Information Space Models for Data Integration, and Entity Resolution

Reid Porter*[a], Linn Collins[b], James Powell[b], Reid Rivenburgh[c]
[a]Space Data Systems Group, Intelligence and Space Research Division,
[b]Knowledge Systems Team, Los Alamos Research Library,
[c]Information Sciences Group, Computer, Computational, and Statistical Sciences Division,
Los Alamos National Laboratory, NM, USA, 87545

ABSTRACT

Geospatial information systems provide a unique frame of reference to bring together a large and diverse set of data from a variety of sources. However, automating this process remains a challenge since: 1) data (particularly from sensors) is error prone and ambiguous, 2) analysis and visualization tools typically expect clean (or exact) data, and 3) it is difficult to describe how different data types and modalities relate to each other. In this paper we describe a data integration approach that can help address some of these challenges. Specifically we propose a light weight ontology for an Information Space Model (ISM). The ISM is designed to support functionality that lies between data catalogues and domain ontologies. Similar to data catalogues, the ISM provides metadata for data discovery across multiple, heterogeneous (often legacy) data sources e.g. maps servers, satellite images, social networks, geospatial blogs. Similar to domain ontologies, the ISM describes the functional relationship between these systems with respect to entities relevant to an application e.g. venues, actors and activities. We suggest a minimal set of ISM objects, and attributes for describing data sources and sensors relevant to data integration. We present a number of statistical relational learning techniques to represent and leverage the combination of deterministic and probabilistic dependencies found within the ISM. We demonstrate how the ISM provides a flexible language for data integration where unknown or ambiguous relationships can be mitigated.

Keywords: Data integration, data fusion, assignment problem, statistical relational learning

1. INTRODUCTION

Data integration and fusion are challenging user requirements, since the data and the tools required to make decisions are typically scattered across multiple systems and networks. To meet these requirements, users need to be able to discover data and tools, and to synthesize intelligence products by taking into account the quality, reliability, and idiosyncrasies of particular sources and tools within the context of analysis priorities. This is an overwhelming task, even for the most experienced user, and it is only getting more difficult as the number and complexity of sources and tools grow.

In this paper we outline a framework to address some of these challenges, based on Information Space Models (ISMs): light-weight descriptions of data sources and tools that enable scalable and probabilistic data integration. To motivate our approach we present a simple domain model relevant to nonproliferation in Fig. 1. This domain model defines three main object types, and defines relationships between these objects. The *Facility* object is associated with a geo-graphic location such as a place, factory or building. The *Shipment* type is associated with transportation of people, information, or materials between *Facilities* (a shipment typically involves a person or a vehicle). The *Enterprise* object is associated with known commercial, industrial or academic activities that might involve multiple *Shipments* and/or multiple *Facilities*.

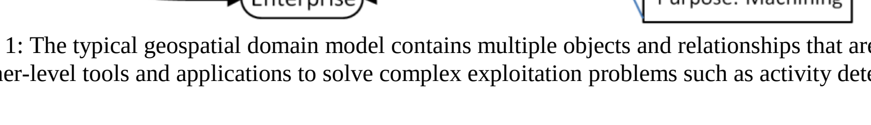


Fig. 1: The typical geospatial domain model contains multiple objects and relationships that are used by higher-level tools and applications to solve complex exploitation problems such as activity detection.

* rporter@lanl.gov

Each of these objects has a set of attributes, as well as a set of relationships to the other objects. There are many different data sources that can be used to populate attributes for each object. Data related to *Shipments* and *Facilities* is often obtained from remotely sensed imagery. Data related to *Enterprises* might come from newspapers and other open media sources. Often data exploitation tools and algorithms aim to exploit one attribute of one object type. For example, accurate detection of material and chemical observables can provide information about the processes occurring within industrial facilities. The goal of data fusion is to support users in combining multiple attributes from multiple objects so that more complex patterns of interest can be detected.

One of the significant challenges in exploiting multi- data source patterns is entity resolution. For example, in Fig. 2, two pieces of data are obtained from two different sensors that suggest that a *Shipment* has occurred between *Facilities* A and B. Recognizing that these two pieces of data are associated with the same real-world entity is potentially very important, however with additional information, it is revealed to be a false alarm. To detect complex multi- data source patterns, we must provide robust and dynamic mechanisms to deal with entity resolution.

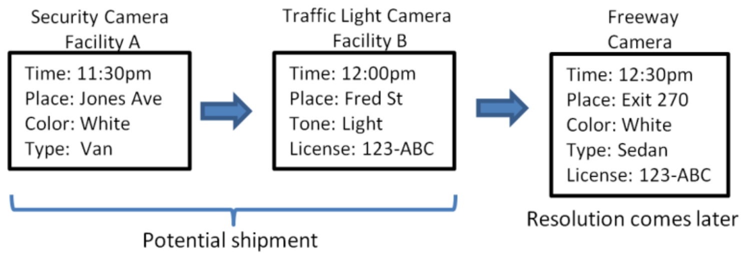


Fig. 2: Example of the entity resolution challenge faced when integrating multiple heterogeneous data sources. This example is inspired by an example in [1].

Our approach is to build a statistical model of the information space that describes how data sources connect to the domain model. There are two unique aspects to our approach:

1. Our models focus on the characteristics of the data sources that directly influence entity resolution.
2. We maintain a clean separation between the information space and the domain space.

The combination leads to what appears to be a missing component in fusion research: an application independent framework for collective inference that supports entity resolution from heterogeneous data sources.

2. BACKGROUND

2.1 Making data user discoverable

One motivation for the Information Space Model (ISM) is to provide users with a comprehensive, transparent view of the data and tools available to be exploited in a given time and place, for a given purpose. In the geospatial domain this is often called a data catalogue [2] and it uses registries of meta-data (potentially distributed across the network of resources) to list detailed information about:

- The data sources that are available, relevant, and reliable
- The types of data available from each source
- The tools and/or applications available for analyzing the data

In our work we use the term Information Space Model instead of data catalogue because we generalize and extend this idea in several key directions. First, we suggest the information space model should reference as much information as possible: use studies, nuances, tidbits and tricks of the trade accumulated by users of that data over time. Examples of this richer, more functional description of data sources might include:

- The primary uses for the data
- The form in which the data will be used
- The tools available for converting the data to that form
- The human resources available to analyze the data, if needed
- Access restrictions and security policies pertaining to the data sources, the data, and its use.

Second, we generalize the ISM to enable automatic discovery and integration of data sources. This second direction is the primary focus of this paper.

2.2 Making data machine discoverable

Data catalogues are invaluable resources for users, however, the catalogue becomes difficult to navigate, search and maintain as the number and frequency of new data and tools grows. Much work therefore focuses on enriching representations to automate aspects of data discovery, and enable the composition of tools and services. Over the past 10 years, a large number of standards have been proposed. Many of these standards are not specific to geospatial data infrastructures, but almost all, could be or have been applied in geospatial domains.

From the business processes community, orchestration languages such as BPEL have been proposed that enable chains of Web Service Description Language (WSDL) services to be deployed with an orchestration engine. The semantic web community is also developing specifications for service descriptions (SAWDL, WSDL-S, WSDL 2) [3], and process descriptions (WSMO, WSML) [4]. A light-weight approach for describing services, and exposing those descriptions via REST services has also been proposed [5].

There are many different standards because automating discovery and execution is a hard problem. It is difficult to describe a service and what it does semantically, and shoehorn that into existing standards. In addition, the approach has to be sufficiently expressive to support reasoning and use of rules to select services based on certain criteria, without being so complex that it becomes computationally impossible to integrate services into a larger task required by an unknown requester [6].

2.3 Data integration

A second approach to automating the exploitation of multiple heterogeneous data sources is data integration. In our work, we define a single global interface through which users (and tools) will access data sources. This approach has the advantage that it hides the complexities associated with data sources and tool chains from the user. The disadvantage compared to techniques described in the previous section is that the system services are fixed and defined ahead of time. As more data sources and tools come online, these services can improve, but any new services will require manual extension of the global interface.

Historically, data integration techniques have focused on databases and were developed as an alternative to data warehousing that could provide lower-cost integration of legacy systems. There are a number of ways database schema's can be mapped to a global, integrated, schema [7]. There are also a number of formal languages that have been proposed to describe this mapping [8], [9]. Traditionally, this mapping is performed manually.

Given that a global interface for multiple databases can be derived, there are still many challenges to robust data integration. Entity matching, entity resolution, data duplication are all examples of a fundamental problem in data integration: identifying what data corresponds to what real-world entity. This problem arises in single datasets e.g. duplicate records. The problem becomes increasingly difficult (and important) as the number and diversity of data sources increases [10]. This is because in addition to data ambiguity (e.g. misspelled names), there is ambiguity in the mapping between data sources and the global interface.

2.4 Ontology matching

In recent years, a large body of work related to data integration has been developed for ontology matching. In this case, the data sources are typically not databases, but more general graph-based data structures. Due to the large number and variety of ontologies that have been developed for various purposes, there is a large amount of ambiguity in how one ontology maps to another ontology, and this has motivated research into methods that can automatically resolve this ambiguity. Recent work in ontology matching draws from the combined toolkit of probabilistic and logical methods [11], [12]. State-of-the-art techniques in this area aim to simultaneously match the data instances (entity resolution) and the ontology to obtain better performance [13].

3. INFORMATION SPACE MODELS

The Information Space Model (ISM) can be considered a *local-as-view* data integration system [14]. We define a common interface which we call the domain model (as did [9]) and then describe how to access each data source through this interface (the ISM). This approach hides the content and access details of any number of heterogeneous data sources and / or tool chains from the user. Unlike traditional data integration, we draw from the combined statistical and relational toolkits and propose a data integration strategy motivated by entity matching. Unlike most approaches to ontology matching, we propose a clear distinction between models of data-sources (the ISM) and models of the application domain (the domain model). We suggest this will help mitigate the need for per-case integration efforts and lead to a more scalable approach to data integration.

Fig. 3 provides a conceptual overview. The ISM is defined in terms of data instances (observations), and data sources. The domain model is defined in terms of entity types, and entity instances (real world objects).

For example, one data source (S^1) might be a red-light traffic camera, which provides observations for a specific vehicle, such as the vehicles' license plate, and approximate vehicle tone (the traffic light camera may only record gray-scale images). A second data source (S^3) might be a parking lot security camera, which observes a vehicles color, make and model (estimated with close-to-the-sensor vehicle recognition algorithms).

The objective of the ISM is to describe data sources (and tools) in sufficient detail to populate a domain model. The domain model represents different entity types (e.g. T^3 = vehicle) and uniquely identifies the real-world entities (e^3 corresponds to a unique vehicle). Multiple data instances, generated from multiple heterogeneous data sources, can be associated with a single entity (illustrated with the dashed lasso that links the ISM to the domain model).

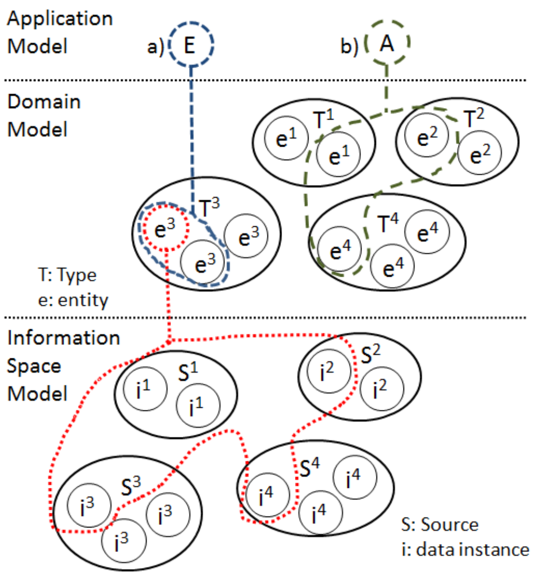


Fig. 3: Conceptual overview of how the ISM interfaces to domain models and applications.

Given a domain model, applications and analysis tools access and integrate data further. A common application is enhanced entity resolution (Application Model a). In this case, entity resolution would leverage the dependencies between the objects within the domain model to improve accuracy. Many recently proposed methods for collective entity resolution may be applicable [15]. A second application (Application Model b) may work directly with the domain model to exploit more complicated patterns, e.g., looking for multi-vehicle meetings and coordinated driving patterns. We do not consider applications of the domain model further in this paper. However readers interested in the vehicle activity detection problem that we use as an example may be interested in our review of wide-area motion imagery exploitation [16].

3.1 ISM template ontology

The ISM is a light weight description of how data sources and tools relate to the domain model. This approach is similar to the template ontology proposed in [17] for bio-informatics data fusion. Similar to their proposal, we expect our model to be extended and adapted to particular situations and applications. In this paper we restrict our attention to fairly simple mappings between data sources and domain models, and focus on model parameters most relevant to ISM based entity resolution. The main objects in the ISM are illustrated in Fig. 4 and described in the next few paragraphs:

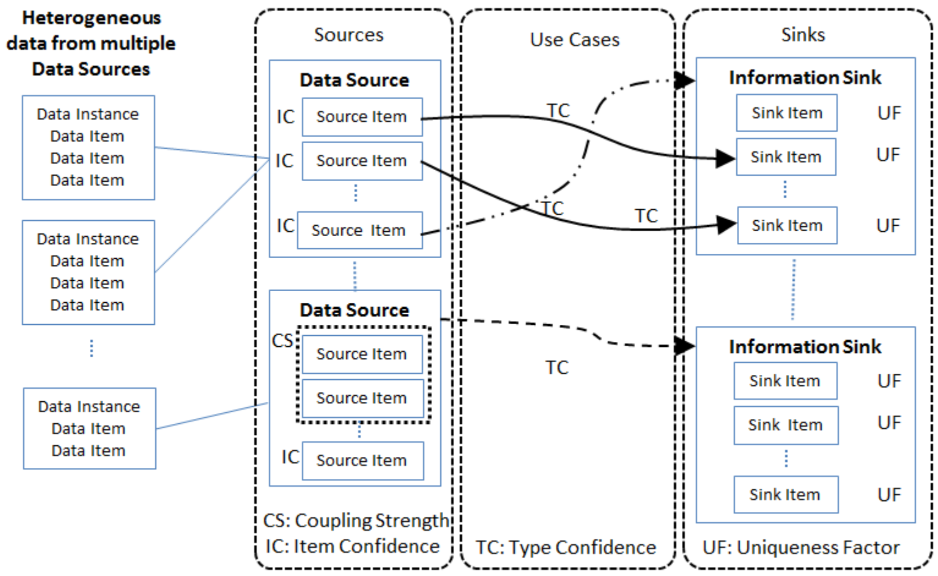


Fig. 4: A minimal set of objects that define an Information Space Model.

Sources: A *Data Source* is an abstract type for describing the context of *Data Instances*. *Data Sources* could be used to describe databases or sensors, they could support streaming or query based modes of operation. The primary role of the *Data Source* is to characterize dependencies between specific pieces of data that are co-collected, or related in some way. For example, a red light camera collects several *Data Items* in one observation: a time stamp, a GPS location, a license plate number. In another example, a database query returns multiple fields of data in each record. Each *Data Item* is uniquely identified by a *Source Item* type. The dependencies between *Source Items* are characterized by the *Data Source*.

In some applications, the dependencies are as simple as a *Coupling Strength* which indicates how likely *Source Items* are associated with the same real world entity. In this paper, we will restrict ourselves to the simple case when *Source Items* defined within a *Data Source* are 100% coupled. This means *Data Sources* define indivisible groups as far as entity resolution is concerned.

Another common use of the *Data Source* is to define attributes that are common across all *Data Instances*. For example, the license plate extraction algorithms used to estimate license plates from red light cameras may have a known accuracy. This would be included within the *Data Source* description as an *Item Confidence*.

Sinks: *Sink Items* define the unique pieces of the domain model through which the ISM will interface. *Sink Items* are grouped into *Information Sinks*. *Information Sinks* are typically associated with domain model objects e.g. vehicles, facilities etc., but this is not necessary. An important attribute of *Sink Items* is the *Uniqueness Factor*. This captures how important that Sink Item is for comparing Data Sources. For example, a *Sink Item* corresponding to a license plate would have a high *Uniqueness Factor* since the data value is meant to be unique to each real world entity. However the *Uniqueness Factor* for the vehicle color would be low since many real world entities have the same color. The main motivation for representing the domain model with *Sink Items* and *Information Sinks* is to facilitate mapping of *Data Sources* through *Use Cases*.

Use Cases: *Data Sources* are linked to the *Sinks* through the *Use Cases*. It is possible that the *Use Case* could build on business process orchestration languages, or related semantic services languages, to model chains (or processes) of data sources and tools. In this paper we only consider a number of simple mappings: a) links between *Source Items* and *Sink Items* (the solid arrows in Fig. 4), b) links between *Source Items* and *Information Sinks* (the dot-dash arrow in Fig. 4), and c) links between *Data Sources* and *Information Sinks* (the dashed arrow in Fig. 4). It some situations, users may have an estimate for how certain they are in each mapping. We call this attribute a *Type Confidence*. Another way for users to represent uncertain mappings is to map a *Source Item* to multiple *Sink Items*. This case will be investigated in detail in Section 5.

3.2 ISM based entity resolution

We imagine a *query* begins with a request for particular items in the domain model. For example, in Fig. 5, a user requests all vehicle license plates observed over a particular spatial area and temporal period. The ISM contains all the *Data Sources* that reference the license plate *Sink Item*. The *Data Instances* associated with these *Data Sources* contain all the relevant data. However we do not know which *Data Instances* are associated with which vehicles: there could be multiple observations (*Data Instances*) of the same vehicle from one *Data Source*, and / or, multiple views (*Data Instances*) of the same vehicle from different *Data Sources*. We will describe several different ways to determine which *Data Instances* belong to which real-world entity. All of these are based on measuring a similarity between *Data Instances*, and the ISM provides the framework to determine how this similarity is calculated.

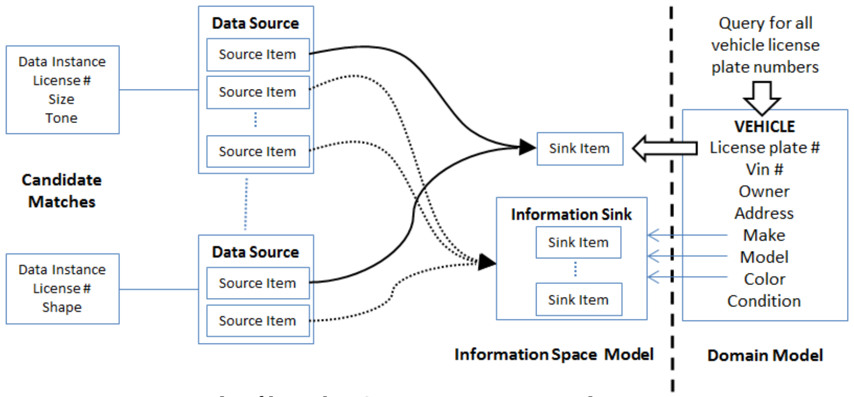


Fig. 5: Example of how the ISM supports entity resolution.

4. UNCERTAINTY IN DATA VALUES

In some cases the ISM can be defined in sufficient detail to make similarity calculations straightforward. That is, *Source Items* that map to the same *Sink Item* can be compared. In Fig. 5 the two *Data Sources* share the *Sink Item* associated with the vehicle license plate number. A similarity score is calculated using a combination of value comparisons (*Data Item* to *Data Item*), and confidence parameters (described in the *Data Source*). If the confidence and uniqueness parameters described in Fig. 4 are scalars, then a reasonable similarity score for two *Data Items* u, v associated with *Source Items* i, j would be:

$$Similarity(u, v) = UF_t . TC_i . TC_j . IC_u . IC_v . Match(u, v) . \quad (1)$$

The function $Match(u, v)$ depends on the data. For example, it may be a string edit distance if u, v are strings, or a Euclidean distance if they are numbers. The best match function for each source item can be specified in the *Data Source* description. In more sophisticated systems, this match function might have to be automatically discovered or inferred.

Matching two *Data Instances* through a single *Sink Item* may be possible if the *Sink Item* has a high *Uniqueness Factor*. But in most cases, supporting evidence will be required. The coupling of *Source Items* within the *Data Source* defines this additional evidence. For example, in Fig. 5 additional *Source Items* within each *Data Source* map to the same *Information Sink* (illustrated with dotted arrows). We calculate the total similarity of two *Data Instances* as the sum of similarities of all comparable *Source Items*:

$$EntitySimilarity(u, v) = \sum_{\{u,v\}\ where\ TypeMatch(i,j)} Similarity(u, v) \quad (2)$$

where $TypeMatch(i, j)$ indicates that *Source Items* map to a common *Sink* identifier, and are therefore comparable.

We now consider two general solution methods for entity resolution. The first treats each candidate pair of *Data Instances* independently and simply thresholds the similarity score defined in Eq. 2. If the similarity is above threshold, then the two candidates are considered to be part of the same entity, or *matched*.

$$EntityMatch(i, j) = \begin{cases} 1 & if\ EntitySimilarity(i, j) > t \\ 0 & otherwise \end{cases} . \quad (3)$$

A second solution method uses collective classification. This means we introduce additional constraints between candidate pairs such as transitivity:

$$EntityMatch(A, B)\ and\ EntityMatch(B, C) \Rightarrow EntityMatch(A, C) \quad (4)$$

Collective classification constraints such as Eq. 4 require more sophisticated machinery to optimize. We use Markov Logic Networks (MLN) [18]. MLN provide a unified framework for describing Markov Random Fields with probabilistic (Eq. 2) and deterministic (Eq. 4) dependences. In our experiments we used the Alchemy software implementation of MC-SAT provided by the University of Washington to generate probability estimates for Eq. 3.

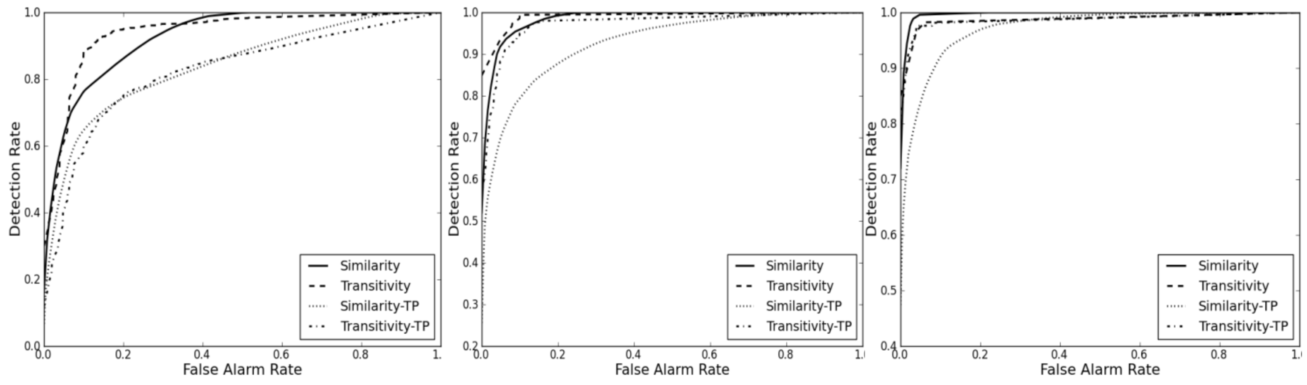


Fig. 6: Entity resolution with and without transitivity constraints for different *Data Source* overlaps. In all experiments each of 5 sensors observes Left) 3, Middle) 4, and Right) 5 of the 10 attributes.

We implemented a number of synthetic experiments to investigate the two approaches. We generate G ground truth entities, where each ground truth entity has N *Sink Items*. We then generate S *Data Sources*. Each *Data Source* is assigned (randomly) M of the N *Sink Items*. We then generate $G * S$ *Data Instances* by allowing each *Data Source* to observe the M *Sink Items* for each ground truth entity. The entity resolution methods consider every possible pair of instances: $0.5(G * S)^2$ candidates.

In Fig. 6, we compare Eq. 3, with and without Eq. 4 enforced. In all cases, the number of ground truth entities and the number of *Data Sources* was 5. The ground truth consisted of 10 attributes, of which each *Data Source* collected data on 3 (on the left), 4 (in the middle) and 5 (on the right). Attributes are strings drawn from alphabet of 10 values, and the match function is simple equivalence: 1 if the strings match and -1 if they do not. The results in Fig. 6 are averaged over 20 trials.

As the number of observed attributes increases from 3 on the left to 5 on the right, there is increasing overlap between *Data Source* observations (more terms in Eq. 2), and the accuracy of all methods improves. *Similarity* corresponds to Eq. 3 and *Transitivity* corresponds to the MLN solution with Eq. 4 enforced. The TP variants correspond to the situation when the *TypeMatch* function is predicted (and therefore error-prone). We discuss the details of how this function is predicted in Section 5. We observe that in both cases, transitivity constraints generally improve performance. We also observe that for transitivity to be effective there needs to be sufficient overlap between *Data Source* observations. On the left *Data Sources* are linked with less than 3 attributes, and when *TypeMatch* is predicted (TP), transitivity offers little improvement. However, as the number of linked attributes increases the additional constraints can help mitigate some of the *TypeMatch* uncertainty.

5. UNCERTAINTY IN DATA TYPES

Information Sinks provide a global interface for multiple *Data Sources*. So far we have discussed situations where *Data Items* are manually mapped to specific *Sink Items*. However, this is often difficult, and sometimes, it may not even be possible. For example, in Fig. 5, our two *Data Sources* generate Size, Tone and Shape attributes that have not been defined in the domain model. To support this situation, we introduced *Information Sinks* – groups of *Sink Items* that are known to be related in some way. Multiple *Source Items* can be mapped to the same *Information Sink*, as illustrated by dotted lines in the Fig. 5. Entity resolution, and other tasks, are now much more challenging because we do not know the *TypeMatch* function in Eq. 2. In this section we investigate solutions to this problem.

We will consider the extreme case: when all *Source Items* are assigned to the same *Information Sink*. This means any *Source Item* in the first *Data Source* could be comparable (or not) to any *Source Item* in the second *Data Source*. We investigate a two-stage approach: 1) we predict which *Source Items* in which *Data Sources* are comparable, and 2) we perform entity resolution with Eq. 3 to evaluate performance.

5.1 Transitivity in types

One way to resolve the types would be to use an approach very similar to the one we used in Section 4. For each two *Data Sources*, we measure the (type-) similarity between all potential *Source Items*. A natural similarity score for *Source Items* is the sum of instance similarity scores across the dataset:

$$TypeSimilarity(i,j) = \sum_{\{u,v\}\ asscoated\ with\ \{i,j\}} Similarity(u,v) \tag{5}$$

Analogous to method 1 in Section 4, we threshold Eq. 5 to determine which *Source Items* in *Data Source* 1 should be matched to which *Source Items* in *Data Source* 2:

$$TypeMatch(i,j) = \begin{cases} 1 & if\ TypeSimilarity(i,j) > t \\ 0 & otherwise \end{cases}. \tag{6}$$

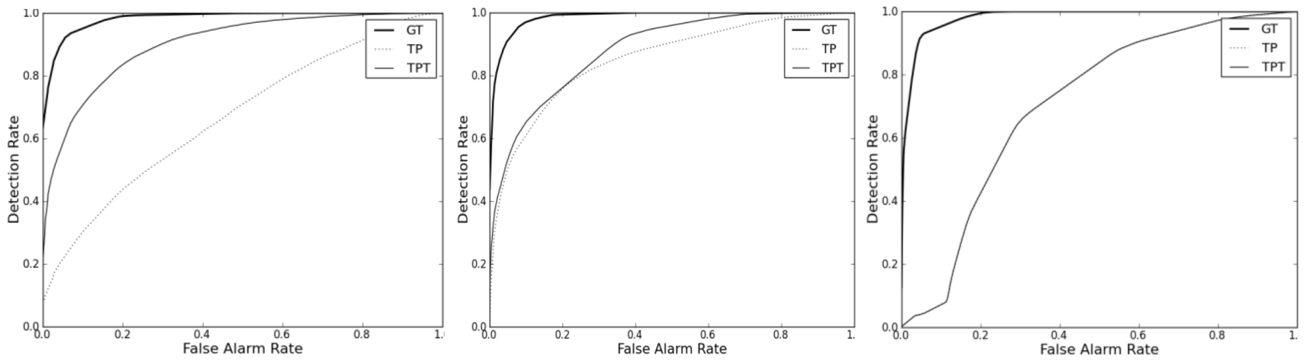


Fig. 7: Entity resolution performance using type prediction (TP) with and without transitivity constraints (TPT), compared against ground truth (GT) for different entity / source regimes. Left) 2 entities and 12 sources, Middle) 4 entities and 12 sources, and Right) 12 entities and 12 sources.

We use these predicted matches in Eq. 2 and measure performance in terms of entity resolution as before. Analogous to method 2, in Section 4, we can also consider constraints between *TypeMatches*, and specifically the transitivity constraints defined by Eq. 4. We implemented a number of experiments similar to Section 4 to evaluate the benefit of these additional constraints and summarize the results in Fig. 7.

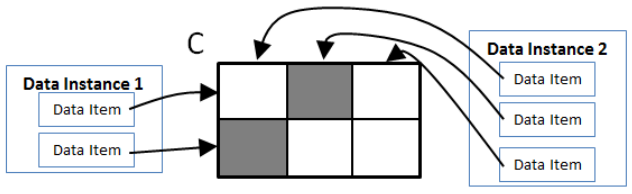
In all cases, there were 12 *Data Sources*, each observing 4 attributes of the 10 ground truth attributes. To generate the type predictor (TP) and type predictor with transitivity (TPT) curves we must choose the threshold in Eq. 6. In this experiment we evaluated all possible thresholds and selected the threshold with the best ROC curve. On the left, there are two ground truth entities leading to 24 *Data Instances*. In this case, the transitivity constraints are seen to greatly help the type predictors. However as the number of ground truth entities increase to 3 (in the middle) the additional value diminishes and in fact by 5 the performance of the classifiers is identical. On the right there are 12 truth entities and the performance of the classifiers is very poor (equivalent to setting Eq. 6 to 1 for all possible type combinations). We suggest that the poor performance of the type predictors is because the sum in Eq. 5 is dominated by terms that are not correct matches. Transitivity is not enough, and further constraints are required.

5.2 A typed assignment problem

In this section we introduce exclusivity constraints: any *Source Item* in one *Data Source* should be matched to at most one *Source Item* in the second *Data Source*. This constraint can be formulated as an assignment problem: given a cost (or profit) associated with each combination of *Source Items*, we find the assignment that minimizes (or maximizes) the cost, subject to the exclusivity constraints. Note, that unlike the transitivity constraints explored in the last section, the assignment problem can be solved in polynomial time with a number of different algorithms such as the Hungarian algorithm and Linear Programming.

The assignment problem is used widely in data fusion and tracking. Typically, it is used to match two (or more) sets of observation. For example, two sets of moving object detections observed at two consecutive points in time. In our type matching application, we would like to find an assignment between multiple sets of *Data Sources*, and in addition, we have multiple examples (instances) of each *Data Source*. This suggests a number of variations on how the assignment problem is used, which we collectively call a Typed Assignment Problem (TAP).

The TAP is illustrated in Fig. 8. On the left we illustrate the typical assignment problem, where two sets of observations (obtained from different data sources) are matched. A cost matrix, C, is formed, where each element of the matrix is a cost associated with assigning specific combinations of *Data Items*. The assignment problem finds the lowest cost matching subject to the constraint that only one match can appear in each row and column. In Fig. 8 we illustrate a hypothetical matching with shaded squares.



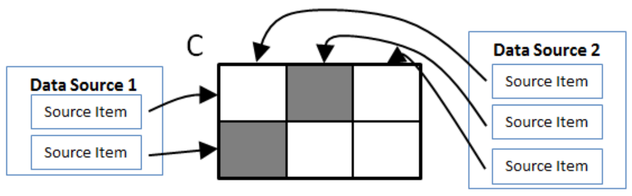


Fig. 8: Left) The typical Assignment Problem associates sets of *Data Items*. and Right) The Typed Assignment Problem associates *Source Items* (data types) within each *Data Source*.

On the right in Fig. 8 we show an alternative assignment problem. In this case we attempt to match *Source Items* associated with each *Data Source*, instead of *Data Items* associated with *Data Instances*. We investigate three different methods.

5.3 Instance method (TAP 1)

Method 1 is the most similar to the standard assignment problem on the left in Fig. 8. We consider the $\boldsymbol{i^{th}}$ and $\boldsymbol{j^{th}}$ instances of data, $\boldsymbol{x^s(i), y^t(j)}$, obtained from two data sources, $\boldsymbol{s, t}$. Each instance has a number of *Data Items* which we denote:

$$x^s(i) = \{\, x_1^s(i), x_2^s(i), \dots, x_K^s(i)\},\qquad (7)$$
$$y^t(j) = \{\, y_1^t(j), y_2^t(j), \dots, y_L^t(j)\}.$$

In method 1 we solve the assignment problem for every candidate pair of data instances independently. This means we form a $\boldsymbol{K\ by\ L}$ cost matrix where:

$$C[k,l] = s\big(x_k^s(i), x_l^t(j)\big) \qquad (8)$$

where $\boldsymbol{s(i,j)}$ is defined by Eq. 1. To account for cases where *Data Items* in one instance are not comparable to *Data Items* in the second instance, we augment the cost matrix in the usual way: we add $\boldsymbol{K}$ additional columns to the matrix that correspond to the "no match" option. The cost of the "no match" is a free parameter. In our experiments we set it to 0 (our similarity score produces $\{\boldsymbol{1, -1}\}$). After solving the assignment problem we calculate the total cost and use this as the *EntitySimilarity* in Eq. 2. For $\boldsymbol{N}$ *Data Instances* we solve a total of $\boldsymbol{\frac{1}{2}(N)^2 - N}$ assignment problems.

5.4 Average method (TAP 2)

Method 2 takes advantage of the fact that the assignment between two *Data Sources* does not change between *Data Instances*. We therefore solve a single assignment problem for each source pair, where each cost is accumulated across the entire dataset:

$$C[k,l] = \sum_{i,j\, \epsilon\, s,t} s\big(x_k^s(i), x_l^t(j)\big) \qquad (9)$$

The optimal assignment is used as the *TypeMatch* function in Eq.2. Note, this approach avoids having to choose the threshold in Eq. 6, but the cost of the "no match" assignment plays a more critical role. In TAP1, the cost is chosen with respect to the *Similarity* score between two pieces of data. In TAP2, this cost is a function of the number of assignments we expect over the entire dataset. We suggest that this prior knowledge is typically much harder to provide. For *S Data Sources* we solve a total of $\frac{1}{2}(S)^2 - S$ assignment problems.

5.5 Voting method (TAP 3)

Our third method is motivated by the fact that for two *Data Sources*, not all instances correspond to the same entity. This means we are often trying to match *Source Items* where there is little or no similarity. In Section 5.1. we hypothesized that this was why the type predictors performed so poorly. Similar to TAP2 we propose to develop a single assignment for each *Data Source* pair. But, instead of assigning the cost based on Eq. 9, we use a weighted vote of assignments from each instance. That is, similar to TAP1, we solve the assignment problem in Eq. 8 for each *Data Instance* pair. The solution (or assignment) is a $K\ by\ L$ binary matrix $with$ elements $A(k,l) = 1$ corresponding to the optimal assignment,

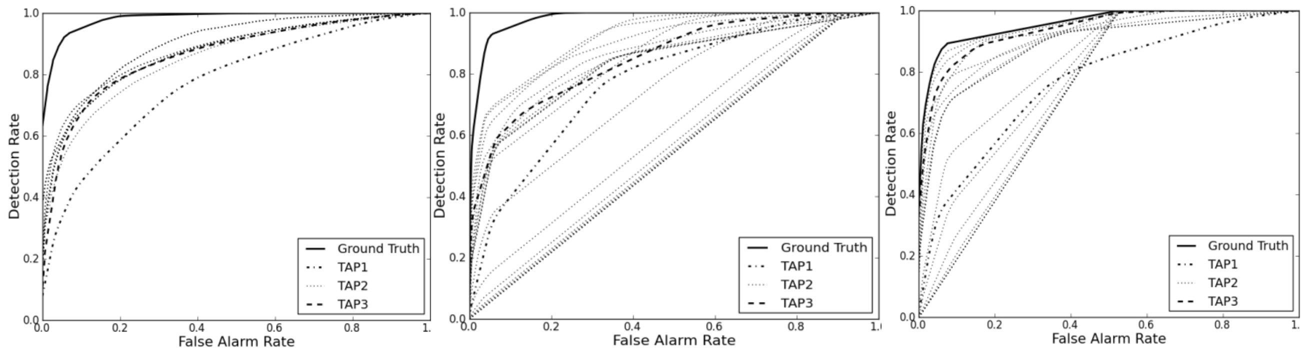


Fig. 9: A comparison of the Typed Assignment Problem (TAP) in 3 different entity / source regimes: Left) 2 entities observed by 12 sources, Middle) 12 entities observed by 12 sources and Right) 12 entities observed by 2 sources.

and a total cost for the assignment, which we denote $c_{i,j}$. Similar to TAP2, TAP3 accumulates a cost matrix across the dataset:

$$C[k,l] = \sum_{\forall\, i,j} c_{i,j}.A(k,l) \tag{10}$$

We calculate a final assignment with this new cost matrix, and use the solution in place of the *TypeMatch* function in Eq. 2. Note, that the cost of “no match” in TAP3 is similar to TAP1. For N *Data Instances* we solve a total of $\frac{1}{2}(N)^2 - N + 1$ assignment problems.

We performed another set of experiments to investigate the three approaches. On the left in Fig. 9 we have 2 ground truth entities observed by 12 *Data Sources*. *Data Sources* observe 4 out of 10 attributes and the results in Fig. 9 are averaged over 20 independent runs. *Ground Truth* corresponds to Eq. 3 calculated with a known *TypeMatch* function. Since the cost of the “no match” is more difficult to specify in method TAP2, we evaluated the method at 20 evenly spaced thresholds between -1 and 1. In the first experiment we observed very little difference between the TAP2 solutions and the TAP3 solution, and the TAP2 solution showed small variance as the “no match” cost was varied. We attribute the similar performance to the fact that there are a small number of *Data Instances* for each *Data Source* pair, and of these matches 50% are correct. This means that the accumulated cost matrix used in TAP2 is very close to the instance based cost matrix when the matches are correct.

In Fig. 9 Middle, there are 12 ground truth entities observed with 12 *Data Sources*. This is the same problem used in Fig. 7 Right, where the type predictors performed very poorly. In Fig. 9 Middle, the TAP methods do a lot better. In this case, the cost of the “no-match” affects the performance of TAP2 significantly. There are values of this parameter that produce the best performance out of all methods, and values that produce the worst performance. Again, TAP3 outperformed TAP1,

In Fig. 9 Right, we investigate the opposite regime: there are only 2 *Data Sources* through which we observe 12 ground truth entities. We observed a similar pattern of performance to Fig. 9 Middle, but in this case all methods generally do better, as we would expect, since there are more examples with which to estimate type matches.

6. DISCUSSION

We have presented a number of techniques for mitigating value and type uncertainty in the ISM. We found that transitivity constraints help mitigate the partial overlap of *Data Sources* during entity resolution. We found that exclusivity constraints help mitigate the type uncertainty within the ISM, and, suggested a number of solution methods based on the assignment problem. The TAP 2 method can potentially provide good performance, and it only requires a single assignment problem to be solved. However, it requires judicious choice of the “no match” cost and this may be difficult in some applications. The TAP3 method is more expensive computationally, but does appear to provide a robust alternative to TAP2, if the per-instance “no-match” cost is easier to define. There are many additional mechanisms for mitigating uncertainty left to explore and many avenues to extend the ISM framework. In the next few sections we discuss some of these directions.

6.1 Collective classification

In this paper we explored a number of collective classification constraints including transitivity and exclusivity. We found transitivity improved entity resolution, but was less important than exclusivity for type matching. A useful third experiment would be to evaluate the additional benefit of transitivity, given exclusivity is enforced. The Typed Assignment Problem methods we developed in this paper all run in polynomial time, whereas the transitivity constraints are NP hard and require approximate solutions, which are often based on Monte Carlo methods [18]. Comparing exclusivity to exclusivity plus transitivity would help determine if the additional computational cost was warranted. If it is, a possible path to more efficient solution methods is to think of transitivity constraints as a multi-dimensional (or multi-index) assignment problem [19]. Although this problem is also NP hard, several different approximate algorithms have been suggested [20] and some these may be appropriate for the Typed Assignment Problem.

6.2 Learning

There are many places where data could be used to learn ISM parameters. This reduces the effort required from users to specify the ISM, and helps improve performance by tailoring the ISM to the data. For example, a very common parameter to optimize in entity resolution is the *Uniqueness Factor*. This would be similar to the approach used for geospatial name matching [21], where a SVM was used to weight similarity scores of different attributes. Another approach, explored in [22], used learning to tailor the similarity score itself. Learning could also be used in the Type Prediction problem explored in Section 5 to find better costs for the assignment problems.

A second major role of learning will be to help map ambiguous *Data Items* to *Information Items* so that they can be accessed explicitly by users. In this paper we developed methods to identify which *Source Item* in one *Data Source* could be compared to which *Source Item* in another *Data Source*. However this does not provide a meaningful identifier for users to query the data. User interaction is required to make this mapping, and learning, may be able to help reduce this interaction.

6.3 Geospatial specializations

In this paper we have discussed very generic models for data integration where all object attributes are treated equally. However in the geospatial domain we have the opportunity to exploit the specific properties of spatial and temporal attributes, and these attributes often provide critical constraints for entity resolution, i.e., the location and time of observations typically has a very high *Uniqueness Factor*. We suggest that most geospatial specializations can be included in our framework through judicious choice of match functions and parameters. However it is also likely that an ISM implementation for the geospatial domain would benefit from specialized indexing of the spatial and temporal attributes [23].

7. SUMMARY

We have outlined a framework for Information Space Models: a light weight description of data sources and tools for data integration and entity resolution. We have explored a number of techniques for mitigating uncertainty in the ISM based on collective classification constraints such as transitivity and exclusivity. We developed novel solution methods for exclusivity that run in polynomial time, based on the assignment problem.

REFERENCES

[1] Jonas, J. "Privacy by Design: Confessions of an Achitect," Presentation]. Available from: jeffjonas.typepad.com/PbD-ConfessionsOfAnArchitect.ppt, (2011).

[2] Nebert, D.D., "Developing Spatial Data Infrastructures: The SDI Cookbook." Global Spatial Data Infrstructure, (2004).

[3] W3C. "Semantic Annotations for WSDL and XML Schema," Available from: http://www.w3.org/TR/sawsdl/,

[4] W3C. "WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web," Available from: http://www.w3.org/Submission/2010/SUBM-WSMO-Lite-20100823/,

[5] Gessler, D., et al., "SSWAP: A Simple Semantic Web Architecture and Protocol for semantic web services," BMC Bioinformatics, **10**(1): p. 309, (2009).

[6] James E. Powell, Krista Black, and L.M. Collins, "A Semantic Registry for Digital Library Collections and Services," D-Lib Magazine, **17**(11/12), (2011).

[7] Batini, C., M. Lenzerini, and S.B. Navathe, "A comparative analysis of methodologies for database schema integration," ACM Comput. Surv., **18**(4): p. 323-364, (1986).

[8] Lutz, M., et al., "A rule-based description framework for the composition of geographic information services," in *Proceedings of the 2nd international conference on GeoSpatial semantics*. Springer-Verlag: Mexico City, Mexico. p. 114-127, (2007).

[9] Lutz, M. and D. Kolas, "Rule-Based Discovery in Spatial Data Infrastructure," Transactions in GIS, **11**(3): p. 317-336, (2007).

[10] Dey, D., S. Sarkar, and P. De, "A Probabilistic Decision Model for Entity Matching in Heterogeneous Databases," Manage. Sci., **44**(10): p. 1379-1395, (1998).

[11] Vaccari, L., et al., "An evaluation of ontology matching in geo-service applications," Geoinformatica, **16**(1): p. 31-66, (2012).

[12] Noessner, M.N.J., C. Meilicke, and H. Stuckenschmidt. "Probabilistic-Logical Web Data Integration," in *In Reasoning Web: 7th International Summer School 2011*. Galway, Ireland: Springer-Verlag, (2011).

[13] Suchanek, F.M., S. Abiteboul, and P. Senellart, "PARIS: probabilistic alignment of relations, instances, and schema," Proc. VLDB Endow., **5**(3): p. 157-168, (2011).

[14] Levy, A.Y., "Logic-based techniques in data integration," in *Logic-based artificial intelligence*. Kluwer Academic Publishers. p. 575-595, (2000).

[15] Singla, P. and P. Domingos, "Entity Resolution with Markov Logic," in *Proceedings of the Sixth International Conference on Data Mining*. IEEE Computer Society. p. 572-582, (2006).

[16] Porter, R., A.M. Fraser, and D. Hush, "Narrowing the Semantic Gap in Wide Area Motion Imagery," IEEE Signal Processing Magazine, **27**(5): p. 56-65, (2010).

[17] Pincus, Z. and M.A. Musen. "Contextualizing Heterogeneous Data for Integration and Inference," in *AMIA 2003 Symposium Proceedings*, (2003).

[18] Domingos, P. and D. Lowd, "Markov Logic: An Interface Layer for Artifical Intelligence." Synthesis Lectures on Artificial Intelligence and Machine Learning, ed. R.J. Brachman and T.G. Dietterich, (2009).

[19] Burkard, R., M. Dell'Amico, and S. Martello, "Assignment Problems," Society for Industrial and Applied Mathematics, (2012).

[20] Bandelt, H.-J., Y. Crama, and F.C.R. Spieksma, "Approximation algorithms for multi-dimensional assignment problems with decomposable costs," Discrete Appl. Math., **49**(1-3): p. 25-50, (1994).

[21] Sehgal, V., L. Getoor, and P.D. Viechnicki, "Entity resolution in geospatial data integration," in *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*. ACM: Arlington, Virginia, USA. p. 83-90, (2006).

[22] Bilenko, M., "Learnable Similarity Functions and Their Application to Record Linkage and Clustering," in *Department of Computer Sciences*. University of Texas at Austin: Austin, TX, (2006).

[23] Güting, R.H. and M. Schneider, "Moving objects databases," Academic Press, (2005).