Learning Systems of Concepts with an Infinite Relational Model

Charles Kemp and **Joshua B. Tenenbaum**
Department of Brain and Cognitive Science
Massachusetts Institute of Technology
{ckemp,jbt}@mit.edu

Thomas L. Griffiths
Department of Cognitive and Linguistic Sciences
Brown University
{tom_griffiths}@brown.edu

Takeshi Yamada and **Naonori Ueda**
NTT Communication Science Laboratories
{yamada,ueda}@cslab.kecl.ntt.co.jp

Abstract

Relationships between concepts account for a large proportion of semantic knowledge. We present a nonparametric Bayesian model that discovers systems of related concepts. Given data involving several sets of entities, our model discovers the kinds of entities in each set and the relations between kinds that are possible or likely. We apply our approach to four problems: clustering objects and features, learning ontologies, discovering kinship systems, and discovering structure in political data.

Philosophers, psychologists and computer scientists have proposed that semantic knowledge is best understood as a system of relations. Two questions immediately arise: how can these systems be represented, and how are these representations acquired? Researchers who start with the first question often devise complex representational schemes (e.g. Minsky's (1975) classic work on frames), but explaining how these representations are learned is a challenging problem. We take the opposite approach. We consider only simple relational systems, but show how these systems can be acquired by unsupervised learning.

The systems we wish to discover are simple versions of the "domain theories" discussed by cognitive scientists and AI researchers (Davis 1990). Suppose that a domain includes several types, or sets of entities. One role of a domain theory is to specify the kinds of entities that exist in each set, and the possible or likely relationships between those kinds. Consider the domain of medicine, and a single type defined as the set of terms that might appear on a medical chart. A theory of this domain might specify that cancer and diabetes are both *disorders*, asbestos and arsenic are both *chemicals*, and that chemicals can *cause* disorders. Our model assumes that each entity belongs to exactly one kind, or cluster, and simultaneously discovers the clusters and the relationships between clusters that are best supported by the data.

A key feature of our approach is that it does not require the number of clusters to be fixed in advance. The number of clusters used by a theory should be able to grow as more and more data are encountered, but a theory-learner should introduce no more clusters than are necessary to explain the data. Our approach automatically chooses an appropriate number of clusters using a prior that favors small numbers of clusters, but has access to a countably infinite collection of clusters. We therefore call our approach the infinite relational model (IRM). Previous infinite models (Rasmussen 2002; Antoniak 1974) have focused on feature data, and the IRM extends these approaches to work with arbitrary systems of relational data.

Our framework can discover structure in relational data sets that appear quite different on the surface. We demonstrate its range by applying it to four problems. First we suggest that object-feature data can be profitably viewed as a relation between two sets of entities — the objects and the features — and show how the IRM simultaneously clusters both. We then use the IRM to learn a biomedical ontology. Ontologies are classic examples of the theories we have described, since they group entities into higher-level concepts and specify how these high-level concepts relate to each other. Next we show that the IRM discovers aspects of the kinship structure of an Australian tribe. Our final example considers a political data set, and we discover a system with clusters of countries, clusters of interactions between countries, and clusters of country features.

The Infinite Relational Model

Suppose we are given one or more relations involving one or more types. The goal of the IRM is to partition each type into clusters, where a good set of partitions allows relationships between entities to be predicted by their cluster assignments. For example, we may have a single type *people* and a single relation *likes*(i, j) which indicates whether person i likes person j. Our goal is to organize the entities into clusters that relate to each other in predictable ways (Figure 1a). We also allow predicate types: if there are multiple relations defined over the same domain, we will group them into a type and refer to them as predicates. For instance, we may have several social predicates defined over the domain *people* $\times$ *people*: *likes*$(\cdot, \cdot)$, *admires*$(\cdot, \cdot)$, *respects*$(\cdot, \cdot)$, and *hates*$(\cdot, \cdot)$. We can introduce a type for these social predicates, and define a ternary relation *applies*(i, j, p) which is true if predicate p applies to the pair (i, j). Our goal is now to simultaneously cluster the people and the predicates (Figure 1c). The IRM can handle arbitrarily complex systems of attributes, entities and relations: if we include demographic attributes for the people, for example, we can

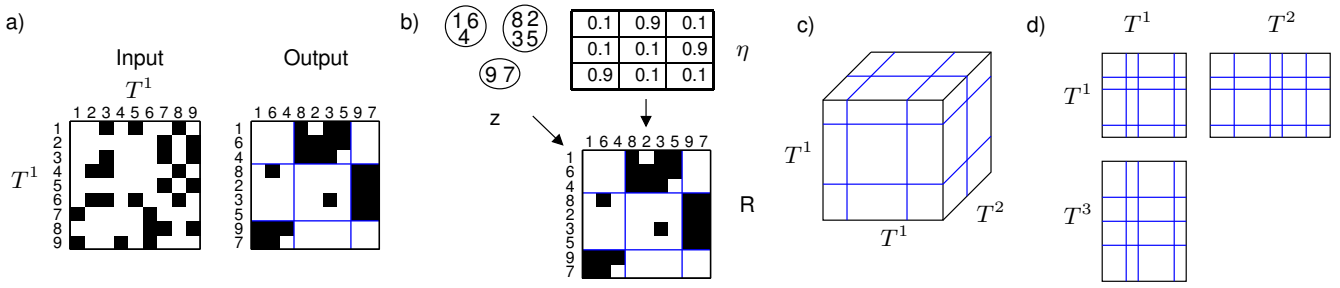

Figure 1: (a) Input and output when the IRM is applied to a binary relation $R : T^1 \times T^1 \rightarrow \{0, 1\}$. The IRM discovers a partition of the entities, and the input matrix takes on a relatively clean block structure when sorted according to this partition. (b) The IRM assumes that relation R is generated from two latent structures: a partition z and a parameter matrix η. Entry $R(i, j)$ is generated by tossing a coin with bias $\eta(z_i, z_j)$, where z_i and z_j are the cluster assignments of entities i and j. The IRM inverts this generative model to discover the z and the η that best explain relation R. (c) Clustering a three place relation $R : T^1 \times T^1 \times T^2 \rightarrow \{0, 1\}$. T^1 might be a set of people, T^2 a set of social predicates, and R might specify whether each predicate applies to each pair of people. The IRM looks for solutions where each three dimensional sub-block includes mostly 1s or mostly 0s. (d) Clustering three relations simultaneously. T^1 might be a set of people, T^2 a set of demographic features, and T^3 a set of questions on a personality test. Note that the partition for T^1 is the same wherever this type appears.

simultaneously cluster people, social predicates, and demographic attributes.

Formally, suppose that the observed data are m relations involving n types. Let R^i be the ith relation, T^j be the jth type, and z^j be a vector of cluster assignments for T^j. Our task is to infer the cluster assignments, and we are ultimately interested in the posterior distribution $P(z^1, \ldots, z^n | R^1, \ldots, R^m)$. We specify this distribution by defining a generative model for the relations and the cluster assignments:

$$P(R^1, \ldots, R^m, z^1, \ldots, z^n) = \prod_{i=1}^{m} P(R^i | z^1, \ldots, z^n) \prod_{j=1}^{n} P(z^j)$$

where we assume that the relations are conditionally independent given the cluster assignments, and that the cluster assignments for each type are independent. To complete the generative model we first describe the prior on the cluster assignment vectors, $P(z^j)$, then show how the relations are generated given a set of these vectors.

Generating clusters

To allow the IRM the ability to discover the number of clusters in type T, we use a prior that assigns some probability mass to all possible partitions of the type. A reasonable prior should encourage the model to introduce only as many clusters as are warranted by the data. Following previous work on nonparametric Bayesian models (Rasmussen 2002; Antoniak 1974), we use a distribution over partitions induced by a Chinese Restaurant Process (CRP, Pitman 2002).

Imagine building a partition from the ground up: starting with a single cluster containing a single object, and adding objects until all the objects belong to clusters. Under the CRP, each cluster attracts new members in proportion to its size. The distribution over clusters for object i, conditioned on the cluster assignments of objects $1, \ldots, i-1$ is

$$P(z_i = a | z_1, \ldots, z_{i-1}) = \begin{cases} \frac{n_a}{i-1+\gamma} & n_a > 0 \\ \frac{\gamma}{i-1+\gamma} & a \text{ is a new cluster} \end{cases}$$

where n_a is the number of objects already assigned to cluster a, and γ is a parameter. The distribution on z induced by the CRP is exchangeable: the order in which objects are assigned to clusters can be permuted without changing the probability of the resulting partition. $P(z)$ can therefore be computed by choosing an arbitrary ordering and multiplying conditional probabilities as specified above. Since new objects can always be assigned to new clusters, the IRM effectively has access to a countably infinite collection of clusters, hence the first part of its name.

A CRP prior on partitions is mathematically convenient, and consistent with the intuition that the prior should favor partitions with small numbers of clusters. Yet it is not a universal solution to the problem of choosing the right number of clusters. In some settings we may have prior knowledge that is not captured by the CRP: for instance, we may expect that the clusters will be roughly equal in size. Even so, the CRP provides a useful starting point for structure discovery in novel domains.

Generating relations from clusters

We assume that relations are binary-valued functions, although extensions to frequency data and continuous data are straightforward. Consider first a problem with a single type T and a single two-place relation $R : T \times T \rightarrow \{0, 1\}$. Type T, for example, could be a collection of people, and $R(i, j)$ might indicate whether person i likes person j. The complete generative model for this problem is:

$$\begin{aligned} z \,|\, \gamma &\sim \text{CRP}(\gamma) \\ \eta(a, b) \,|\, \beta &\sim \text{Beta}(\beta, \beta) \\ R(i, j) \,|\, z, \eta &\sim \text{Bernoulli}(\eta(z_i, z_j)), \end{aligned} \tag{1}$$

where $a, b \in \mathcal{N}$. The model is represented graphically in Figure 1b.

Here we assume that an entity's tendency to participate in relations is determined entirely by its cluster assignment. The parameter $\eta(a, b)$ specifies the probability that a link exists between any given pair (i, j) where i belongs to cluster a and j belongs to cluster b. We place symmetric conjugate priors (with hyperparameter β) on each entry in the η matrix.

To specify the most general version of the IRM, we extend Equation 1 to relations of arbitrary arity. Consider an m dimensional relation R involving n different types. Let d_k be the label of the type that occupies dimension k: for example, the three place relation $R : T^1 \times T^1 \times T^2 \rightarrow \{0, 1\}$ has $d_1 = d_2 = 1$, and $d_3 = 2$. As before, the probability that the relation holds between a group of entities depends only on the clusters of those entities:

$$R(i_1, \ldots, i_m)|z^1, \ldots, z^n, \eta \sim \text{Bernoulli}(\eta(z^{d_1}_{i_1}, \ldots, z^{d_m}_{i_m})).$$

In settings with multiple relations, we introduce a parameter matrix η^i for each relation R^i.

Inference

Consider again a binary relation R over a single type T. Since we use conjugate priors on the entries in η, it is simple to compute $P(R|z) = \int P(R|\eta, z)p(\eta)d\eta$:

$$P(R|z) = \prod_{a,b \in \mathcal{N}} \frac{\text{Beta}(m(a, b) + \beta, \bar{m}(a, b) + \beta)}{\text{Beta}(\beta, \beta)}$$

where $m(a, b)$ is the number of pairs (i, j) where $i \in a$ and $j \in b$ and $R(i, j) = 1$, $\bar{m}(a, b)$ is the number of pairs where $R(i, j) = 0$, and $\text{Beta}(\cdot, \cdot)$ is the Beta function. If some entries in R are missing at random, we can ignore them and maintain counts $m(a, b)$ and $\bar{m}(a, b)$ over only the observed values. Even though η is integrated out, it is simple to recover the relationships between clusters given z. The maximum *a posteriori* value of $\eta(a, b)$ given z is:

$$\frac{m(a, b) + \beta}{\bar{m}(a, b) + m(a, b) + 2\beta}.$$

Since we integrate out η, inference can be carried out using Markov chain Monte Carlo methods to sample from the posterior on cluster assignments $P(z|R) \propto P(R|z)P(z)$ (Jain & Neal 2004), or by searching for the mode of this distribution. We are interested in discovering the single best representation for each data set mentioned in this paper, and we search for the best partition z by repeatedly running hill climbing from an initial configuration where a single cluster is used for each type. We also search for the best values of the parameters γ and β using an exponential prior $p(\gamma) \propto e^{-\gamma}$ and an improper prior $p(\beta) \propto \beta^{\frac{-5}{2}}$.

The search uses proposals that move an object from one cluster to another, split a cluster, or merge two clusters. The goal of the IRM can be understood intuitively by representing the relation R as an adjacency matrix. Our search procedure tries to shuffle the rows and columns of this matrix so that it assumes a clean block structure like the matrix in Figure 1a. The same idea applies to relations with more than two dimensions: Figure 1c shows a ternary relation, and here the aim is to shuffle the dimensions so that the matrix takes on a 3-dimensional block structure. Figure 1d shows three relations involving three types. The goal is again to create matrices with clean block structures, but now the partition for T^1 must be the same wherever this type appears.

Related work

Statisticians and sociologists have used the *stochastic blockmodel* to discover social roles in network data. This model relies on a generative process identical to Equation 1, except that the z_i are drawn from a multinomial distribution over a fixed, finite number of clusters (Nowicki & Snijders 2001). Several alternative approaches to relational learning (e.g. Kubica *et al.* (2002)) focus on clique structures, where relational links are expected primarily between members of the same cluster. An advantage of the blockmodel is that it also handles other kinds of relational structures — hierarchies, for example, where members of one cluster tend to send links to individuals from higher-status clusters.

Recent work in machine learning has extended the intuition behind the blockmodel in several directions. There are approaches that learn overlapping clusters for a single type (Wolfe & Jensen 2004) and approaches that handle multiple relations and types using Probabilistic Relational Models (Taskar, Segal, & Koller 2001; Getoor *et al.* 2002). Existing models often focus on data sets with some specific form: for example, the Group-Topic model (Wang, Mohanty, & McCallum 2005) simultaneously clusters entities (e.g. people) and attributes associated with links between those entities (e.g. words). Compared to much of this work, a distinctive feature of the IRM is its ability to automatically handle arbitrary collections of relations, each of which may take any number of arguments. The IRM is a lightweight framework that can be applied to data sets with qualitatively different forms: note that a single generic piece of code was used to analyze all of the data sets in this paper.

Another distinctive feature of the IRM is its ability to learn increasingly complex representations as more data are encountered. This ability allows the model to choose a size vector specifying the number of clusters in each type, and is particularly important for structure discovery in novel domains, where we may have little or no prior knowledge about the number of clusters in each type. Other approaches to choosing the number of clusters are also possible: for example, we could learn systems of many different sizes and choose the best using cross-validation or Bayesian model selection. These alternatives may be practical when there is only one type, but scale badly as the number of types increases: if there are n types, each point in an n-dimensional space of size vectors must be separately considered.

Finally, most previous approaches to relational clustering discover only clusters of objects, and our emphasis on clustering predicates is somewhat unusual. An intelligent system should attempt to find patterns at all levels, and clustering entities, features and relations is one step towards this goal.

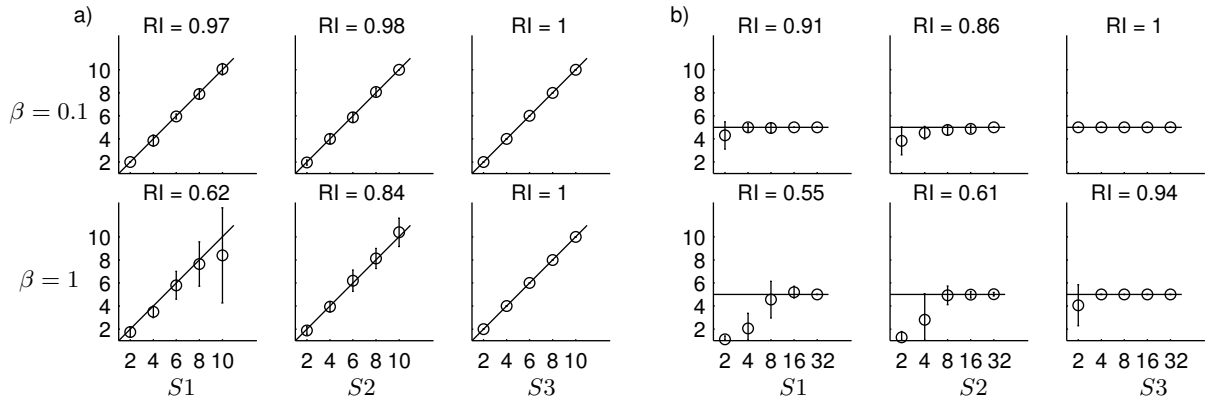

Figure 2: Each sub-plot shows the number of clusters recovered by the IRM (y axis) against (a) the true number of clusters or (b) the number of entities per cluster. In (a), the total number of entities is fixed at 40 and the true dimensionality varies between 2 and 10. In (b), the number of clusters is fixed at 5 and the total number of entities varies between 10 (2 per cluster) and 160 (32 per cluster). The columns represent results for three relational systems described in the text ($S1$, $S2$, $S3$) and the rows show performance for clean data (top) and noisy data (bottom). Each datapoint is an average across 10 randomly generated data sets and across all the types in each system, and error bars show standard deviations. RI is the mean adjusted Rand index, which measures the quality of the clusters discovered.

Synthetic data

We generated synthetic data to explore the IRM's ability to infer the number of clusters in each type. We considered data sets with three different forms. System $S1$ has two types T^1 and T^2 and a single binary relation $R : T^1 \times T^2 \rightarrow \{0, 1\}$. System $S2$ uses four types and three binary relations with domains $T^1 \times T^2$, $T^1 \times T^3$ and $T^2 \times T^4$, and tests the IRM's ability to work with multiple types and relations. System $S3$ is a single ternary relation with domain $T^1 \times T^2 \times T^3$, and tests the IRM's ability to work with higher-order relations. For the first set of simulations, each type included 40 objects, and we generated data sets where the dimensionality d — the true number of clusters in each type — varied between 2 and 10. For each setting of d the clusters were approximately equal in size. For each system and each setting of d, we varied the β parameter (see Equation 1) used to generate the data. When β is small, each relation has sub-blocks that are relatively clean, but the data become noisier as β increases.

The top row of Figure 2a shows that the IRM accurately recovers the true number of clusters in each type when the data are clean ($\beta = 0.1$). The results are averaged across all the types in each system: note that the true dimensionality is always the same for all the types in any given data set. The second row suggests that performance remains stable when $\beta = 1$ and the data are noisier. Performance for the ternary relation (system $S3$) is still perfect when $\beta = 1$, and experiments with $d = 10$ showed that performance only begins to suffer once β reaches 6 and the data are extremely noisy. This result suggests that nonparametric Bayesian approaches may be particularly useful for relational problems: the more that is known about relationships between types, the easier it should be to discover the number of clusters in each type.

To assess the quality of the clusters found by the IRM, we used the adjusted Rand index (Hubert & Arabie 1985). Compared to a ground-truth partition, a randomly generated partition has an expected score of zero, and the correct partition has a score of 1. Mean scores are shown above each plot in Figure 2a, and we see that the IRM accurately recovers both the true number of clusters and the composition of these clusters.

As more and more entities are observed, the expected number of clusters in a data set should probably increase. The IRM has this property: under the CRP prior, the expected number of clusters grows as $O(\log(n))$. Yet this aspect of the prior should not exert too strong an influence: even if the number of entities is large, a successful method should be able to recognize when the true dimensionality is small. We tested this aspect of our model by generating data sets where the true dimensionality was always 5, and the number of entities in each type varied between 10 and 160. Figure 2b shows that when the data are clean, the IRM successfully recovers the dimensionality regardless of the number of entities used. For noisier data sets, there is little statistical evidence for the true clusters when there are only a handful of entities per cluster, but the model reaches the right dimensionality and stays there as the number of entities increases.

Clustering objects and features

Even though the IRM is primarily intended for relational data, it can also discover structure in object-feature data. Any object-feature matrix can be viewed as a relation $R : T^1 \times T^2 \rightarrow \{0, 1\}$ between a set of objects (T^1) and a set of features (T^2), and the IRM provides a strategy for co-clustering, or simultaneously clustering both sets. Since features can be viewed as unary predicates, co-clustering is a simple instance of predicate clustering. Of the many existing approaches to co-clustering, the IRM is closest to the

O1 killer whale, blue whale, humpback, seal, walrus, dolphin
O2 antelope, horse, giraffe, zebra, deer
O3 monkey, gorilla, chimp
O4 hippo, elephant, rhino
O5 grizzly bear, polar bear

F1 flippers, strain teeth, swims, arctic, coastal, ocean, water
F2 hooves, long neck, horns
F3 hands, bipedal, jungle, tree
F4 bulbous body shape, slow, inactive
F5 meat teeth, eats meat, hunter, fierce
F6 walks, quadrapedal, ground

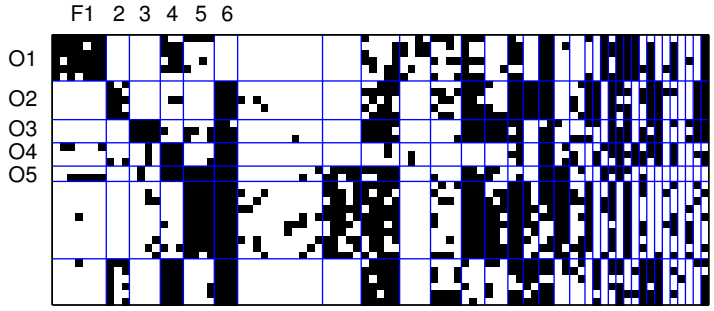

Figure 3: Animal clusters, feature clusters, and a sorted matrix showing the relationships between them. The matrix includes seven of the twelve animal clusters and all of the feature clusters. Some features refer to habitat (jungle, tree, coastal), and others are anatomical (bulbous body shape, has teeth for straining food from the water) or behavioral (swims, slow).

	Animals (11.5)		Medicine (15)		Alyawarra (16)	
IRM	0.50	(12)	0.53	(14)	0.59	(15)
IMM	0.41	(5)	0.47	(9)	0.38	(5)

Table 1: Adjusted Rand indices comparing the best IRM solution and the best IMM solution to ground truth partitions. In parentheses are the true number of clusters (top row) and the number of clusters found by each model (bottom rows).

work of Hofmann & Puzicha (1999).

Figure 3 shows that coherent clusters emerge when the IRM is applied to a 50 by 85 animal-feature matrix collected in a psychological experiment (Osherson *et al.* 1991). Feature ratings were collected on a scale from 0 to 100, and we created a binary matrix by thresholding at the global mean. The feature clusters capture the *coherent covariation* of features across the objects in the data set. Importantly, the model also discovers relationships between feature and object clusters: for example, aquatic mammals tend to have aquatic features.

The IRM reduces to the infinite mixture model (IMM, Rasmussen 2000) if we choose not to cluster the features, assuming instead that each feature is generated independently over the animal partition. Applied to the Osherson data, the IMM finds 5 animal clusters and the IRM finds 12. Any single feature may provide weak evidence for the additional structure discovered by the IRM, but grouping several of these features allows the IRM to discover a finer-grained partition.

We asked two human subjects to sort the animals into groups. One used 13 groups and the other used 10, and we compared the model solutions to these partitions using the adjusted Rand index. The IRM matched both human solutions better than the IMM, and Table 1 reports the mean values achieved.

Learning ontologies

Although there are many kinds of domain theories, ontologies have played a particularly important role in the development of AI. Many researchers have developed ontologies and used them to support learning and inference, but the acquisition of ontological knowledge itself has received less attention. We demonstrate here that the IRM discovers a simple biomedical ontology given data from the Unified Medical Language System (UMLS, McCray 2003).

The UMLS includes a semantic network with 135 concepts and 49 binary predicates. The concepts are high-level concepts like 'Disease or Syndrome', 'Diagnostic Procedure', and 'Mammal.' The predicates include verbs like *complicates*, *affects* and *causes*. We applied the IRM to the ternary relation $R : T^1 \times T^1 \times T^2 \rightarrow \{0, 1\}$, where T^1 is the set of concepts and T^2 is the set of binary predicates (see Figure 1c). We have already seen that features (unary predicates) can be clustered, and here we see that predicates of higher orders can also be clustered. Our general philosophy is that *every* type is potentially a candidate for clustering, although there may be problem-specific reasons why we choose not to cluster some of them.

Figure 4 shows some of the clusters that emerge when we cluster both concepts and predicates. 14 concept clusters and 21 predicate clusters are found in total. We assessed the quality of the concept clusters using a 15 cluster partition created by domain experts (McCray *et al.* 2001). The expert-designed partition includes clusters labeled 'Living Things', 'Chemicals and Drugs' and 'Disorders' that match some of the clusters shown in Figure 4. Again, the IRM discovers not just clusters, but relationships between these clusters. By computing maximum *a posteriori* values of $\eta(a, b)$, we identify the pairs of clusters (a, b) that are most strongly linked, and the predicates that link them. Some of the strongest relationships tell us that biological functions *affect* organisms, that chemicals *cause* diseases, and that biologically active substances *complicate* diseases.

If we are interested only in discovering concept clusters, the IMM can be applied to a flattened version of the relational data. Suppose that a is an element of T^1, and we wish to flatten the ternary relation $R : T^1 \times T^1 \times T^2 \rightarrow \{0, 1\}$. The features of a correspond to all values of $R(a, x^1, x^2)$ where $x^1 \in T^1$ and $x^2 \in T^2$ and all values of $R(x^1, a, x^2)$. Any relational system can be similarly converted into an object feature matrix involving just one of its component dimensions. Table 1 suggests that the IRM solution for the relational data matches the expert partition somewhat better than the best solution for the IMM on the flattened data.

a)

Concept clusters

1.Organisms	2.Chemicals	3.Biological functions	4.Bio-active substances	5.Diseases
Alga	Amino Acid	Biological function	Antibiotic	Cell dysfunction
Amphibian	Carbohydrate	Cell function	Enzyme	Disease
Animal	Chemical	Genetic function	Poisonous substance	Mental dysfunction
Archaeon	Eicosanoid	Mental process	Hormone	Neoplastic process
Bacterium	Isotope	Molecular function	Pharmacologic substance	Pathologic function
Bird	Steroid	Physiological function	Vitamin	Expt. model of disease

Predicate clusters

affects

analyzes
assesses effect of
measures

diagnoses
indicates
prevents
treats

carries out
exhibits
performs

b)

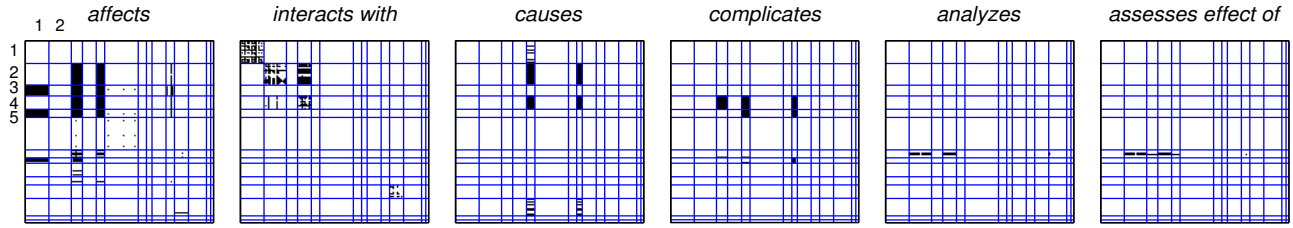

Figure 4: (a) Predicate and concept clusters found using the UMLS data. We have labeled the concept clusters and shown only six members of each. (b) Adjacency matrices for six predicates, where the rows and columns are sorted according to the 14 cluster concept partition. The first five clusters are the clusters shown in (a): we see, for instance, that chemicals affect biological functions and that organisms interact with organisms.

Learning kinship systems

Australian tribes are renowned among anthropologists for the complex relational structure of their kinship systems. We focus here on the Alyawarra, a tribe from Central Australia (Denham 1973). To a first approximation, Alyawarra kinship is captured by the Kariera system shown in Figure 5a. The tribe has four kinship sections, and Figure 5 shows how the sections of individuals are related to the kinship sections of their parents. For example, every member of section 1 has a mother in section 4 and a father in section 3. We show here that the IRM discovers some of the properties of this system.

Denham asked 104 tribe members to provide kinship terms for each other. Figure 5c shows six of the 26 different kinship terms recorded: for each term, the (i, j) cell in the corresponding matrix indicates whether person i used that term to refer to person j. The four kinship sections are clearly visible in the first two matrices. *Adiadya* refers to a classificatory younger brother or sister: that is, to a younger person in one's own section, even if he or she is not a biological sibling. *Umbaidya* is used by female speakers to refer to a classificatory son or daughter, and by male speakers to refer to the child of a classificatory sister. We see from the matrix that women in section 1 have children in section 4, and vice versa. *Anowadya* refers to a preferred marriage partner. The eight rough blocks indicate that men must marry women, that members of section 1 are expected to marry members of section 2, and that members of section 3 are expected to marry members of section 4.

We applied the IRM to the ternary relation $R : T^1 \times T^1 \times T^2 \rightarrow \{0, 1\}$ where T^1 is the set of 104 people and T^2 is the set of kinship terms (see Figure 1c). Denham recorded demographic information for each of his informants, and we created a "ground truth" partition by assigning each person to one of 16 clusters depending on gender, kinship section, and a binary age feature (older than 45). The best solution for the IRM uses 15 clusters, and Figure 5b shows that these clusters are relatively clean with respect to the dimensions of age, gender, and kinship section.

As for the biomedical data, we can apply the IMM to a flattened version of the data if we are interested only in clustering the people. Table 1 suggests that the IRM solution captures the true structure substantially better than the IMM.

Clustering with multiple types and relations

The problem of theory discovery is especially interesting when there are multiple types and relations. Our final example shows that the IRM discovers structure in a political data set including 14 countries, 54 binary predicates representing interactions between countries, and 90 features of the countries (Rummel 1999). To create a binary data set, we thresholded each continuous variable at its mean and used one-of-n coding for the categorical variables. The resulting data set has three types: countries (T^1), interaction predicates (T^2) and country features (T^3), and two relations: $R^1 : T^1 \times T^1 \times T^2 \rightarrow \{0, 1\}$, and $R^2 : T^1 \times T^3 \rightarrow \{0, 1\}$. The IRM analyzes R^1 and R^2 simultaneously and discovers partitions of all three types.

The model partitions the 14 countries into the five groups shown in Figure 6a. The data come from 1965 and there are two groups from the Western bloc, a group from the communist bloc, and two groups from the so-called "neutral bloc." The model discovers 18 clusters of interaction predicates, and Figures 6b through 6i represent some of the clusters that emerge. Note that the countries in each matrix are sorted according to the order in 6a. The clusters in 6b and 6e represent positive and negative interactions, and the cluster in 6i confirms that the country partition is well explained by bloc membership.

The IRM divides the country features into the five groups

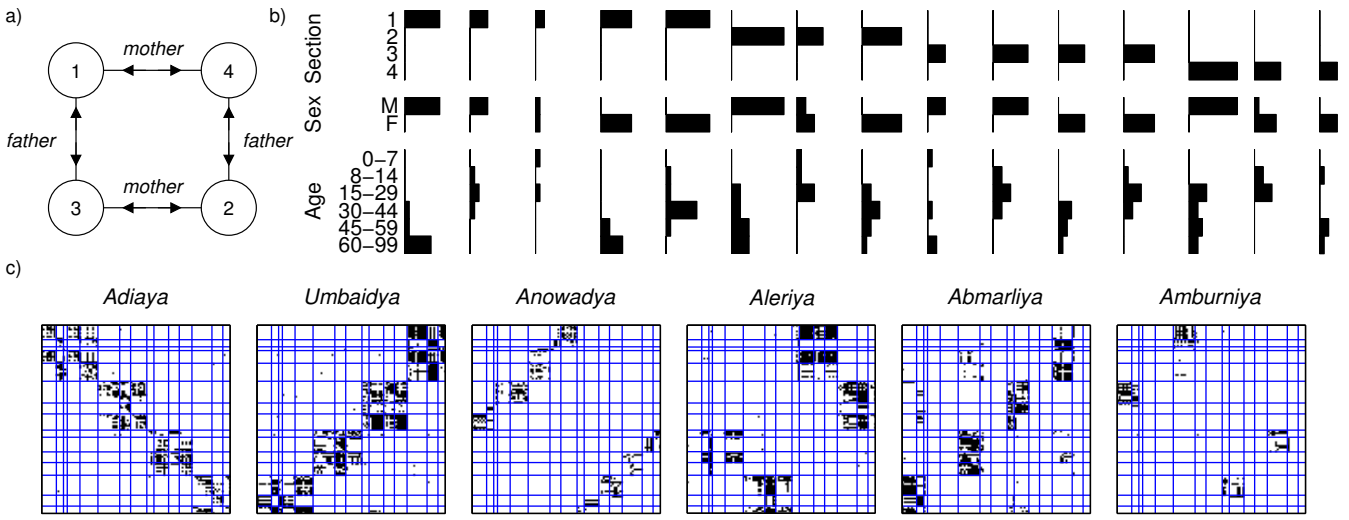

Figure 5: (a) The Kariera kinship system. Each person belongs to one of four kinship sections, and the section of any person predicts the sections of his or her parents. (b) Composition of the 15 clusters found by the IRM. The six age categories were chosen by Denham, and are based in part on Alyawarra terms for age groupings (Denham 1973). (c) Data for six Alyawarra kinship terms. The 104 individuals are sorted by the clusters shown in (b).

shown in Figure 6a. The first group includes a single feature — 'non-communist' — which captures one of the most important aspects of this Cold-War data set. The second and third clusters include features that are characteristic of Western and communist countries respectively, and the fourth cluster includes features that are often true of developing countries but never true of the UK and the USA. [1]

Conclusion

We presented the Infinite Relational Model, a framework for simultaneously clustering one or more sets of entities and discovering the relationships between clusters that are possible or likely. Our framework supports the discovery of simple theories that specify the kinds of entities in a domain and the relations that hold between them. These theories capture important aspects of semantic knowledge, but we are ultimately interested in more expressive representations that can capture a greater proportion of human knowledge. Logical representations, for example, will probably be needed to fully capture knowledge about kinship.

There are many ways to formalize the notion of a relational system, and it is useful to arrange these formalizations along a spectrum from simple to complex. We considered relatively simple systems, which allowed us to give a principled account of how these systems might be learned. Methods for learning logical theories (Muggleton & De Raedt 1994; Kok & Domingos 2005) consider relational systems at the more complex end of the spectrum, and it may be worth thinking about hybrid approaches where the clusters discovered by the IRM serve as primitives in more complex theories. Adding representational power while preserving learnability is an imposing challenge, but we hope that approaches like ours will help bring algorithms for relational learning closer to theories of the organization and development of human semantic knowledge.

Acknowledgments Supported in part by AFOSR MURI contract FA9550-05-1-0321, the William Asbjornsen Albert memorial fellowship (CK) and the Paul E. Newton Chair (JBT). We thank Steven Sloman for providing the animal-feature data and Woodrow Denham for providing the Alyawarra data.

[1] Figure 6 reflects some inconsistencies in the original data: for instance, 6i suggests that Israel is part of the neutral bloc, but the second labeled feature in 6a suggests that Israel is part of the Western bloc.

References

Antoniak, C. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics* 2:1152–1174.

Davis, E. 1990. *Representations of commonsense knowledge*.

Denham, W. 1973. *The detection of patterns in Alyawarra nonverbal behavior*. Ph.D. Dissertation, University of Washington.

Getoor, L.; Friedman, N.; Koller, D.; and Taskar, B. 2002. Learning probabilistic models of link structure. *Journal of Machine Learning Research* 3:679–707.

Hofmann, T., and Puzicha, J. 1999. Latent class models for collaborative filtering. In *Proc. 16th International Joint Conference on Artificial Intelligence*.

Hubert, L., and Arabie, P. 1985. Comparing partitions. *Journal of Classification* 2:193–218.

Jain, S., and Neal, R. M. 2004. A split-merge Markov chain Monte Carlo procedure for the Dirichlet Process mixture model. *Journal of Computational and Graphical Statistics* 13:158–182.

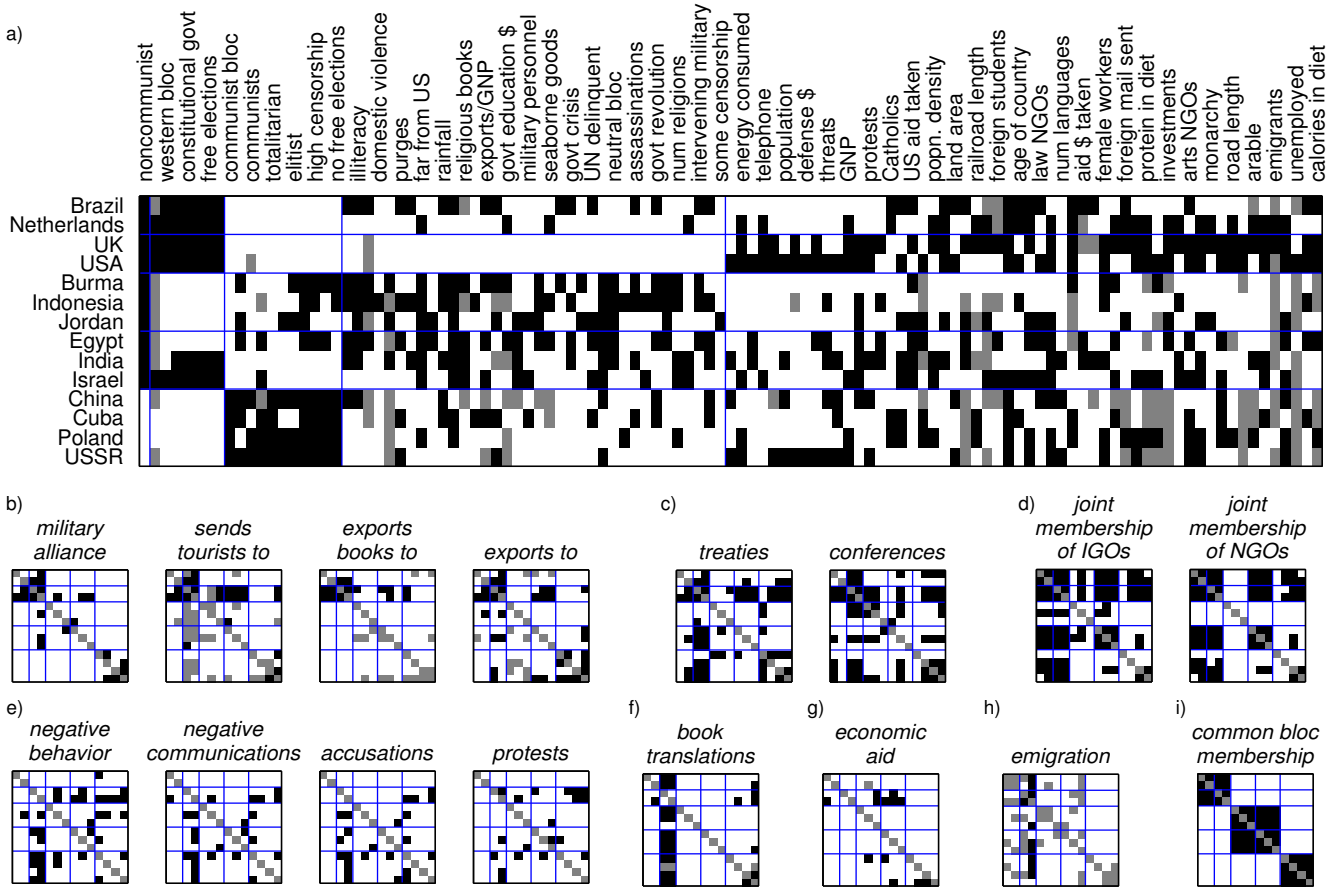

Figure 6: (a) Country clusters and feature clusters for the political data. Every second feature is labelled, and gray entries indicate missing data. (b) – (i) Representatives of eight predicate clusters found by the IRM. The countries in each matrix are ordered according to the partition in (a).

Kok, S., and Domingos, P. 2005. Learning the structure of Markov logic networks. In *Proc. 22nd International Conference on Machine Learning*.

Kubica, J.; Moore, A.; Schneider, J.; and Yang, Y. 2002. Stochastic link and group detection. In *Proc. 17th National Conference on Artificial Intelligence*.

McCray, A. T.; Burgun, A.; and Bodenreider, O. 2001. Aggregating UMLS semantic types for reducing conceptual complexity. In *Medinfo*, volume 10, 216–20.

McCray, A. T. 2003. An upper level ontology for the biomedical domain. *Comparative and Functional Genomics* 4:80–84.

Minsky, M. 1975. A framework for representing knowledge. In Winston, P., ed., *The psychology of computer vision*. New York, NY: McGraw-Hill. 211–277.

Muggleton, S., and De Raedt, L. 1994. Inductive logic programming: theory and methods. *Journal of Logic Programming* 19,20:629–679.

Nowicki, K., and Snijders, T. A. B. 2001. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association* 96(455):1077–1087.

Osherson, D. N.; Stern, J.; Wilkie, O.; Stob, M.; and Smith, E. E. 1991. Default probability. *Cognitive Science* 15:251–269.

Pitman, J. 2002. Combinatorial stochastic processes. Notes for Saint Flour Summer School.

Rasmussen, C. E. 2002. The infinite Gaussian mixture model. In *NIPS*, volume 13.

Rummel, R. J. 1999. Dimensionality of Nations project: attributes of nations and behavior of nation dyads, 1950–1965. ICPSR data file.

Taskar, B.; Segal, E.; and Koller, D. 2001. Probabilistic classification and clustering in relational data. In *Proc. 18th International Joint Conference on Artificial Intelligence*, volume 15.

Wang, X.; Mohanty, N.; and McCallum, A. 2005. Group and topic discovery from relations and text. In *Proc. KDD workshop on link discovery*.

Wolfe, A. P., and Jensen, D. 2004. Playing multiple roles: discovering overlapping roles in social networks. In *Proc. ICML workshop on statistical relational learning and its connections to other fields*.