

IBM Research Report

Multi-Relational Learning for Genetic Data: Issues and Challenges

Claudia Perlich

IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Srujana Merugu

University of Texas at Austin



Research Division

Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Multi-Relational Learning for Genetic Data: Issues and Challenges

Claudia Perlich
IBM T.J. Watson Research Center
cperlich@stern.nyu.edu

Srujana Merugu
University of Texas at Austin
merugu@ece.utexas.edu

ABSTRACT

We present ongoing research on applying statistical relational learning techniques, in particular, propositionalization, to the challenging and interesting real-world domain of functional gene classification of the Yeast genome *Saccharomyces Cerevisiae*. The main objective of this paper is to identify and describe the structural and statistical properties of this domain and examine how they conflict with the assumptions of the traditional relational learning approaches. Such properties are, in fact, shared by many relational application domains and potential solutions will be of interest far beyond the particular genetic application. We also report some preliminary experimental results on potential solutions for overcoming the limitations of our modeling approach by extending the existing automated feature construction strategies to accommodate the specific domain properties.

1. MOTIVATION AND INTRODUCTION

The field of multi-relational learning has progressed considerably in the last decade of active research. However, it has yet to be clearly demonstrated that the available tools can handle large scale real-world problems that involve noisy, sparse and complex data. One of the main reasons for this situation is the lack of good benchmark sets in the relational modeling field that would allow a deeper understanding of the relative merits of different approaches. Recently, the ILP 2005 Conference has introduced an ILP Challenge based on a genetic dataset (Yeast genome) in order to assess the existing ILP and relational learning techniques. This genetic dataset has a number of properties that render it a suitable candidate for benchmarking: public availability, strong scientific interest in good answers, reasonable size (about 4000 labeled entities), limited predictability (so far) with lots of room for improvement, and challenging structural properties that also appear commonly in other relational domains. This domain may be of particular interest for comparisons of ILP approaches (see for instance the work

by Clare [4] on this domain) and the recent statistical approaches; it is at the same time very difficult (it appears noisy as long as we have not discovered the true underlying relationships), as well as contains complex and highly structured background information.

The objective of this paper is two-fold. First, we wish to promote the genetic domain as an interesting playground for all the new relational learning techniques and secondly, we present an initial exploration of the challenging issues and potential solutions. Rather than attempting a direct performance comparison with other relational learners at this early stage of our work, we leave it to the specialists to demonstrate the capabilities of probabilistic relational models, relational network classifiers, etc. We cannot hope to do these approaches justice as that will require adjustments and transformations of the data representation to produce optimal results. Instead, we will discuss in general terms the challenging properties of this real-world domain and explore potential approaches to address them in the framework of a propositionalization approach using our relational learning system ACORA.

2. DATA AND DOMAIN PROPERTIES

The relational structure of this particular genetic domain has multiple sources: the complexity of the object of interest (genes) and the multitude of heterogeneous information produced by a variety of genomic experiments such as genomic sequencing, micro-array testing, and homology analysis.

Before we discuss the specifics of this domain in more detail, we would like to point out that many of the higher-level characteristics are not unique to the genetic domain but are likely to be shared by many other relational modeling tasks such as fraud detection for instance. These characteristics impose severe restrictions on the expressive power and/or learnability of relational models as we will discuss further in Section 3.

In this article, we will confine our analysis to the data set provided for the ILP challenge, which contains a subset of all information on the genes in the yeast genome *Saccharomyces Cerevisiae*. The data is provided as prolog facts and can be readily represented in the form of the following relational tables:

Functional Class Annotation: The first table consists of the functional annotations for some of the genes based on a hierarchical functional classification scheme called *FunCat* [14]. Classes in *FunCat* are organized in a forest (multiple trees) and it is possible for the genes to have multiple

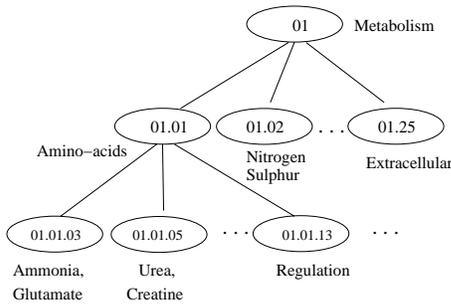


Figure 1: One of the trees in the FunCat hierarchical classification scheme

class labels. Figure 2 shows some of the classes in one such tree of the *FunCat* classification scheme and Table 1 contains a sample record from the functional annotation table.

Functional Annotation	
Gene ID	Class Labels
ytq0045	02.11
ytq0045	20.01.15
ytq0061	02.13.03
...	...

Table 1: Sample records from the functional annotation table. The strings 02.11, 02.13.03 and 20.01.15 correspond to different functional classes as shown in Figure 2.

Yeast-Yeast Homology: The second table contains information on the inter-gene similarity for the Yeast genome. In particular, a BLAST [1] distance score indicating the closeness of the match and the unlikelihood of the match being found by chance is provided for pairs of genes as shown in Table 2. Since the score is obtained from a comparison of the original nucleotide sequences of the genes, it is highly likely that similar genes (low score) would have common functional class labels. It is important to note that the similarity scores are available only for some pairs of genes and the absence of a score does not imply a zero or an infinite score.

Yeast-Yeast Homology		
Gene ID	Gene ID	Score
ytq0045	ytq0070	3e-05
ytq0045	ytq0065	0.065
...

Table 2: Sample record of inter-gene homology of Yeast genome.

Yeast-SwissProt Homology: The third table contains the similarity or homology scores between each gene in the Yeast genome and the proteins in the SwissProt [2] database. In addition to homology scores (Table 3), the data set contains a description of the SwissProt proteins in the form of another relational table (Table 4) indexed by the SwissProt ID. This description includes the keyword associated with

the protein, the type of organism in which it is produced, and other features such as molecular weight and sequence length.

Yeast-SwissProt Homology		
Gene ID	SwissProt ID	Score
ytq0045	o54069	2e-89
ytq0045	p07657	5e-32
...

Table 3: Sample record from the Yeast-SwissProt homology table.

SwissProt Description				
SwissProt ID	Keyword	Category	Weight	Length
o00086	repeat	candida	56239	521
...

Table 4: Sample record from the SwissProt description table.

Secondary Protein Structure: The data set also contains the secondary structure of the proteins synthesized from each of the genes in the Yeast genome. This secondary structure is actually not observed but rather predicted by PROF [10], a model that uses as input the primary gene sequence (which was not provided). Predicting secondary structure is a tough scientific problem by itself. This secondary structure is represented by a sequence of three symbols a, b, c corresponding to the structural components alpha helix, beta sheet and random coil respectively. Figure 2 shows the secondary structure of the protein generated by one of the Yeast genes. Each of these protein sequences is represented in the form of multiple records corresponding to homogeneous sections by specifying the order of occurrence, the type and the length of the section as in Table 5. The total length of the protein sequence varies from gene to gene so that the number of records associated with each gene also varies.

Secondary Protein Structure			
Gene ID	Order	Component	Length
ytq0045	1	c	1
ytq0045	2	a	20
ytq0045	3	c	3
ytq0045	4	b	5
ytq0045	5	c	11
ytq0045	6	a	24
...

Table 5: Sample records from the secondary protein structure table.

On a higher level that abstracts from this particular domain, we can identify a number of challenging properties (some of which we will reconsider in Section 3 in terms of implicit statistical dependencies) that are likely to occur across a variety of relational domains:

- **Weighted links:** The two Homology tables do not capture a well-defined relationship between pairs of

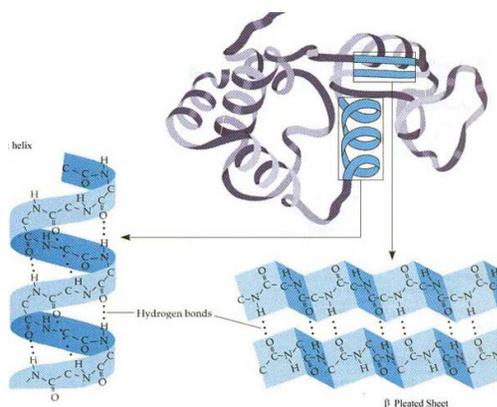


Figure 2: Secondary protein structure of one of the Yeast genes.

object in the traditional sense (i.e., Boolean interpretation that the relationship either exists or not), but rather a vaguely defined similarity relationship. Similarity is arguable a relationship, but the score is clearly an essential component. This case is somewhat similar to the common situation of relationship uncertainty, where the existence of a relationship is only known with some probability. The assumption is nevertheless that the relationship either existed or not. This distinction between probability and similarity may not be relevant in most cases, but it can lead to inconsistencies in the interpretation of missing links. What does it mean of a pair of object does not appear in a relationship table with link uncertainty? It is reasonable to argue, that we have no evidence for a relationship and therefore the best guess of a probability is zero. This may not be true for our case of a similarity relationship, here a pair missing from the full cross product does not need to be dissimilar.

- **Sparse data:** A closer look at the Homology tables reveals a high degree dispersion. Some genes have many entries, many only one. This is somewhat unexpected given that we could expect to see the full cross product of pairs. So the dispersion is clearly not natural and there must be the result of a non-random selection process. This is particularly concerning as we know that degree dispersion causes modeling bias [6] and that missing pairs have no good default explanation (e.g., highly dissimilar).
- **Ordered data:** The secondary structure is a prime example of ordered data. Since relational databases have only a set interpretation, the order is expressed by introducing a numeric field that keeps track of the relative position. Domains with similar structure include time series and sequences of time-stamped events. Sequences translate statistically into strong dependencies between attributes. Modeling techniques that assume class-conditional independence will be hurt severely.
- **Multiple class-hierarchies:** The particular class-structure of the genetic domain is not a relational property and therefore a general topic of machine learning.

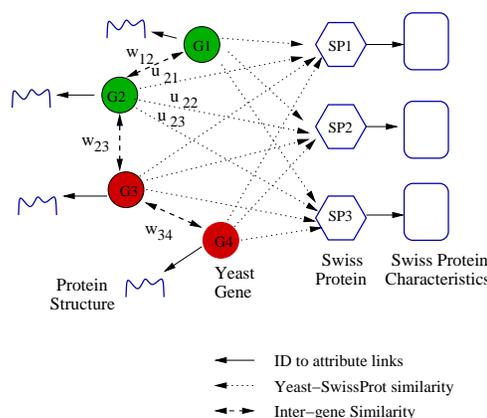


Figure 3: A graphical representation of the various components of genetic data.

However, the role of collective inference [7] may become more relevant for such tasks.

3. METHOD AND CHALLENGES

Many relational learning approaches appear suitable for the classification task in this domain. A very extensive study of this domain was done by Amanda Clare in her Ph.D. thesis combining a variety of techniques. Her approach follows essentially a “domain expert propositionalization” outline where she uses knowledge about the domain in combination with various (including first order) machine learning techniques for feature construction and finally uses C4.5 for classification. Similar to Clare’s work, we use a propositionalization approach that constructs features from the information in the background tables and finally estimates a traditional classification model (e.g., decision tree of logistic regression). The details of the ACORA system can be found in [12]. The algorithm uses breadth-first search (the desired depth is set by the user) over all possible joins on identifier attributes starting from the target table. For each join, it aggregates all sets of attributes independently using both traditional aggregation operators like average, count, etc., as well as estimating class-conditional distributions of the attributes (similar to a naive Bayes classifier). Such class-conditional distributions are particularly useful for categorical attributes with many possible values, where the COUNT or MODE operators can capture only limited discriminative information. The features that are constructed from distributions include vector distances (Cosine, Euclidean, etc.) and counts for a subset of particularly predictive values.

ACORA can also join back to the target table and thereby take explicit advantage of known class labels (similar to the approach of the relational neighbor classifier [8]), but ACORA does not perform any further collective inference [7].

3.1 Issues

Although ACORA can incorporate the provided information and appears to be appropriate for the gene classification task, it makes three implicit assumptions that limit its expressive power severely [11] and will make ACORA and *any approach that makes similar assumptions* surely fall short on this domain:

- Class-conditional independence between the attributes of related objects
- Bags of related objects are implicitly assumed to be random samples
- Relationships are known with certainty

Why do these assumptions hurt us so much? Let us revisit the available background information:

- Secondary structure of the synthesized proteins
- Similarity between gene and objects and their properties
- Similarity between genes

Secondary Structure: Recall that the representation of the secondary structure is simply a table that for each gene has a list of structural components a, b, c, the order of the components and the length. If the rows are treated as random samples and each of these three attributes is aggregated independently (e.g., using the average for the two numeric attributes and the MODE for the structure component), the actual sequence information is lost. The only information that is available to the classification model after the aggregation is the number of components (the average of the order number is equal to the number of components divided by 2), the average length of the components, and the most dominant component (typically the unclassified random rest c). We have no reason to believe that any of this information will be able to discriminate the functional role of a protein. The sequence was captured in the data simply by an order field. Such a representation violates both assumptions: the fields of the table are clearly not independent and the particular rows are certainly not a random sample.

Similarity: The similarity information violates the assumption that links are known with certainty (and also class-conditional independence). In this case an entry in the Yeast-Yeast Homology table of the form Gene ID, Gene ID, Score is not even a relationship in the normal sense. It does not mean that the two genes have anything to do with each other. In fact, an occurrence of a pair of genes in this table does not need to mean anything. From our analysis of the data, it simply means that somebody at some point ran a query to determine the similarity (with whatever intention) and added the value to the database. It is, therefore, unclear whether the occurrence of a pair in this table has anything to do with the functionality of either of the two genes. What may be relevant, however, is the similarity score itself, if one is willing to believe (as we typically are in machine or human learning) that similar things will behave similarly. In particular, we would expect that a pair of gene with a high similarity score will result in highly similar proteins that probably have a similar functionality. But this is unfortunately not the interpretation that a standard relational learner like ACORA would apply. The class-conditional independence of the distance attribute and the object identifier will obscure any meaning. In order to take advantage of the distance, the system needs to know that a particular field is actually a measure of strength rather than some other numeric value (e.g., price of a product). It is also important to observe that such a similarity indicator affects

not only the interpretation of the particular join, but also all further links resulting from this join. Our domain has a table that contains all the db references. Since it was a 1:n relation with the object it had to be put into a separate table. Such references are clearly more relevant if the similarity of the gene and the object corresponding to the references was high.

3.2 Potential Solutions

Before we take a close look at the potential solutions, let us clarify why assumptions are necessary and often good. In other words: why cannot we just throw them all out. In order to build models for prediction and generalization, we need a distance metric on the object space. We can only make a prediction for a new object that was never seen before, if we (or the model) can say that it is more similar to those previously seen objects than to others. If the objects are presented in a numeric feature vector, we can for instance assume a Euclidean space or a transformation thereof. If the objects are very complex, we have to think more explicitly about the definition of similarity. And this is where assumptions are vital. They define a similarity. Without our assumptions, each and every object is unique and equally dissimilar to all others. Without assumptions, we cannot generalize beyond what we have seen already. In particular, representing the secondary structure in its original form as a long string of the form “abcbcacbabacabcaba” is typically not suitable for learning either.

We can follow one of three approaches to address the mismatch between the learner and the domain properties:

1. Change the assumptions of the relational learner to match the properties of the domain,
2. Change the domain representation to match the assumptions of the learner,
3. Parameterize the learner to allow the user to specify the appropriate notion of distance.

The latter approach was adopted in many ILP implementations in the form of a declarative language bias[5] where it was up to the user to define the search space (and thereby his assumptions about relevant dependencies). The first approach would lead us from a more or less general relational learner to an increasingly domain specific learning tool that is only suitable to the particular application and is similar to the route taken by Clare. It is likely to achieve the best results since it allows us to optimally accommodate any arbitrary domain property.

In the following section, we will explore the second option and try a number of simple changes of the domain representation and analyze the effect of such changes on the predictive performance.

4. EXPERIMENTAL RESULTS

In this paper, the main focus of our discussion is the characteristics of the domain and not the complexity of the classification task. Therefore, we selected initially a single binary classification task that involves identifying genes that synthesize the proteins involved in Cellular Transport (class 20). Genes with classes below this node (e.g., 20.03.10) were also labeled as positive. A subset of 3000 genes was used for

training and 1000 for testing. The class prior was 0.76. All results in the following section are out of sample performance on the same test in terms of classification accuracy and ranking (AUC [3]). ACORA constructed for each experiment a standard set of features including COUNT, average for numeric attributes, cosine and Euclidean distances between the object and the class-conditional distribution and finally, the counts for the 5 most discriminative values (having the largest absolute difference between the two class-conditional probabilities). For further details see [12].

ACORA finally performs a feature selection and estimates a logistic classification model, but our experiments using C4.5 showed no significant differences.

4.1 Naive Approach

Initially, we pretended to know nothing about the domain and the meaning of the fields in our database. In particular, we ignored the mismatch between the semantics of the domain and the assumptions of the method (independence of the attributes, bags are interpreted as random samples and aggregates as properties of the distribution). The results of these experiments are shown in Table 6 for the different information types. Gene Classes uses the class labels of other yeast genes that were linked through the Homology table.

Information used	Accuracy	AUC
Structure	0.788	0.693
Protein Homology	0.802	0.667
Genes Homology	0.800	0.661
Gene Classes	0.813	0.724
Protein Properties	0.804	0.704
All Information	0.810	0.732

Table 6: Results ignoring the properties of the domain. The prior was 0.767.

4.2 Ordered sets and sequences: Learning from secondary structure

Consider the example of ordered sequence information of the secondary structure of gene ydq0045. The secondary protein structure is $ca^{20}c^3b^5c^{11}a^{24}b^3c^{11}a^{35}c^{15}$, where z^n denotes n repetitions of the letter z . This information is provided in the structure table shown in Table 7.

Secondary Protein Structure			
Gene ID	Order	Component	Length
ytq0045	1	c	1
ytq0045	2	a	20
ytq0045	3	c	3
ytq0045	4	b	5
ytq0045	5	c	11
ytq0045	6	a	24
ytq0045	7	b	3
ytq0045	8	c	11
ytq0045	9	a	35
ytq0045	10	c	15

Table 7: Original representation of the secondary protein structure for gene ydq0045.

The problem with this representation is the mismatch with the independence assumptions that ACORA makes.

In particular, it will assume that the length attributes can be aggregated separately from the component field. In addition, the order that is captured by the order field is lost under the implicit assumption of random sample using the MEAN or reflexivity using SUM to aggregate the numeric fields. The only information that ACORA can extract is the number of a, b, c components, the total number of components and the average length of the components.

There are a number of possible approaches to maintain some of the sequence and even length information, all of which will extract shorter sub-sequences. Clare followed a sophisticated approach for sequences and used PolyFARM [4], (an outgrow of the relational learner WARMR [13] in the style of and extended declarative language) to extract 20.000 frequent patterns.

We explored 3 simpler ad-hoc methods:

1. Initially we focus only on the sequence of the protein components and ignore the component length. We extract all sub-sequences of a particular length using a sliding window as shown in the left Table 8. This approach is very similar to time series analysis where one includes lagged observations.
2. We code the length implicitly by repeating the component letter before selecting sub-sequences. However, this approach produces mostly sub-sequences of single letter repetitions.
3. Finally we code the sequence with the explicit length information. In this case we produce many unique sub-sequences with low coverage across genes. As an alternative we applied a log (to the base 3) transformation of the length field as shown in the right Table 8. This transformation is somewhat consistent with the assumption that only the relative size (short, medium, long) matters, not the specific number of molecules.

Sub-sequences		Sub-sequences	
Gene ID	Without length	Gene ID	Log length
ytq0045	cac	ytq0045	c1a3c2
ytq0045	acb	ytq0045	a3c2b2
ytq0045	cbc	ytq0045	c2b2c3
ytq0045	bca	ytq0045	b2c3a3
ytq0045	cab	ytq0045	c3a3b2
ytq0045	abc	ytq0045	a3b2c3
ytq0045	bca	ytq0045	b2c3a4
ytq0045	cac	ytq0045	c3a4c3

Table 8: Alternative representations of the secondary protein structure using sub-sequences of three components with (right) and without (left) length information.

The performances of the models using only the new structure representations are shown in Table 9. We did not include the results for the length adjustment by repeated components because they did not look any different.

In summary, we do not seem to be able to find any additional information above and beyond what the naive approach using the original table has achieved. We are unfortunately not able to compare our results to the ones obtained

Structure Representation	Accuracy	AUC
Component length 2	0.791	0.685
Component length 3	0.786	0.688
Component length 4	0.792	0.689
Component length 5	0.796	0.686
Component length 6	0.80	0.683
Component length 7	0.787	0.673
Component length 8	0.782	0.674
Component length 9	0.791	0.681
Component log length 2	0.775	0.656
Component log length 3	0.794	0.667
Component log length 4	0.792	0.641
Component log length 5	0.784	0.646
Component log length 6	0.783	0.666
Component log length 7	0.782	0.675
Component log length 2	0.79	0.677
Component log length 2	0.785	0.658

Table 9: Results using the secondary structure information for classification task 20.

by Clare. She does report accuracy results using the structural information, but not on this particular classification task (this may indicate that it was not predictive for her either).

4.3 Similarity and relationship uncertainty

The presented case of object similarity is only one particular type of weighted relationships. More commonly, we face domains with actual relationship uncertainty, where the link is annotated by a probability. In either case it would be wrong to ignore the weight or to treat it as an independent property. Stochastic Logic Programming [9] is one of the earlier approaches that incorporate explicit probabilistic information into relational reasoning. Note that the weight of a BLAST[1] distance score can be interpreted as the likelihood of random match. We consider again three different approaches to account for the weights:

1. Select only pairs where the score was below a given cutoff. The biological literature on the interpretation of BLAST values suggests a cutoff of 0.000001 in combination with the percentage length of alignment. Unfortunately, we do not have the alignment information but still used this cutoff. One shortcoming of this approach is the fact that for a number of genes, no object had a score smaller than the cutoff.
2. Select for each gene a subset of the n objects with the smallest score, where we tried values of 10, 20, and 50 for n .
3. Treat the similarity score (we used 1-score) as a probability and use it explicitly in the construction of the aggregates and weight the evidence accordingly. This can no longer be accomplished by simply changing the data representation and required adjustments to ACORA.

The results of our experiments are shown in Table 10. We now include not only the information from one background table as in the structure experiments, but all of the information that is available through a Homology link. This includes in particular the object properties and the class labels of other Yeast genes in the training set.

Method	Accuracy	AUC
Stochastic aggregation	0.812	0.737
Subset size 10	0.834	0.792
Subset size 20	0.828	0.769
Subset size 50	0.823	0.758
Subset with cutoff	0.812	0.744

Table 10: Results using the Homology Tables for classification task 20.

We now see a clear improvement over the initial results in Table 6. Both the accuracy and (even more so) the ranking performance have increased significantly. An interesting observation from this set of experiments is that the least sophisticated approach using a small and constant neighborhood performs better (significantly) than other approach. The neighborhood size has a clear negative effect on both performance measures.

4.4 Performances across classes

We finally present in Table 11 the predictive results for all 16 top classes of the classification trees using 3000 training examples and 1000 test examples with all available information.

Class	Prior	Accuracy	AUC
01	0.653	0.785	0.806
14	0.747	0.808	0.8
11	0.766	0.787	0.781
20	0.767	0.832	0.792
16	0.782	0.8	0.723
10	0.783	0.78	0.75
42	0.839	0.839	0.673
32	0.872	0.875	0.641
34	0.884	0.88	0.663
12	0.899	0.923	0.81
43	0.908	0.908	0.731
02	0.913	0.92	0.746
30	0.939	0.938	0.793
40	0.943	0.942	0.785
18	0.951	0.96	0.746
38	0.973	0.973	0.93

Table 11: Results for all 16 top classes.

The generalization results show very different levels of predictability. The AUC performances range from only 0.641 (for class 32) up to 0.93 (for class 38). We also see a strong skew in the class priors ranging from 0.653 for the most common class 01 to 0.973 for the least common class 38. Both characteristics may have a strong impact on the relative performance of different relational learning methods and provide a nice space for benchmarking and performance analysis.

5. SUMMARY AND CONCLUSIONS

The objective of this paper is mostly to motivate a discussion of current limitations of relational models that are related to specific domain properties and in particular, violate the various independence assumptions. We feel that such properties are common across a large variety of relational domains and need further attention. Our somewhat

ad-hoc solutions could provide some predictive power, but there are clearly better ways to tackle the raised issues and a more thorough empirical analysis is called for. As a secondary objective, we would like to invite more researchers in the field to consider this genetic dataset as a platform for performance comparisons. We currently do not have a sufficient base of benchmark problems in combination with published results to start a more general analysis of the relative strength and weaknesses of our relational approaches. More work on this domain could be another contribution to a better understanding of the capabilities of relational learning approaches.

6. REFERENCES

- [1] S. F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.
- [2] A. Bairoch and R. Apweiler. The swiss-prot protein sequence database and its supplement trembl. *Nucleic Acids Research*, 28:45–48, 2000.
- [3] A.P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.
- [4] A. Clare and King R.D. Data mining the yeast genome in a lazy functional language. In *Practical Aspects of Declarative Languages (PADL'03)*, 2003.
- [5] L. Dehaspe and L. De Raedt. DLAB: A declarative language bias formalism. In *International Symposium on Methodologies for Intelligent Systems*, pages 613–622, 1996.
- [6] D. Jensen and J. Neville. Linkage and autocorrelation cause feature selection bias in relational learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*, 2002.
- [7] D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 593–598, 2004.
- [8] S.A. Macskassy and F. Provost. A simple relational classifier. In *Proceedings of the Workshop on Multi-Relational Data Mining (KDD)*, 2003.
- [9] S.H. Muggleton. Stochastic logic programs. In *Proceedings of the 5th International Workshop on Inductive Logic Programming*, page 29. Department of Computer Science, Katholieke Universiteit Leuven, 1995.
- [10] M. Ouali and R.D. King. Cascaded multiple classifiers for secondary structure prediction. *Protein Science*, (9):1162–1176, 2000.
- [11] C. Perlich and F. Provost. Aggregation-based feature invention and relational concept classes. In *Proceedings of the Ninth ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2003.
- [12] C. Perlich and F. Provost. ACORA: Distribution-based aggregation for relational learning from identifier attributes. *Forthcoming in Journal of Machine Learning*, 2005.
- [13] K. Ross, D. Ashwin, and S. Dehaspe. WARMR: A data mining tool for chemical data. *Journal of Computer Aided Molecular Design*, (15):173–181, 2001.
- [14] A. Ruepp, A. Zollner, D. Maier, K. Albermann, J. Hani, M. Mokrejs, I. Tetko, U. Guldener, G. Mannhaupt, M. Munsterkotter, and HW. Mewes. The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, 32:5539–5545, 2004.