The Identification problem: A description

Juan Amiguet-Vercher
Database group
University of Twente
Enschede, The Nehterlands
J.Amiguet@utwente.nl

Peter Apers
Database group
University of Twente
Enschede, The Nehterlands
P.M.G.Apers@utwente.nl

Andreas Wombacher
Database group
University of Twente
Enschede, The Nehterlands
A.Wombacher@utwente.nl

***Abstract*—Scientific data are often annotated based on their properties, which are not maintained during further data processing. Not maintaining annotations results in loss of information. Decisions made on such incomplete information may be wrong. In this paper the problem of propagating annotations along a data processing chain is formulated. In particular, an annotation of a data element is an identification that this data element exhibits a specific property. The propagation of this property from the input of an operation to its output is called the identification problem. In this paper the identification problem is described as a clustering problem.**

***Keywords*-data processing; meta-data; operation; data identification;**

I. Introduction

In todays world large volumes of data are captured for diverse applications. To facilitate the interpretation of the data annotations are common both in business and scientific applications. Large amounts of work have been done in generating annotations notably in the field of data-mining. It is customary to issue an annotation identifying a class for data. Sadly annotations are only generated as the end result, and little effort has been done in propagating them during further data processing.

Not maintaining annotations results in loss of information. If an annotation identifies a datum as anomalous the user can exercise more caution when making a decision based on it. Or consider the result of the application as non valid altogether. The annotations denote properties of the data such as belonging to a certain class. If the annotation is not propagated it may be lost. Since it is not always possible to use the same classifier on both input and output data.

In this paper the problem of propagating annotations along a data processing chain is formulated. The propagation of annotations can be understood as the generation of a mapping between input and output annotations. This mapping is derived as the solution of a clustering operation. By assigning input to output annotations under constraints derived from the existence of the annotation, and characteristics of the operation and the data structure.

In particular, the annotation of a data element is an *identification* that this data element exhibits a specific property. The propagation of this property from the input of an operation to its output is called the identification problem. In this paper the *identification problem* is described as a clustering problem. Part of the identification resides in deriving the presence of the annotation in the output and how much of it is annotated. Further our position is the identification problem can be posed as a clustering problem. To delineate the position we derive the description of the problem from elicited requirements and also provide for two operations an initial solution to the clustering problem.

An initial solution is presented using entropy as an optimisation function.

A more formal description of the problem in terms of topologies, enabling the further development and validation is also described.

II. Motivation

The meaning of the annotation, further called property of data elements, determines the applications of the identification problem. The meaning is assigned and interpreted by the user, but is not necessary for solving the identification problem.

If the property signifies data captured during a system malfunction. We can determine if the malfunction matters and if so what is its extent in the final result. Based on such an assessment the end user can decide if the results produced by the application are valid or not.

A. Application description

To illustrate the identification problem we introduce a sample weather monitoring application (Fig. 1).

The application consists of three temperature sensors (Fig. 1 part: S1,S2,S3) located in a mountainous area. The sensors report data on an hourly basis. They are located at given locations (Fig. 1 part: (X,Y)). Simple statistics (Minimum, Average and Maximum) are calculated from the data (Fig. 1 part: Statistics) on a weekly basis. From the statistics the weekly average is separated in the projection operation (Fig. 1 part: Projection). The two other values are disregarded in this sample application, to keep the explanation succinct. Based on the weekly average and the locations of the sensors a spacial interpolation is done (Fig. 1 part: Interpolation).

The temperature interpolation is used by a team of hydrologists in order to predict the water flow arriving in the related catchment.

The three sensors report on a timely basis and the results are incorporated into the water flow prediction models.

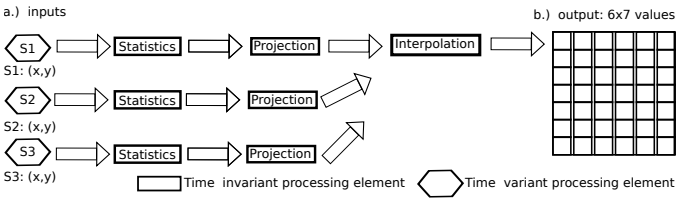

Figure 1. Sample weather monitoring application

The temperature sensors are deployed in a hostile environment. Strong winds, abundant precipitations, and prolonged cloud cover may hinder the correct functioning of the temperature sensor. As such they undergo regular maintenance. As part of the maintenance tasks the sensors are re-calibrated and broken sensors are replaced.

Besides annotating the data as being acquired during a maintenance period. In the case of an un-calibrated sensor the person performing the maintenance checks when was the last calibration done. And estimates based on the weather patterns from which moment was the calibration of sensor no longer reliable. Both the maintenance period and the estimated un-calibration of the sensor are properties which the data exhibit. These two properties are not related to the data values. They can not be derived from the values reported by the sensor. Hence, we can not rely on data values to solve the identification problem.

1) Does the anomaly matter?: We have now to determine if the data generated by one or many un-calibrated sensors impacts on the end result. For this we have to determine if any such data is present in the output. The simplest way of achieving this is by attaching a marker to the data. And ensure that the marker is found on the output only if the property is found in the data.

For example in the case of the statistics operation (Fig. 1 part: Statistics) if all the data for one week where un-calibrated. It would make sense that the output where also un-calibrated.

However if there where some properly calibrated data in the data being aggregated as part of the statistics operation. It would be impossible to recognise an individual calibrated datum. The ratio from inputs to outputs although constant prevents the tracking of a single datum in the output. From a weeks data, only three values are produced, the Minimum, Average and Maximum. We can not recognise directly a single reading in the output of the Statistics operation.

2) If so, what is its extent?: If there is un-calibrated data present in the input. We are also interested in knowing what is the extent of this data. That is, in the case of an interpolation, (Fig. 1 part: Interpolation) the un-calibrated data is used to compute part or all of the data. Determining to which parts of the output it contributes enables the user to determine the impact of the un-calibrated data in the result. And as such the validity of the results.

Further spacial interpolation operations represent a change in the data structure. In the case of our application three averages are combined together. The locations of each one of the averages determine their participation in the computation of an output. This participation is a function of the distance.

We have then three single data points as input originating form the projection operation (Fig. 1 part: Projection). Giving as output an array with of six times seven values (Fig. 1 part: b.). This change of data structure enlarges the impact of the un-calibrated input. Hence it is important to ascertain the extent of the un-calibrated data in the ouput.

3) General operations: Aggregations and interpolations are two examples of orthogonal operations. That is aggregations only have an impact on the amount of data going in and out, but not on the data structure. Interpolations have the same amount of data, namely one in one out, but they modify the data structure. Other operations can be classified as combinations of both. That is in terms of ratios between inputs and outputs, and changes in the data structure.

Given data and a property which it exhibits, we want to:

- Determine if the property is present.
- Determine the extent of the property.

In the output of any operation.

III. Required properties

We have previously stated our goals and illustrated the issues conforming the identification problem. In this section we will describe the properties which are required for a generic data identification method.

A. Data Identification

Previously we motivated the need to identify data holding a given property. For the identification problem it does not matter what the property is, or its origin. Only that it exists in the data.

This makes the identification problem independent for each property. The data are hence identified by attaching an independent marker per interesting property to each data point. Data are considered identified if they have at least one marker attached to it.

A *marker* is a flag placed on data identifying it as holding a specific property.

We can then state the first two required properties of the identification probles as:

Req. 1: Properties are independent from each other.
Req. 2: Data are identified by a marker.

B. Operation properties

In order to solve the identification problem we require some information from the operation. The operation determines the contribution of an input in the computation of an output. We also aim to be as generic as possible. This forbids the direct analysis of the operation as part of the identification problem. Thus, the approach is to classify operations based on their properties to nevertheless influence the identification problem.

Req. 3: We require operation properties to solve the identification problem.

The properties we use concern the interaction of the operation with the input and output data structures:

- Are several data aggregated into a single output datum?
- Do the input and output data structure differ?
- Given two operations on same input and output data structures do they have the same solution to the identification problem?

An operation representing each class is used to illustrate each of these aspects.

We focus mainly on aggregations as examples of temporal transforms and interpolations as examples of structural transforms. The two operations represent orthogonal cases, that is, for the identification problem, the combination of the solution to both extreme cases allows to handle many other operations.

1) Temporal transformation: In temporal transformations several input data are combined into a single output datum. There exist several kinds of temporal transformation. Aggregations being the most common. An example of an aggregation is an average. In the case of an average all inputs participate equally.

Follows that the number of identified inputs, together with a threshold, influences the identification problem. The threshold is chosen to avoid the output being over or under identified. That is to avoid all the outputs or none of them being identified, respectively.

Req. 4: The number of identified inputs, together with a threshold, influences the identification problem, for temporal transformations.

2) Structural transformation: When the output data structure differs from the input data structure the operation performs a structural transformation. For example in the case of a spatial interpolation [12].

The value of each output is the outcome of the computation on at least one input. To ascertain if an input participates in the calculation of an output we require the distance between the location of the input and the output and a threshold. A way of measuring the distance is by projecting the locations of the inputs directly on to the output data structure. Then a suitable threshold is chosen on the distance to avoid over or under identifying the output.

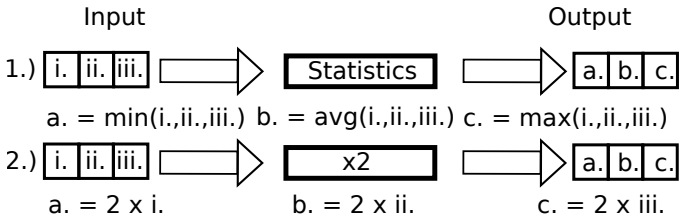

Figure 2. Simple statistics operation

Req. 5: The projection of the locations of the inputs into the output data structure is necessary to determine the distance between inputs and outputs.

Req. 6: The distance from an input and a threshold, influences the identification problem, for structural transformations.

3) Operation interpretation of the data structure: We have seen that the relationship between the operation and the data structure plays a role in the identification problem. However not all operations interpret a data structure in the same way.

We can see in (Fig. 2) how two operations with the same input and output data structures require different solutions to the identification problem. The first operation (Fig. 2 part: 1.) calculates basic statistics, (minimum, average, and maximum) on the inputs. As such each of the three outputs (a.,b.,c.) is an aggregation of all 3 inputs (i.,ii.,iii.). Conversely for a multiplication operation (Fig. 2 part: 2.) we see that each of the outputs depends on a single input. Summing up:

Req. 7: The interpretation of the input and output data structures by the operation influences the identification problem.

C. Preserving the identification relationship

For the identification of the output to be useful the identified data has to preserve its relationship with the non identified data. That is the output should not be over or under identified with regards to the input. This is achieved by selecting a threshold.

The purpose of the threshold is to preserve the relationship between the inputs and the outputs. That is, if an input A has less identified data than an input B then the corresponding identification on the output must preserve this characteristic.

This threshold is applied on the number of identified inputs, or on the distance between projected inputs and outputs, for temporal or structural transforms respectively.

Req. 8: A threshold is chosen in both the amount of identified inputs and the distance between projected inputs and outputs. Thus warranting that the property is not over or under represented in the output.

D. Handling multiple operations

We know that in the case of a structural transform the distance between projected inputs and outputs influences the

identification problem. Further we can treat this distance as a count by considering the number of elements on the output data structure separating the projected inputs and the output.

Both the number of identified inputs and the distance from a projected input to an output are orthogonal. That is both counts influence the identification process independently.

Further on both counts a threshold is applied in order to preserve the identification relationship. Hence dealing with a generic operation is possible through the combination of both counts and associated thresholds.

Req. 9: The identification problem for a generic operation can be handled through the combination of both counts and the selection of an appropriate thresholds.

To facilitate future reference we summarise in (Fig. 3) the required properties.

Req. 1: Properties are independent from each other.
Req. 2: Data are identified by a marker.
Req. 3: We require operation properties to solve the identification problem.
Req. 4: The number of identified inputs, together with a threshold, influences the identification problem, for temporal transformations.
Req. 5: The projection of the locations of the inputs into the output data structure is necessary to determine the distance between inputs and outputs.
Req. 6: The distance from an input and a threshold, influences the identification problem, for structural transformations.
Req. 7: The interpretation of the input and output data structures by the operation influences the identification problem.
Req. 8: A threshold is chosen in both the amount of identified inputs and the distance between projected inputs and outputs. Thus warranting that the property is not over or under represented in the output.
Req. 9: The identification problem for a generic operation can be handled through the combination of both counts and the selection of an appropriate thresholds.

Figure 3. Required properties for information identification

IV. Revised problem statement

We will now provide and illustrate a more abstract description of the problem and introduce the relevant building blocks.

A. Identification problem

The identification problem consists of identifying output data. This is done based on the existence of the property in the input data and the interaction between the operation and the input and output data structures.

The identification problem can be solved with a mapping. In order to construct such a mapping we use clustering.

The mapping relates input and output markers, which we first introduce.

1) Marker: A *marker* consists of a binary flag indicating the presence of a given property in the data. A binary flag suffices since we consider all properties to be independent from each other (Fig. 3 req: 1). Since the data is stored in a data structure the marker is associated with the data structure. This fulfils required property (Fig. 3 req: 2).

Now that markers have been introduced we illustrate how the mapping is built through clustering.

2) Clustering: Clustering has two outputs. The assignment of the inputs to clusters and a centroid for each cluster, a representative.

The assignment is made so that the similarity between the inputs is the greatest. And the representative is chosen to be the most resemblant to all the inputs in that cluster.

We will see now how the clustering operation builds the mapping solving the identification problem both for temporal and structural transformations.

Temporal transformations: In the case of temporal transformation the solution to the identification problem is one where many inputs are mapped to an output. Temporal transformations have a single output. This output can contain the property or not. Yet we have more than two inputs, otherwise the operation would be trivial.

We know that several input markers have to be treated equally (Sect. III-C). That is those elements are similar with regards to the mapping. This makes the assignment to clusters the ideal operation for constructing such a mapping.

Required property (Fig. 3 req: 4) is fulfilled by using a clustering operation. Hence supporting our position that the identification problem is a clustering problem.

Structural transformations: Further in the case of structural transformations the solution to the identification problem is one where one input is mapped to one output. Yet there are multiple possible outputs for one input. All elements within the same distance of an input are affected by the data from that input in the same manner. So if an input is identified, all output markers with nearby markers identified are possible representatives,

Since we can only have one output marker its best if it were the most representative. This makes the selection of a cluster centroid the ideal operation for constructing such a mapping.

Required properties (Fig. 3 req: 5,6) are fulfilled by using a clustering operation. Further supporting our position that the identification problem is a clustering problem.

3) Data structure interpretation: The interpretation of the input and output data structures by the operation also plays a role in the identification problem (Fig. 3 req: 7). It structures the input and output of the clustering problem. This is achieved by building a partial order amongst the inputs and the outputs.

This is best illustrated with an example. In the case of the statistics operation (Fig. 2 part: 1.) all three outputs (a.-c.) can not be differentiated. That is all three are outputs of the same inputs under the same kind of operation, an aggregation. This can be expressed by restricting the output partial order two only two markers. Namely either all outputs are identified or no outputs are identified.

B. Handling general operations

From the clustering operation we obtain two different results. The assignment to clusters and the centroids. Each of the two results represents a solution to the identification problem for a kind of operation. We know that temporal and structural transforms are orthogonal in their properties.

A mapping which combines both outputs from a clustering would solve the identification problem for an operation which was both a temporal and a structural transform. Such a mapping would associate several input markers to one output marker. And the output marker would be selected to be the most representative amongst similar output markers. Required property (Fig. 3 req: 9) is hence fulfilled by using a clustering operation.

C. Topological framework

We can now simplify the description of the identification problem by describing a formal framework for clustering. Illustrating the clustering/identification problem in terms of sets, topologies and functions between them (Fig. 4).

Partial orders can be build amongst the input and output markers. This enables the building of two topologies (Fig. 4 part: T, T′) for the input and output markers respectively [13]. One or both of the topologies, depending on the kind of operation are then clustered. The outcome of this clustering are topologies (Fig. 4 part: T″,T‴) respectively.

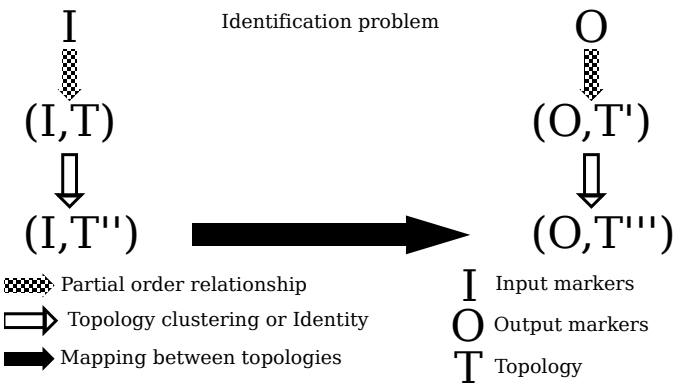

Figure 4. Topological description of the method

We then apply clustering to solve the identification problem for different kinds of operations.

1) Clustering the input topology: In the case of temporal transformation we associate several input markers to one output marker. This is achieved by clustering the input topology (Fig. 4 part: T) into as many clusters as elements in the output topology (Fig. 4 part: T′). The output topology is left as is through an identity function and becomes (Fig. 4 part: T″). Making the mapping (Fig. 4 part: Mapping between topologies) one-to-one. That is each cluster of input markers has an assigned output marker.

Bear in mind that one of the markers indicates presence of the property, the other not presence. Hence we can determine if the property is present in the output.

2) Clustering the output topology: Further in the case of a structural transform several output markers are suitable to represent an input marker. That is each input marker can be assigned to a subset of the possible output markers. The subsets need not be disjoint. By clustering the output topology (Fig. 4 part: T′) and reducing it to only the centroids. Hence a solution to the identification problem. In other words the clustering transforms the output topology (Fig. 4 part: T′) to (Fig. 4 part: T‴) making the mapping (Fig. 4 part: Mapping between topologies) one-to-one. That is each input marker is assigned an output marker.

The output markers represent areas of the data structure where the property is present. As such the determination of an output marker enables to asses how present is the property in the output of an operation.

D. Clustering constraints

The topological framework poses the identification problem as a clustering problem. However certain constraints are placed to facilitate the creation of the mapping and represent operation and data structure properties. These constraints are expressed in the form of the topologies.

1) Topologies: The characteristics in the relations between elements (Fig. 3 req: 3) build the topologies.

For example in the case of temporal transformations the number of identified inputs influences the identification problem (Fig. 3 req: 4). Hence all inputs containing the same number of markers have to be considered equal. Further inputs with more markers, represent data in which the property is more present. As such the outputs are more likely to posses the property. This characteristic of the partial order is easily expressed in a string topology.

Similarly in the case of structural transforms the projection of the inputs in the output data structure influences the identification problem (Fig. 3 req: 5). This influence is easily represented by ensuring that the topology takes on a diamond shape. This enables the clustering to consider the projected positions of the inputs.

The representation and manipulation of the partial orders as topologies fulfils required properties (Fig. 3 req: 3.,4.,5.)

Further the introduction of topologies provides suitable properties for the derivation of the solution, and its formal validation.

V. Initial solution

We have seen that the identification problem is solved by building a one-to-one mapping between two topologies. The two topologies are the result of a clustering operation. Before they are clustered the topologies express a partial order over the markers. This partial order incorporates characteristics of

the data structure and the interpretation of the data structures by the operation.

The elements of the topologies are markers. Markers are elements of the set $\{i \cdot i \in \mathbb{Z}_2^n\}$. Where n is the size of the data structure. This choice of representation is made since we are only interested in the presence or absence of the property.

The mapping is then built by clustering the markers taking into account the partial orders. This clustering is implemented as an optimisation under constraints.

A. Clustering as an optimisation problem

To implement the clustering of the markers as an optimisation problem two elements are necessary a set of constraints, and an optimisation function.

The constraints ensure that the solution is a valid one. In our case the constraints are derived from the partial orders built amongst the markers.

Further by defining a measure over the set of possible solutions the optimisation function aims to reach the optimal mapping. The maximum value is reached for the optimal solution. We also need one common measurement for both clustering outputs.

1) Temporal transformations: In the case of temporal transformations the requirement to not over or not under identify the property translates easily into an entropy measure. The frequency of the output markers is calculated based on the frequency of the input markers and the mapping. That is the frequency of an output marker is the sum of the frequencies of all the input markers that are mapped to it. The output marker frequencies conform hence a probability distribution from which the entropy can be calculated. Maximising the entropy warrants that the property is not over nor under identified in the output.

2) Structural transformations: For structural transformations the requirements is to find the most representative marker for each input. This is achieved by ensuring that the representative markers are as different as possible. The difference between representative markers can be measured in terms of the distance between them. Since the distance is bound, we can normalise it, making a normalised distance distribution. We then calculate the entropy of the normalised distance distribution. Maximising the entropy warrants that the chosen representative markers are as different as possible.

B. Aggregation

In the case of an aggregation operation we aim to find the assignment of each input marker to one of the two possible output markers. We do this by clustering the input markers under constraints derived from the partial orders (Sect. IV-D1). As an optimisation criteria we want both outputs to have the same frequency if at all possible (Sect. V-A). This warrants that the property is not over or under identified in the output.

Figure (Fig. 5) shows what a sample solution looks like for an aggregation. Below each marker (000, $\cdots$, 111) we have the frequency with which the marker appears in the input data set. The ovals in figure (Fig. 5) depict the two clusters that are built. The input markers contained in the upper oval are mapped to (1) output marker. Only (000) is mapped to (0) output maker, it has a very high frequency (0.75). The value of (0.75) means that most of the time the data does not exhibit the property. It is impossible, in this situation, to reach the ideal optimal frequency of one half for each of the two outputs markers.

We can also see in figure (Fig. 5) the input and output topologies. We know that all elements with a similar amount of markers have to be treated equally. This is represented by the string topology (Fig. 5 part: Inputs) in which the markers with the same number of ones are grouped together.

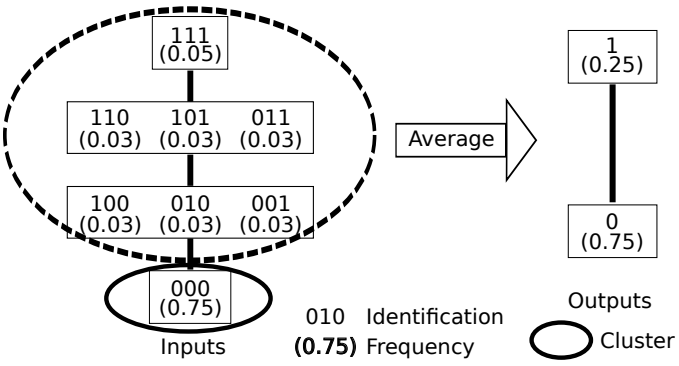

Figure 5. Optimal mapping between input and output identifications for an aggregation

C. Interpolation

In the case of an interpolation we aim to find the best representatives amongst the output markers for each input marker. We know that the positions of the projected input sources participate in the selection of the candidates. We call them *seeds*. Seeds are indicated by an @ sign in (Fig. 6).

For each one of the seeds a representative, potentially itself, is selected as the centroid of the cluster. The centroids have to be as different as possible. Hence as far apart from each other as possible.

We can see in (Fig. 6 part: 2.) that four centroids, marked with an "X" , have been selected. We see however that given the distances between them another equally optimal solution is possible. That is the distances between all the centroids do not change if markers (110) and (011) are chosen instead of (100) and (001) respectively.

Further in figure (Fig. 6 part: 2.) we see the input and output topologies. Both are diamond topologies enabling the projection of input markers to output markers. In order

to build a one-to-one mapping between these two topologies the output topology is clustered. Reducing the output topology to diamond containing only the centroids, and their relations.

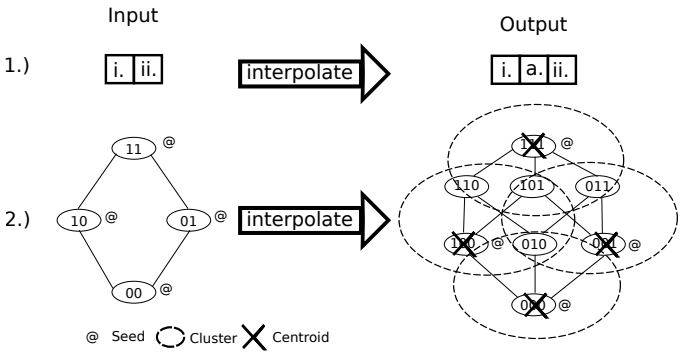

Figure 6. Optimal solution for a spacial interpolation

VI. Related work

Previous work by the same authors [1] defines in a less rigorous way the identification problem. Further, in the present article the identification problem is described as a clustering problem. A topological framework is introduced to facilitate further research.

A. Identification and meta-data

Markers identifying the existence of a property exist mainly as meta-data tags. A tag being a token attached to data denoting a property derived from the data. Meta-data tags are generally hand assigned. They aim to facilitate the organisation, retrieval and inference of information about images [4], [11], or documents [3].

The identification problem in this setting translates in to associating a tag to a document based on similarities. In [3] tag occurrence and co-occurrence together with the citation graph build a context in which the distance between documents can be measured. Inside this distance a similarity threshold is found during the traversal of the citation graph. Documents which are related, inside the threshold, share the tags.

A similar method can be applied to images. In [4] [11] the context is built on the basis of features extracted from the image. The time at which the image was taken is used in [4] to narrow down the context in which similarities are measured. Only images which are taken at around the same time are considered for similarity analysis. The similarity aims to share manually assigned tags in a collection of home made pictures. In such a setting the propagation of all the tags inside clusters, based on time and similar features, provides good results.

The similarity, implemented as a clustering operation in [11] relies on the detection of special regions in the image. The special regions are then classified. And a common representative for each class found. The tags on the representatives are then propagated to images containing similar regions.

We deviate from the prior works in that we consider the propagation of a property across an operation. Similarity does not play a direct role as such.

We share with the literature the concern of the potentially unbounded number of tags, in our case properties. In [6] the aim is to assist with the identification process. That is to supply to a user a series of suitable tags for a document. Leaving the identification problem in the hands of the user. The tags are selected based on similarity with previously annotated documents.

Tags can also be used to facilitate the sharing information about scientific tasks. In [9] scientific workflows are annotated. Enabling the re-utilisation of workflows and the documentation of the why and how of different scientific artefacts. The annotations provided by the scientist are also searchable enabling the re-use of scientific workflows or parts thereof by other scientist.

Annotations are also used to enrich chemical experiments [8]. In this work the annotations are provided by scientist in order to provide searching and reasoning on top of multiple chemical experiments. The capture aims to be as early as possible, sometimes even during the planning stages of the experiment. The annotations assist here with the identification and re-use of chemical experiment results.

A complementary field is that of data mining. In which, there exist several techniques for extracting data with desired properties. A survey of the ten most popular techniques can be found in [14]. All of the techniques mentioned rely on the values of the data to derive common properties. For example the expectation maximisation algorithm relies on data values to train a mixture of normal distributions. Based on this mixture new data points can be assigned to one of several distributions or classes. Data in a class can then be identified with a property.

We do not rely on data values, nor on attributes of the property. Only the presence or absence of the property together with operation and data structure properties influence the identification problem.

B. Properties on relational data

In the database domain data property propagation is a well understood problem. Mainly to convey provenance data on static relations [7], [5] and [10]. Propagation of properties on static relations is generally controlled through extensions in the query language enabling the user to specify if an annotation is to be propagated or not. All approaches share that the property markers are associated with the data, potentially holding several markers. The property of interest may also span several columns of the relation. To minimise the encoding of the area which is annotated on the table the optimal ordering of columns has to be determined [7].

There exist also the possibility of manipulating annotations in the same way as data, in [5] this is achieved through the introduction of keywords. In [7] the annotation table can be queried in the same way as any other table, enabling the querying of data through the annotations attached to it.

When manipulating relational data advantage can be taken of relational algebra to propagate annotations. Bowers [2] presents a calculus which enables the propagation of annotations across transforms expressed in relational algebra. Our work differs in the following points: i) In Bowers Annotations are composed to derive new annotations based on their understanding and ii) We do not restrict our operations to only a relational description. However there exist some resemblance in the way the identification problem is described. Both our approach and Bowers rely on the composition of functions: annotation, and transform, to describe the problem.

VII. Conclusions and future work

In this paper we have motivated and described the identification problem as a clustering problem. The identification problem aims to identify properties in the outputs of operations. This is done based on the input identification of the property and some information about the operation. The extra information are operation properties, hence decoupling the identification problem from the operation. Further the identification problem only relies on the existence or absence of the property. This restricts the identification problem to handle one property at a time.

The clustering problem used to describe the identification problem is justified as its two outputs, mapping and centroids handle the two orthogonal elements of the identification problem. That is the consideration of temporal and structural transformations.

We introduced a topological framework enabling the verification of techniques to solve the identification problem. And propose an initial solution to the identification problem as an optimisation problem. Two illustrative solutions to the identification problem are also described.

Further work is required to define constraints on the functions and topologies for different kinds of operations. The formal validation of the optimisation problem as a solution to the identification problem is work in progress.

References

[1] J. Amiguet, A. Wombacher, and T. E. Klifman. Annotations: dynamic semantics in stream processing. In *Proceedings of the 3rd workshop on Ph.D. students in information and knowledge management*, PIKM '10, pages 1–8. ACM, 2010.

[2] S. Bowers and B. Ludäscher. A calculus for propagating semantic annotations through scientific workflow queries. *Current Trends in Database Technology–EDBT 2006*, pages 712–723, 2006.

[3] A. Budura, S. Michel, P. Cudré-Mauroux, and K. Aberer. To tag or not to tag-: harvesting adjacent metadata in large-scale tagging systems. In *Proceedings 31st internat. ACM SIGIR conference on R&D in IR*, pages 733–734. ACM, 2008.

[4] R. Carvalho, S. Chapman, and F. Ciravegna. Attributing semantics to personal photographs. *Multimedia Tools and Applications*, 42:73–96, 2009.

[5] Laura Chiticariu, Wang Chiew Tan, and Gaurav Vijayvargiya. Dbnotes: a post-it system for relational databases based on provenance. In Fatma Özcan, editor, *SIGMOD Conference*, pages 942–944. ACM, 2005.

[6] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J.A. Tomlin, et al. Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In *12th internat. conf. on WWW*, page 186. ACM, 2003.

[7] Mohamed Y. Eltabakh, Walid G. Aref, Ahmed K. Elmagarmid, Mourad Ouzzani, and Yasin N. Silva. Supporting annotations on relations. In *EDBT '09*, pages 379–390, New York, NY, USA, 2009. ACM.

[8] J. Frey, D. De Roure, K. Taylor, J. Essex, H. Mills, and E. Zaluska. Combechem: A case study in provenance and annotation using the semantic web. *Provenance and Annotation of Data*, pages 270–277, 2006.

[9] A. Gándara, G. Chin, P. Pinheiro da Silva, S. White, C. Sivaramakrishnan, and T. Critchlow. Knowledge annotations in scientific workflows: an implementation in kepler. In *Scientific and Statistical Database Management*, pages 189–206. Springer, 2011.

[10] Floris Geerts, Anastasios Kementsietsidis, and Diego Milano. MONDRIAN: Annotating and querying databases through colors and blocks. In Ling Liu, Andreas Reuter, Kyu-Young Whang, and Jianjun Zhang, editors, *ICDE*, page 82. IEEE Computer Society, 2006.

[11] J.S. Hare and P.H. Lewis. Saliency-based models of image content and their application to auto-annotation by semantic propagation. In *Proceedings of Multimedia and the Semantic Web/European Semantic Web Conference*, volume 2005. Citeseer, 2005.

[12] N.S.N. Lam. Spatial interpolation methods: a review. *Cartography and Geographic Information Science*, 10(2):129–150, 1983.

[13] B. Mendelson. *Introduction to topology*. Dover Books on Mathematics Series. Dover Publications, 1990.

[14] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey McLachlan, Angus Ng, Bing Liu, Philip Yu, Zhi-Hua Zhou, Michael Steinbach, David Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14:1–37, 2008.