Analogical Reasoning with Relational Bayesian Sets

Ricardo Silva
Gatsby Unit
University College London
rbas@gatsby.ucl.ac.uk

Katherine A. Heller
Gatsby Unit
University College London
heller@gatsby.ucl.ac.uk

Zoubin Ghahramani
Department of Engineering
University of Cambridge
zoubin@eng.cam.ac.uk

Abstract

Analogical reasoning depends fundamentally on the ability to learn and generalize about relations between objects. There are many ways in which objects can be related, making automated analogical reasoning very challenging. Here we develop an approach which, given a set of pairs of related objects $\mathbf{S} = \{A^1{:}B^1, A^2{:}B^2, \ldots, A^N{:}B^N\}$, measures how well other pairs $A{:}B$ fit in with the set $\mathbf{S}$. This addresses the question: is the relation between objects A and B analogous to those relations found in $\mathbf{S}$? We recast this classical problem as a problem of Bayesian analysis of relational data. This problem is non-trivial because direct similarity between objects is not a good way of measuring analogies. For instance, the analogy between an electron around the nucleus of an atom and a planet around the Sun is hardly justified by isolated, non-relational, comparisons of an electron to a planet, and a nucleus to the Sun. We develop a generative model for predicting the existence of relationships and extend the framework of Ghahramani and Heller (2005) to provide a Bayesian measure for how analogous a relation is to other relations. This sheds new light on an old problem, which we motivate and illustrate through practical applications in exploratory data analysis.

1 CONTRIBUTION

Consider the following illustrative problems in exploratory data analysis:

Example 1 A researcher has a large collection of papers. She plans to use such a database in order to write an comprehensive article about the evolution of her field through the past two decades. In particular, she has a collection organized as pairs of papers, where one cites the other, i.e., a collection of pairs $A{:}B$ meaning A cites B. There are several reasons why a paper might cite another: A is a big bibliographic survey, or B was written by the advisor of the author of A, or B was given a best paper award, or the authors were geographically close, or a combination of several such features. Such combinations define a (potentially very large) variety of *subpopulations* of pairs of papers. While there might be relevant information about A and B in the database, which subpopulations of citations $A{:}B$ belongs to is never explicitly indicated in the data. Yet the researcher is not completely in the dark: she already has an idea of important subgroups of citations which are representative of the most interesting subpopulations, although it might be difficult to characterize any such set with a simple description. She would like to know which other pairs of papers might belong to such subgroups. Instead of worrying about writing some simple query rules that explain the common properties of such subgroups, she would rather have an intelligent information retrieval system that is able to identify which other pairs in the database are linked in an analogous way to those pairs in her selected sets. □

Example 2 A scientist is investigating a population of proteins, within which some pairs are known to interact, while the remaining pairs are known not to interact. In this study, it is known that recorded gene expression profiles of the respective genes can be used as a reasonable predictor of the existence or not of an interaction. The current state of knowledge is still limited regarding which subpopulations (i.e., classes) of interactions exist, although a partial hierarchy of such classes for some proteins is available. Given a selected set of interacting proteins that are believed to belong to a particular level of the class hierachy, the researcher would like to query her database to discover other plausible pairs of proteins whose mechanism of linkage is of the same nature as in the selected set, i.e.,

to query for analogous relations. Ideally, she would like to do it without being required to write down query rules that explicitly describe the selected set. □

Such are problems of *analogical reasoning*, instantiated as practical problems of information retrieval for exploratory data analysis. Even under the absence of clear class labels for links such as paper citations and protein interactions (what is recorded is which pairs are linked and which are not), one might have at hand a subpopulation of interest that is best described by a sample of linked objects. The question to be asked is of an exploratory nature: which other objects in my relational database are linked in a similar way? In both examples, one has a relational database, and it is possible to create models for predicting the *existence* or lack of a relationship using features such as paper attributes and gene expression profiles. In both examples, it is not fully known how to explicitly describe classes of relations that are believed to exist (and it is a nuisance to select negative examples by hand to learn a classifier).

We propose a method for retrieving relations based on the Bayesian scoring function as proposed in the Bayesian sets method (Ghahramani and Heller, 2005): given a set of related items that are postulated to come from a subpopulation of interest, the goal is to rank existing links according to a measure of similarity with respect to this set. We interpret this problem as the classical problem of analogical reasoning. That is, suppose we have a pair (or set of pairs) of objects A:B. Which other pairs of objects in relational database best reflect a relation analogous to A:B? This paper provides a novel and probabilistically sound solution to this problem. Moreover, this work extends the Bayesian sets method to discriminative models.

We will focus solely on finding pairwise relations. The idea can be extended to more complex relations, but we will not pursue this here.

In Section 2 we discuss related work while describing the difference between analogical reasoning and standard retrieval tasks. The approach is introduced in Section 3 and evaluated in Section 4.

2 RELATED WORK

To define an analogy is to define a measure of similarity between structures of related objects (pairs, in our case). The key aspect is that, typically, we are not interested in how each individual object in a candidate pair is similar to individual objects in the query pairs. As an illustration, consider an analogical reasoning question from a SAT-like exam where for a given pair (say, *water:river*) we have to choose (out of 5 pairs) the one that best matches the type of relation implicit in such a "query." In this case, it is reasonable to say *car:traffic* would be a better match than (the somewhat nonsensical) *soda:ocean*, since cars flow through traffic, and so does water through a river. Notice that if we were to measure the similarity between *objects* instead of *relations*, it now seems reasonable to say that *soda:ocean* in this case would be a much closer pair. In the examples given in the previous section, similarity between pairs of objects is only meaningful to the extent by which such features are useful to predict the existence of the relationships.

There is a large literature on analogical reasoning in artificial intelligence and psychology. We refer to French (2002) for a survey, as well as to some recent machine learning papers on clustering (Marx et al., 2002), prediction (Turney and Littman, 2005) and dimensionality reduction (Memisevic and Hinton, 2005). Here we will use a Bayesian framework for inferring similarity of relations. Given a set of relations, our goal will be to score others as relevant or not. The score is a Bayesian model comparison generalizing the "Bayesian sets" score (Ghahramani and Heller, 2005) to discriminative models over pairs of objects.

The graphical model formulation of Getoor et al. (2002) incorporates models of link existence in relational databases, an idea used explicitly in Section 3 as the first step of our problem formulation. In the clustering literature, the probabilistic approach of Kemp et al. (2006) is motivated by principles similar to those in our formulation: the idea is that there is an infinite mixture of subpopulations that generates the observed relations. Our problem, however, is to retrieve other elements of a subpopulation described by elements of a query set, a goal that is also closer to the classical paradigm of analogical reasoning.

To emphasize once more, our focus here is not on predicting the presence or absence of links, as in, e.g., (Popescul and Ungar, 2003) but rather on retrieving similar links from among those already known to exist in the relational database. Neither is our focus to provide a fully unsupervised clustering of the whole database of pairs (as in, e.g., Kemp et al., 2006), nor to use relational information to improve classification of other attributes (as in, e.g., Getoor et al., 2002).

3 FUNCTIONS AS ANALOGIES

We now describe the analogical reasoning principle more formally. Let $\mathcal{A}$ and $\mathcal{B}$ represent object spaces. To say that an interaction A:B is analogous to $\mathbf{S} = \{A^1{:}B^1, A^2{:}B^2, \ldots, A^N{:}B^N\}$ is to define a measure of similarity between the pair and the set of pairs. However, this similarity is not (directly) given by the

information contained in the distribution of objects $\{A^i\} \subset \mathcal{A}$, $\{B^i\} \subset \mathcal{B}$, but by the *mappings* classifying such pairs as being linked:

> **Bayesian analogical reasoning formulation**: Consider a space of latent functions in $\mathcal{A} \times \mathcal{B} \rightarrow \{0, 1\}$. Assume that A and B are two objects classified as linked by some unknown function $f(A, B)$, i.e., $f(A, B) = 1$. We want to quantify how similar the function $f(A, B)$ is to the function $g(\cdot, \cdot)$, which classifies all pairs $(A^i, B^i) \in \mathbf{S}$ as being linked, i.e., $g(A^i, B^i) = 1$. The similarity should be a function of the observations $\{\mathbf{S}, A, B\}$ and our prior distribution over $f(\cdot, \cdot)$ and $g(\cdot, \cdot)$.

Functions $f(\cdot)$ and $g(\cdot)$ are unobserved, hence the need for a prior that will be used to integrate over the function space. The resulting similarity metric will be defined through a Bayes factor, as explained next. For simplicity, we will consider a family of latent functions that is parameterized by a finite-dimensional vector: the logistic regression function with multivariate Gaussian priors for its parameters.

For a particular pair ($A^i \in \mathcal{A}$, $B^j \in \mathcal{B}$), let $X^{ij} = [\Phi_1(A^i, B^j)\ \Phi_2(A^i, B^j) \ldots \Phi_K(A^i, B^j)]^T$ be a point on a feature space defined by the mapping $\Phi : \mathcal{A} \times \mathcal{B} \rightarrow \Re^K$. Let $C^{ij} \in \{0, 1\}$ be an indicator of the existence of a link between A^i and B^j in the database. Let $\Theta = [\theta_1, \ldots, \theta_K]^T$ be the parameter vector for our logistic regression model

$$P(C^{ij} = 1 | X^{ij}, \Theta) = \text{logistic}(\Theta^T X^{ij}) \tag{1}$$

where $\text{logistic}(x) = (1 + e^{-x})^{-1}$. Our measure of similarity for a pair (A^i, B^j) with respect to a query set $\mathbf{S}$ is the probabilistic similarity measure of Bayesian sets (Ghahramani and Heller, 2005) on a log-scale:

$$\begin{aligned} score(A^i, B^j) &= \log P(C^{ij} = 1 | X^{ij}, \mathbf{S}, \mathbf{C}^{\mathbf{S}} = 1) \\ &- \log P(C^{ij} = 1 | X^{ij}) \end{aligned} \tag{2}$$

where $\mathbf{C}^{\mathbf{S}}$ is the vector of link indicators for $\mathbf{S}$: i.e., $C^1 = 1, C^2 = 1, \ldots, C^N = 1$ indicates that all pairs in $\mathbf{S}$ are linked.

The general framework is as follows. We are given a relational database $(\mathbf{D}_A, \mathbf{D}_B, \mathbf{L}_{AB})$, where the first two components of this database are sampled respectively from $\mathcal{A}$ and $\mathcal{B}$. Relationship table $\mathbf{L}_{AB}$ is a binary matrix assumed to be generated by a logistic regression model of link existence. A query proceeds according to the folllwing steps:

1. the user selects a set of pairs $\mathbf{S}$ that are linked in the database;

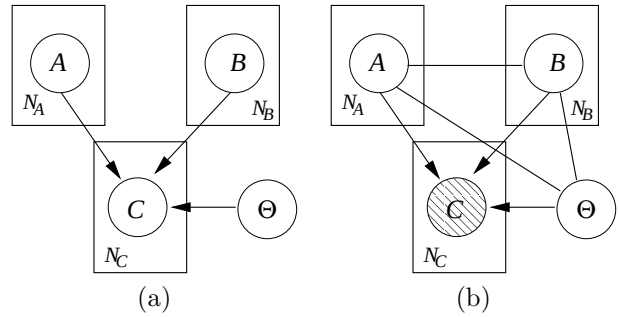

Figure 1: (a) Graphical plate representation for the relational Bayesian logistic regression, where N_A, N_B and N_C are the number of objects of each class. (b) Extra dependencies induced by further conditioning on C are represented by undirected edges.

2. the system performs Bayesian inference to obtain the corresponding posterior distribution for Θ, $P(\Theta | \mathbf{S}, \mathbf{C}^{\mathbf{S}})$, given a Gaussian prior $P(\Theta)$;

3. the system iterates through all linked pairs, computing the following for each pair

$$\begin{aligned} &P(C^{ij} = 1 | X^{ij}, \mathbf{S}, \mathbf{C}^{\mathbf{S}} = 1) = \\ &\int P(C^{ij} = 1 | X^{ij}, \Theta) P(\Theta | \mathbf{S}, \mathbf{C}^{\mathbf{S}} = 1)\, d\Theta \end{aligned} \tag{3}$$

   as well as $P(C^{ij} = 1 | X^{ij})$ by integration over $P(\Theta)$[1], and then sorts them according to the score in Equation (2);

The corresponding plate model is illustrated in Figure 1(a). Latent parameter vector $\Theta = \{\theta_1, \theta_2, \ldots, \theta_K\}^T$ and objects A and B are ancestors of link indicator C. By conditioning on $C = 1$, elements of Θ will be connected to and share information from input data $\{A, B\}$, as in Figure 1(b). This information can be passed forward to evaluate other points. The suggested setup scales as $O(K^3)$ due to the matrix inversions necessary for (variational) Bayesian logistic regression (Jaakkola and Jordan, 2000). If necessary, a further approximation for $P(\Theta | \mathbf{S}, \mathbf{C}^{\mathbf{S}})$ might be imposed if the dimensionality of Θ is too high.

3.1 Choice of features and relational discrimination

Our setup assumes that the feature space Φ provides a reasonable classifier to predict the existence of links. It is evident that the proposed framework could be

[1] Since the integral used in the Bayesian logistic function does not have a closed formula, in all of these expressions we use the Bayesian variational approximation by Jaakkola and Jordan (2000). A short summary of this approach is given by Silva et al. (2007).

used for non-relational problems with arbitrary classification functions. However, our analogical reasoning formulation is a relational model to the extent that it models presence and absence of interactions between objects: by conditioning on the link indicators, the similarity score between A:B and C:D is always a function of pairs (A, B) and (C, D) that is not in general decomposable as similarities between A and C, B and D. Again, this is illustrated by Figure 1. One might also have records of pairwise information on other relational tables $\mathbf{D}_{AB}$ besides the targeted one. For instance, one might have a measure in a protein database that is a binary indicator of both proteins being produced in the same area in the cell or not, or the number of common proteins that interact with the pair. Our method then learns to rank similarity of relations based on features extracted for a relational database, and such features have a role similar to the latent variables in block-models, as discussed in the sequel. Useful predictive features can also be generated automatically with a variety of algorithms (e.g., the "structural logistic regression" of Popescul and Ungar, 2003). See also Džeroski and Lavrač (2001). Jensen and Neville (2002) discuss shortcomings of automated methods for automated feature selection in relational classification.

Our analogical reasoning formulation also assumes all subpopulations of interest are measured on the same feature space. This allows for comparisons between, e.g., cells from different species, or webpages from different web domains, as long as $\Phi(\cdot, \cdot)$ is the same. The most general analogical reasoning formulation would not have this requirement, but for the problem to be well-defined, features from the different spaces must be related somehow. A hierarchical Bayesian formulation for linking different feature spaces is one possibility which might be treated in a future work.

3.2 Priors

The choice of prior is based on the observed data, in a way that is analogous to the choice of priors used in the original formulation of Bayesian sets (Ghahramani and Heller, 2005). Let $\widehat{\Theta}$ be the maximum likelihood estimator of Θ. Since the number of possible pairs grows at a quadratic rate with the number of objects, we do not use the whole database for maximum likelihood estimation. Also, since most databases have a sparse link matrix, to get $\widehat{\Theta}$ we use all linked pairs as members of the positive class ($C = 1$), and sample unlinked pairs as members of the negative class ($C = 0$)[2].

We use the prior $P(\Theta) = \mathcal{N}(\widehat{\Theta}, (c\widehat{\mathbf{T}})^{-1})$, where $\mathcal{N}(\mathbf{m}, \mathbf{V})$ is a normal of mean $\mathbf{m}$ and variance $\mathbf{V}$. Matrix $\widehat{\mathbf{T}}$ is the empirical second moments matrix of the linked object features in $\mathbf{X}$, a measure of their variability. Constant c is a smoothing parameter set by the user. In our experiments, we selected it to be twice the total number of links.

Empirical priors are a sensible choice, since this is a retrieval, not a predictive, task. Basically, the entire data set is the population. A data-dependent prior based on the population is quite important for an approach like Bayesian sets, since deviances from the "average" behaviour in the data are useful to discriminate between subpopulations.

Silva et al. (2007) present several illustrations of the score function behavior under different choices of priors and query sets.

3.3 Connections to Bayesian sets and block models

The model in Figure 1(a) is a typical conditional relational model, i.e., conditioned on objects and parameters the resulting relations are i.i.d. Under the original Bayesian sets formulation, the score function can be described by (the logarithm of) the Bayes factor comparing the models in Figure 2.

In contrast, consider the following direct modification of the Bayesian sets formulation to this problem: flatten the data, creating for each pair (A^i, B^j) a row in the database with an extra binary indicator of relationship existence. Create a joint model for pairs by using the marginal models for $\mathcal{A}$ and $\mathcal{B}$ and treating different rows as being independent. This ignores the fact that the resulting flattened data points are not really i.i.d. under such a model, because the same object might appear in multiple relations (Džeroski and Lavrač, 2001). The model also fails to capture the dependency between A^i and B^j that arises from conditioning on C^{ij}, even if A^i and B^j are marginally independent. Nevertheless, heuristically this approach can sometimes produce some good results, and for several types of probability families it is very computationally efficient.

A different approach for modeling relational data is the block-model used for years by statisticians and sociologists for modeling social networks (Kemp et al., 2006; Airoldi et al., 2006). The basic idea is to use hidden variables[3] in place of our feature vector X^{ij}: this is partially motivated by the fact that typically,

[2] In our experiments, we sample 10 "negative" pairs for each "positive" one, and weight them to reflect the proportion in the database (e.g., if we sample 10 negatives for each positive, while in the database there are 200 negatives for each positive, we count each negative case as being 20 cases replicated.)

[3] Such hidden variables are usually discrete indicators of some latent cluster membership for objects. The model typically requires a cross-clustering for object membership

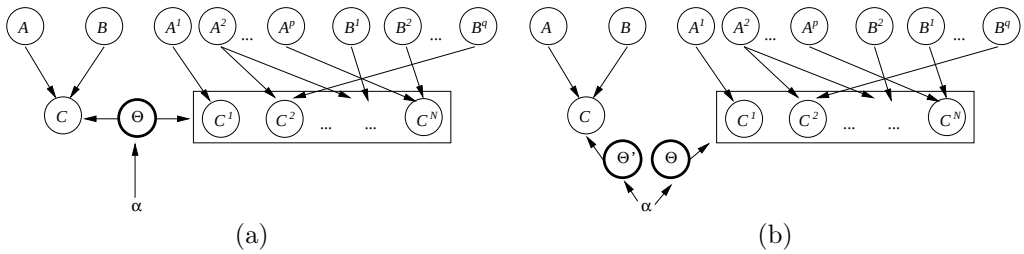

Figure 2: The score of a new data point $\{A, B, C\}$ is given by the Bayes factor that compares models (a) and (b). Node α represents the hyperparameters for Θ. In (a), the generative model is the same for both the new point and the query set represented in the rectangle. Notice that our conditioning set $\mathbf{S}$ of pairs in $\{A^i\} \times \{B^j\}$ might contain repeated instances of a same point, i.e., some A^i or B^j might appear multiple times in different relations, as illustrated by nodes A^i with multiple outgoing edges. In (b), the new point and the query set do not share the same parameters.

in social network analysis, there are no easily available features of the population that are recorded. To compute quantities such as the marginal likelihood of the model one has to integrate out a large number of hidden variables.

For a moderate number of objects straight evaluation of all pairs might be computationally infeasible in the block-model setup. Our discriminative model only needs the unlinked pairs when setting a prior, which is accomplished by sampling when the total number of unlinked pairs is too large. Throwing part of the information from unlinked pairs is arguably less harmful to our goals than to the clustering procedure performed with block-models.

Our model assumes link indicators are independent given object features, which might not be the case for particular choices of feature space. In theory, block-models sidestep this issue by learning all the necessary latent features that account for link dependence. An important future extension of our work would consist of tractably accounting for residual link association that is not intermediated by our observed features.

3.4 On continuous relations

Although we focus on measuring similarity of qualitative relationships, the same idea could be extended to *continuous* measures of relationship. For instance, Turney and Littman (2005) measure relations between words by their co-occurrences in texts next to a set of joining terms, such as the two words being connected by a specific preposition. Several similarity metrics can be defined on this vector of continuous relationship (e.g., cosine distance). However, given data on word features and a predictive model for such quantitative relations, one can directly adapt our model[4].

and relation membership.

[4]Notice that our approach would still not be directly comparable to the one by Turney and Littman, since unlike them we would make use of some external data source for the features of the words.

4 EXPERIMENTS

We now describe two experiments on analogical retrieval using the proposed model. Evaluation of the significance of retrieved items often relies on subjective assessments (Ghahramani and Heller, 2005). To simplify our study, we will focus on particular setups where objective measures of success can be derived.

Our main standard of comparison will be a "flattened Bayesian sets" algorithm (which we will call "standard Bayesian sets," SBSets, in constrast to the relational model, RBSets). Using a multivariate independent Bernoulli model as in the original paper (Ghahramani and Heller, 2005), we join linked pairs into single rows, and then apply the original algorithm directly on this joined data. This algorithm serves the purpose of both measuring the loss of not treating relational data as such, as well as the limitations of treating similarity of pairs through the generative models of $\mathcal{A}$ and $\mathcal{B}$ instead of the generative model for the latent predictive function $g(\cdot, \cdot)$.

In both experiments, objects are of the same type, and therefore, dimensionality. The feature vector X^{ij} for each pair of objects $\{A^i, B^j\}$ consists of the V features for object A^i, the V features of object B^j, and measures $\{Z_1, \ldots, Z_V\}$, where $Z_v = (A^i_v \times B^j_v)/(\|A^i\| \times \|B^j\|)$, $\|A^i\|$ being the Euclidean norm of the V-dimensional representation of A^i. We also use a constant value (1) as part of the feature set as an intercept term for the logistic regression. The $\mathbf{Z}$ features are exactly the ones used in the cosine distance measure, a common and practical measure widely used in information retrieval (Manning et al., 2007). They also have the important advantage of scaling well with the number of variables in the database. Moreover, adopting such features will make our comparisons in the next

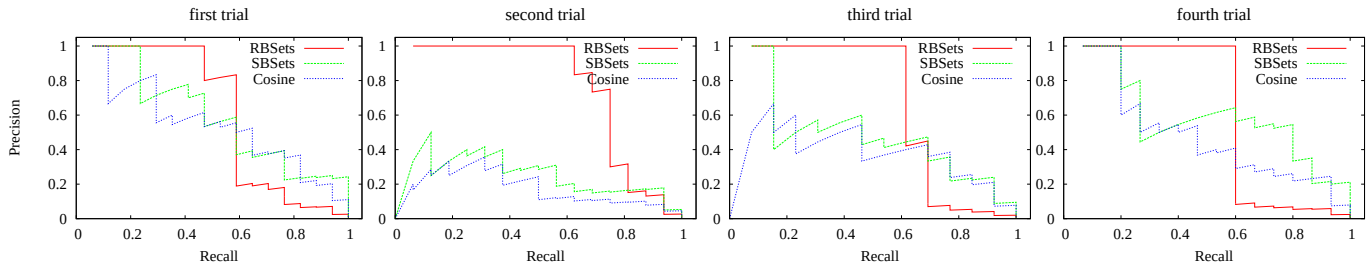

Figure 3: Precision/recall curves for four different random queries of size 10 for the three algorithms: relational Bayesian sets (RBSets), regular Bayesian sets with Bernoulli model (SBSets-B) and cosine distance.

sections more fair, since we evaluate how well cosine distance performs in our task. Notice X^{ij} represents asymmetric relationships as required in our applications. For symmetric relationships, features such as $|A_v^i - B_v^j|$ could be used instead.

4.1 Synthetic experiment

We first discuss a synthetic experiment where there is a known ground truth. We generate data from a simulated model with six classes of relations represented by six different instatiations of Θ, $\{\Theta_0, \Theta_1, \ldots, \Theta_5\}$. This simplified setup defines a multiclass logistic softmax classifier that outputs a class label out of $\{0, 1, \ldots, 5\}$. Object spaces $\mathcal{A}$ and $\mathcal{B}$ are the same, and defined by a multivariate Bernoulli distribution of 20 dimensions, where each attribute has independently a probability 1/2 of being 1. We generate 500 objects, and considered all 500^2 pairs to generate $250,000$ feature vectors X. For each X we evaluate our logistic classifier to generate a class label. If this class is zero, we label the corresponding pair as "unlinked." Otherwise, we label it as "linked." The intercept parameter for parameter vector Θ_0 was set manually to make class 0 appear in at least 99% the data[5], thus corresponding to the usual sparse matrices found in relational data.

The algorithms we evaluate do not know which of the 5 classes the linked pairs originally corresponded to. However, since the labels are known through simulation, we are able to tell how well ranked are points of a particular class given a query of pairs from the same class. Our evaluation is as follows. We generate precision/recall curves for three algorithms: our relational Bayesian sets RBSets, "flattened" standard Bayesian sets with Bernoulli model (SBSets) and cosine distance (summing over all elements in the query). For each query, we randomly sampled 10 elements out of the pool of elements of the least frequent class (about 1% of the total number of links), and ranked the remaining 2320 linked pairs. We counted an element as a hit if it was originally from the selected class.

RBSets gives consistently better results for the top 50% retrievals. As an illustration, we depicted four random queries of 10 items in Figure 3. Notice that sometimes SBSets can do reasonably, often achieving better precision at the bottom 40% recalls: by the virtue of having few objects in the space of elements of this class, a few of them will appear in pairs both in the query and outside of it, facilitating matching by object similarity since half of the pair is already given as input. We conjecture this explains the seemingly strong results of feature-based approaches on the bottom 40%. However, when this does not happen the problem can get much harder, making SBSets much more sensitive to the query than RBSets, as illustrated in some of the runs in Figure 3.

4.2 The WebKB experiment

The WebKB data is a collection of webpages from several universities, where relations are directed and given by hyperlinks (Craven et al., 1998). Webpages are classified as being of type *course*, *department*, *faculty*, *project*, *staff*, *student* and *other*. Documents from four universities (*cornell*, *texas*, *washington* and *wisconsin*) are also labeled as such. Binary data was generated from this database using the same methods of Ghahramani and Heller (2005). A total of 19,450 binary variables per object are generated. To avoid introducing extra approximations into RBSets, we reduced dimensionality in the original representation using singular value decomposition, obtaining 25 measures per object. This also improved the results of our algorithm and cosine distance. For SBSets, this is a way of creating correlations in the original feature space.

To evaluate the gain of our model over competitors, we will use the following setup. In the first query, we are given the pairs of webpages of the type *student* $\rightarrow$ *course* from three of the labeled universities, and eval-

[5] Values for vectors $\Theta_1, \Theta_2, \ldots, \Theta_5$ were otherwise generated by independent multivariate Gaussian distributions with zero mean and standard deviation of 10

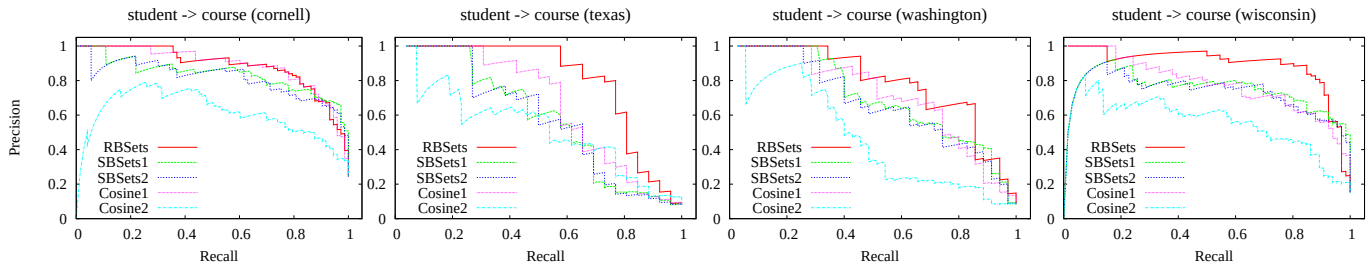

Figure 4: Results for *student* → *course* relationships.

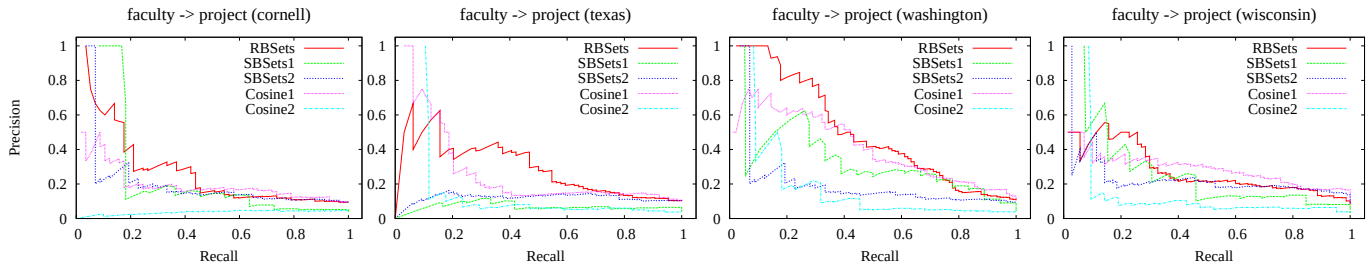

Figure 5: Results for *faculty* → *project* relationships.

uate how relations are ranked in the fourth university. Because we know class labels (while the algorithm does not), we can use the class of the returned pairs to label a hit as being "relevant" or "irrelevant." We label a pair (A^i, B^j) as relevant if and only if A^i is of type *student* and B^j is of type *course*, and A^i links into B^j.

This is a very stringent criterion, since other types of relations could also be valid (e.g., *staff* → *course* appears to be a reasonable match). However, this facilitates objective comparisons of algorithms. Also, the *other* class contains many types of pages, which allows for possibilities such as a *student* → *"hobby"* pair. Such pairs might be hard to evaluate (e.g., is that particular hobby incrementally demanding in a way that coursework is? Is it as fun as taking a machine learning course?) As a compromise, we omit all pages from the category *other* in order to better clarify differences between algorithms[6].

Precision/recall curves for the *student* → *course* queries are shown in Figure 4. There are four queries, each corresponding to a search over a specific university given all valid *student* → *course* pairs from the other three. There are four algorithms on each evaluation: the standard Bayesian sets with the original 19,450 binary variables for each object, plus another 19,450 binary variables, each corresponding to the product of the respective variables in the original pair of objects (SBSets1); the standard Bayesian sets with the original binary variables only (SBSets2); a standard cosine distance measure over the 25-dimensional representation (Cosine 1); a cosine distance measure using the 19,450-dimensional text data with TF-IDF weights (Cosine 2); our approach, RBSets.

In Figure 4, RBSets demonstrates consistently superior or equal precision-recall. Although SBSets performs well when asked to retrieve only *student* items or only *course* items, it falls short of detecting what features of *student* and *course* are relevant to predict a link. The discriminative model within RBSets conveys this information through the parameters.

We also did an experiment with a query of type *faculty* → *project*, shown in Figure 5. This time results between algorithms were closer. To make differences more evident, we adopt a slightly different measure of success: we count as a 1 hit if the pair retrieved is a *faculty* → *project* pair, and count as a 0.5 hit for pairs of type *student* → *project* and *staff* → *project*. Notice this is a much harder query. For instance, the structure of the *project* webpages in the *texas* group was quite distinct from the other universities: they are mostly very short, basically containing links for members of the project and other project webpages.

Although the precision/recall curves convey a global picture of performance for each algorithm, they might

[6]As an extreme example, querying *student* → *course* pairs from the *wisconsin* university returned *student* → *other* pairs at the top four. However, these *other* pages were for some reason course pages - such as `http://www.cs.wisc.edu/~markhill/cs752.html`

Table 1: Area under the precision/recall curve for each algorithm and query.

	C1	C2	RB	SB1	SB2	C1	C2	RB	SB1	SB2
	student → *course*					*faculty* → *project*				
cornell	**0.87**	0.61	**0.87**	0.84	0.80	0.19	0.04	**0.24**	0.18	0.18
texas	0.55	0.54	**0.77**	0.62	0.48	0.24	0.07	**0.29**	0.07	0.12
washington	0.67	0.64	**0.76**	0.69	0.44	0.40	0.11	**0.48**	0.29	0.18
wisconsin	0.75	0.73	**0.88**	0.77	0.55	**0.28**	0.07	0.27	0.20	0.21

not be completely clear way of ranking approaches for cases where curves intersect on several points. In order to summarize individual performances with a single statistic, we computed the area under each precision/recall curve (with linear interpolation between points). Results are given in Table 1. Numbers in bold indicate the algorithm with the highest area. The dominance of RBSets should be clear.

Silva et al. (2007) describe another application of RBSets, in this case for symmetric protein-protein interactions. In this application, there are no individual object features on which Cosine and SBSets can rely (every X^{ij} measures a pairwise feature), and RBSets performs substantially better.

5 CONCLUSION

We have emphasized the process of analogical reasoning as a retrieval of similar relationships, and presented a probabilistically sound approach for this problem. There is of course much more to analogical reasoning than calculating the similarity of complex relational structures. For instance, there is the issue of judging *how significant* the similarity is. Considering that the retrieved objects might be of a very different nature than those in the query set, one might also want to *explain* why the relations are judged to be similar. Ultimately, in case-based reasoning and planning problems (Kolodner, 1993), one might have to *adapt* the similar structures to solve a new case or plan.

One should see the contribution of this paper as a step towards a formal measure of analogical similarity. Much remains to be done to create a complete analogical reasoning system, but the described approach has immediate applications to information retrieval and exploratory data analysis.

References

E. Airoldi, D. Blei, E. Xing, and S. Fienberg. Mixed membership stochastic block models for relational data with application to protein-protein interactions. *Proceedings of the International Biometrics Society Annual Meetings*, 2006.

M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the World Wide Web. *Proceedings of AAAI'98*, pages 509–516, 1998.

S. Džeroski and N. Lavrač. *Relational Data Mining*. Springer, 2001.

R. French. The computational modeling of analogy-making. *Trends in Cognitive Sciences*, 6, 2002.

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

Z. Ghahramani and K. Heller. Bayesian sets. *18th NIPS*, 2005.

T. Jaakkola and M. Jordan. Bayesian parameter estimation via variational bounds. *Statistics and Computing*, 10:25–37, 2000.

D. Jensen and J. Neville. Linkage and autocorrelation cause feature selection bias in relational learning. *Proceedings of ICML*, 2002.

C. Kemp, J. Tenenbaum, T. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. *Proceedings of AAAI'06*, 2006.

J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.

C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. In press, 2007.

Z. Marx, I. Dagan, J. Buhmann, and E. Shamir. Couple clustering: a method for detecting structural correspondence. *JMLR*, 3:747–780, 2002.

R. Memisevic and G. Hinton. Multiple relational embedding. *18th NIPS*, 2005.

A. Popescul and L. H. Ungar. Structural logistic regression for link analysis. *Multi-Relational Data Mining Workshop at KDD-2003*, 2003.

R. Silva, E. Airoldi, and K. Heller. Small sets of interacting proteins suggest latent linkage mechanisms through analogical reasoning. *Gatsby Technical Report, GCNU TR 2007-001*, 2007.

P. Turney and M. Littman. Corpus-based learning of analogies and semantic relations. *Machine Learning Journal*, 60:251–278, 2005.