations with larger output data structures is also described.

A. Adaptive approach

The adaptive approach requires as inputs the input annotation distribution and both the input and output partial orders. The aim of the approach is to maximise the information, i.e. the entropy, of the output annotations. There exist two equivalent techniques one being an optimisation for small output alphabets.

The first technique consists of an exhaustive search through all the possible mappings. Computing the output distribution for each valid mapping, and selecting the one with the highest entropy. The output distribution is computed summing for each output the input probabilities of the input elements mapped to it. The complexity for this technique resides in the enumeration of all valid mappings which can be lengthy for large input and output alphabets.

The second technique can only be applied to operations whose output partial order consists of only two elements. This is the case of a large class of operations, i.e. all aggregations of simple data types. To maximise the entropy of the output distribution the same probability should be assigned to both outputs. This is achieved by finding the median of the input cumulative distribution.. However the bin sequence of the input cumulative probability distribution needs to be adapted to reflect the input partial order. That is the probabilities of the elements belonging to equivalence classes are summed, since their elements can not be differentiated, and the ordering has to match that of the input partial order.

B. Mapping combination

When an operation is localised a mapping can be constructed for each independent output. The independent mappings can be combined to form a suitable propagation mapping. The combination of mappings is done by performing the cross product of all the mappings amongst themselves. This approach enables for the optimised version of the adaptive approach to be used for the derivation of the independent mappings.

It further allows to decompose complex data structures into smaller ones, determine the mapping for the partial data structure, using the optimised method, and then recombine to obtain a total mapping. This offers a speed advantage for large localised operations.

VII. Evaluation

We aim to evaluate two aspects of our method: i) The adaptability of our method to diverse kinds of annotations and ii) mapping composability in the case of localised operations.

Before we delve into the two experiments certain aspects of the experimental setup and the evaluation metrics are worth considering.

A. Experimental setup

In order to evaluate our technique two things are required, data and a baseline technique for comparison. We explain here in greater detail how the synthetic data used in our experiments is generated and the manual propagation technique used as a baseline.

1) Data generation: Both experiments are performed on synthetic annotations. The annotations are generated with a uniformly distributed function giving values between zero and one. Then four arbitrary thresholds (A1= 0.95; A2= 0.80; A3= 0.5; A4= 0.2) are selected to annotate data values above them. Generating four input annotations with four different distributions (See Fig. 5 (Input)). For instance A1 represents a typical outlier annotation since very little data are generally outliers. A4 represents a process or provenance based annotation since most of the data would be originating from the same sensor or operation.

2) Manual propagation approach: A naive way of constructing the propagation mapping is to assign manually the elements of the input alphabet to elements of the output alphabet. This task can be tedious and error prone for operations with large output alphabets. This technique is used as a baseline for the propagation.

B. Evaluation metrics

We have seen previously that propagation aims to preserve the annotation. This can be interpreted twofold, to preserve the input distribution, or to maximise the entropy of the annotation. Both interpretations are evaluated with measurements based on the Jensen-Shannon Divergence (JSD).

The preservation of the input distribution is evaluated by measuring the JSD between *input annotation distribution* and *output stream distribution* (See Section IV-B). The lower the divergence the more preserved the annotation.

The second interpretation, maximisation of the annotation entropy, is evaluated as the JSD between the *output stream distribution* and the equi-probable distribution. The equi-probable distribution having the most entropy and entropy being a concave function any deviation will result in less entropy. We show in appendix (A) how JSD relates to entropy.

C. Adaptability

The first experiment aims to verify the adaptability of our propagation technique (See Section VI-A). For this we compare the propagated annotations across an un-weighted average (See Fig. 3 (a)) between the manual and the adaptive techniques. For the manual technique, three mappings are selected, two extremes. minimum and maximum propagation, and a third one in the middle. That is two mappings with all elements, except the extremes, assigned to propagate or not respectively and a third mapping in which half the elements propagate.

The propagations are then compared with the help of the evaluation metrics defined earlier (See Section VII-B).

D. Operation property representation

The second experiment aims to verify that for localised operations mappings can be combined. That is, the same annotation preserving mapping can be obtained in two ways, by directly applying the adaptive technique (See Section VI-A), or by applying the adaptive technique to each localised part of the input and then combining the two independent mappings (See Section VI-B). As part of the experiment we also evaluate the propagation of the two separate inputs.

This is evaluated by propagating annotations across an average vector operation (See Fig. 4) and measuring with the help of the metrics defined earlier(See Section VII-B).

VIII. Results

We now present the results of the two experiments described earlier (See Section VIII-A) and (See Section VIII-B).

A. Adaptability

The first column (See Fig. 5 (Input)) presents the histograms of the input annotation distribution for four different annotations (A1, A2, A3, A4). The second column depicts the histogram of the output stream distribution for our proposed adaptive technique (See Fig. 5 (Adaptive)). Third and successive columns present the histograms of the output stream distribution for three manually selected mappings, minimum, middle and maximum (Min, Mid, Max) respectively.

Annotation preservation can be seen in the two following ways in the output stream distribution histograms: i) The more resembling the output stream distribution and the input annotation distribution histograms are the more preserved the input distribution is. ii) The more equi-probable the output stream distribution, the more entropy the annotation holds, i.e. the more information is preserved. This is represented by a more *flat* output stream distribution histogram.

1) Input distribution preservation: is achieved by the middle manual mapping (See Fig. 5 (Mid)) since for all annotations its output stream histogram is the most resembling to the input annotation histogram. This is further verified in figure 6 since the middle (Mid) mapping always has the lowest divergence with the input distribution (JSD I/O).

2) Entropy maximisation: is achieved by the adaptive method (See Fig. 5 (Adaptive)) since for each annotation it shares the histogram with the most entropy amongst all the histograms of the manually constructed mappings. That is, in the case of annotations A1 and A2 with the minimum mapping, for annotation A3 the middle mapping and for annotation A4 the maximum mapping. Hence the adaptive method performs as well as the manual method, without the need of inspecting the data and manually constructing the mapping. This is further verified in figure 6 where the divergence with the equi-probable distribution (JSD Equi/O) is always the lowest for the adaptive method, always matching the lowest of the manually set thresholds for each annotation.

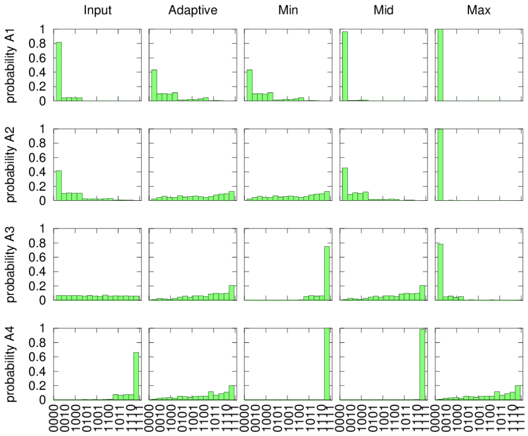

Fig. 5: Input stream histograms for Annotations (A1,A2,A3,A4) and output stream histograms for three different techniques (Separate, Combined, and Exhaustive)

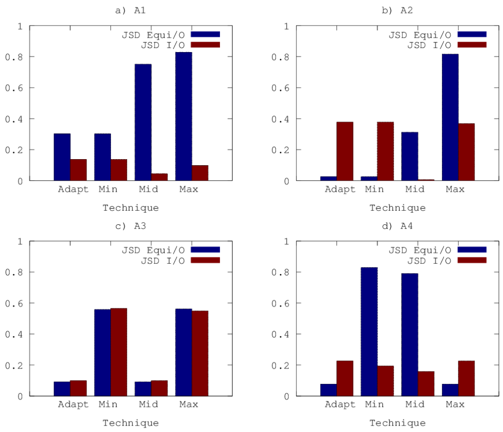

Fig. 6: Pairwise JSD (equi-probable distribution, output stream distribution) and (input stream distribution,output stream distribution) for manual thresholds (Min,Max), Adaptive and Equiprobable (Mid), and annotations (A1,A2,A3, A4).

B. Operation property representation

In figure 7 the same propagation behaviour can be seen for all four annotations, with regards to both the preservation of the input annotation distribution and the entropy maximisation criteria. That is, due to the localisation of the inputs the propagation can be handled independently (See Fig. 7 (Separate)) giving the same annotation preservation as for our mapping combination technique (See Fig. 7 Combined) and the non optimised version of the adaptive technique which we call exhaustive (See Fig. 7 (Exhaustive)).

This result can be interpreted in two ways: i) When propagating annotations across a localised operation we can compute an independent mapping for each input or a global mapping with our adaptive technique warranting that we find the same optimal mapping with regards to information preservation. ii) It further supports the idea that a family of operations is characterised by a single partial order.

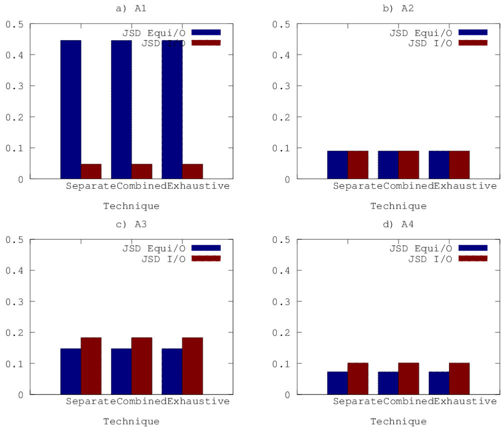

Fig. 7: Pairwise JSD (equi-probable distribution, output stream distribution) and (input stream distribution,output stream distribution) for Separate, Combined and Exhaustive search techniques, and Annotations (A1,A2,A3,A4).

IX. Conclusions and future work

We have shown that suitable representations of data and operation properties can be found and manipulated in order to propagate data properties to the output of linear data reducing operations. Further a given set of operation properties can uniquely identify an operation.

We presented a generic technique for the construction of partial orders describing operation properties and two mapping construction techniques preserving data properties. Such a preservation can be interpreted in two ways, preservation of the input distribution or preservation of the information of the data properties. We present an optimal technique for the information preservation evaluated in terms of the Jensen-Shannon Divergence. The manual technique used as a benchmark in the experiments is optimal with regards to distribution preservation.

Two experiments where carried out on synthetic data. One to verify the preservation of data properties and a second to verify the composability of propagation mappings in localised operations. The composability result further validates the applicability of our adaptive technique to more complex data structures for localised operations.

In future we aim to extend the technique to other classes of operations. Notably non-linear and data expanding operations such as interpolation.

Appendix

We infer the maximisation of entropy as the difference between equi-probable and output stream distributions. The lower the divergence is, the higher the entropy of the output stream distribution.

This is illustrated in the following argument. The Jensen-Shannon Divergence (JSD) is expressed in terms of the entropy of two statistical distributions P and Q as $JSD(P,Q) = H(\frac{P+Q}{2}) - \frac{1}{2}(H(P) + H(Q))$ [**briet2009properties**]. If we consider that P is an equi-probable distribution we have $H(\frac{P+Q}{2}) < H(P)$ and $H(Q) < H(P)$ so we can maximise $JSD(P,Q)$ with $H(P) - \frac{\alpha}{2}H(P)$ where $1 \leq \alpha < 2$. Having $1 \leq \alpha$ when $H(Q) = 0$ and $\alpha < 2$ because $P \neq Q$. Giving $JSD(P,Q) < \frac{2-\alpha}{2}H(P)$. Thus $JSD(P,Q)$ is bound by $0 < JSD(P,Q) \leq \frac{1}{2}H(P)$. So as long as $P \neq Q$. $JSD(P,Q)$ is positive and greater than 0. So once Q diverges from the equi-probable distribution $JSD(P,Q)$ becomes positive. Entropy is a continuous function hence $JSD(P,Q)$ is also a continuous function. The divergence is hence progressive, the more entropy is lost the more the divergence increases.