

Approaching the ILP 2005 Challenge: Class-Conditional Bayesian Propositionalization for Genetic Classification

Claudia Perlich

IBM T.J. Watson Research Center
1101 Kitchawan Road, Route 134
Yorktown Heights, NY 10598

Abstract. This report presents a statistical propositionalisation approach to relational classification and probability estimation on the genetic ILP Challenge domain. The main difference between our and existing propositionalisation approaches is its ability to construct features from categorical attributes with many possible values and in particular the object identifiers. Our classification and ranking results on the genetic domain are promising but will require further evaluation in comparison with other relational models.

1 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-life problems that involve noisy, sparse, and complex data. The ILP Challenge 2005 provided a genetic dataset (Yeast genome) as a platform to assess performances of existing ILP and statistical relational learning techniques. This report presents the results of a propositionalisation approach that extends traditional aggregation operators like COUNT, MEAN or MODE (as considered for example in [4]) with class-conditional Bayesian aggregates. Our approach was developed for learning from networked domains and for aggregation of categorical attributes with high cardinality. The genetic dataset can be represented as a network of genes where the weighted edges correspond to the homology information. How good our results are relative to other approaches will ultimately be determined by the outcome of the challenge. So rather than attempting a direct performance comparison with other relational learners at this stage, we will leave it to the specialists to demonstrate the capabilities of their methods. We cannot hope to do these approaches justice, as they will require adjustments, specific data representation, construction of background knowledge, and language biases to produce optimal results.

2 ACORA: Density-Based Propositionalization

Our relational learner ACORA (Automated Construction of Relational Attributes) follows a propositionalisation approach [4] to convert the information in a multi-relational database into a propositional feature-vector representation by constructing

<p>Input: The domain specification (list of tables T_k, attributes T_{kj}, and types), and a database DB including a target table T_t with labeled training objects t_t.</p> <ol style="list-style-type: none"> 1. Read specification and build domain graph G 2. Initialize breadth-first list L with target table: $L = \{T_t\}$ 3. Initialize feature table $F = \text{non-identifier attributes}(T_t)$ 4. Loop 5. $T_c = \text{First}(L)$ 6. Foreach table T_g in DB linked to T_c in G through some identifiers T_{ck}, T_{gj} 7. $J = \text{Join } T_c \text{ and } T_g \text{ under the condition } T_{ck}=T_{gj}$ 8. Foreach attribute $T_{ga}, a \neq j$ 9. Estimate class-conditional distributions CD 10. Foreach target observation $t \in T_t$ 11. Find bag of related attribute values $B(T_{ga}, t)$ 12. Foreach applicable aggregation operator A_s 13. Construct $A_s(B(T_{ga}, t), CD)$ 14. Append aggregates A_s as new columns to feature table F 15. End Foreach 16. End Foreach 17. Append the join result (J) to the end of list L 18. End Foreach 19. If (stopping criterion) GOTO 22. 20. End Foreach 21. End Loop 22. Feature selection and model estimation from F

Fig. 1. Pseudocode of the ACORA Algorithm where T_k is a table and T_{kj} an attribute and t_{ki} an attribute value.

attributes from background relations. ACORA is dominantly used for binary classification and probability estimation but can also address regression tasks. The system is domain independent and fully automated. Aside from the database schema and the attribute types, it does not require any form of declarative bias to structure the search space. The algorithm proceeds in four steps (for further details see [9]):

1. Exploration of related entities using joins,
2. Feature construction through aggregation,
3. Feature selection, and
4. Model estimation.

The exploration step joins the target table with related background tables. The aggregation methods are applied to the results of the join, followed by a standard selection procedure that identifies predictive features, and finally a classification model is estimated from the feature-vector representation. Fig. 1 presents ACORA's algorithm in pseudocode. Since both feature selection and model estimation are standard machine learning procedures we will not describe them further.

Exploration: Figure 2 shows for the illustration of the algorithm a small subset of the genetic domain. We will assume that it is known which attributes are identifiers

Target Table		Homology			Protein Data			
Gene ID	Class	Gene ID	Protein ID	Score	Protein ID	Length	Weight	Category
ytq0045	1	ytq0045	p29875	4e-83	p29875	226	26782	lasius
ytq0061	0	ytq0045	o98048	2e-10	o98048	228	26620	yponomeuta
ytq0132w	0	ytq0061	o29875	0.63	p33501	378	43535	anopheles
...

Fig. 2. Data example for aggregation

and can be used for joins between tables. Given the identifiers (in the example Gene ID and Protein ID), any relational domain can be represented as a graph where the vertices correspond to tables and the edges connect tables that share identifier attributes of equal types. ACORA’s search explores this domain graph starting from the target table using breadth-first search. Since the graph may be cyclic, it is necessary to impose a stopping criterion such as maximum depth or the total number of joins. Cycles also imply that some chain of joins can lead back to the target table. This raises the question whether the classes of training data itself become part of the background knowledge. This view has been argued previously (see for instance [5]) for networked domains like this genetic domain.

Aggregation: Every search step is a sequence of joins that produces a table with multiple entries (if at least one of the joins captures a one-to-many relationship) for each target object. Consider for instance a sequence of two joins from the Target table to Homology (on Gene ID), followed by a join to Protein Data on the Protein ID. For each gene in the target table (e.g., ytq0045) we find a bag of related object and their attributes (e.g., the bag of the ‘Length’ values that are related to ytq0045 is {226,228}, the bag of ‘Category’ values is {lasius,yponomeuta}). ACORA aggregates each such bag of attribute values separately and estimates in addition to the typically used SQL operators COUNT, MEAN, and SUM, for each attribute the two class-conditional distributions. This procedure is closely related to a naive Bayes classifier. Given these two distributions, we can now define the class-conditional likelihood of a particular bag of attributes as a new feature. As an alternative to likelihood, any vector distance measure (ACORA uses cosine, Euclidean, and Mahalanobis) between the actual bag and a class-conditional density in vector format can be used for feature construction. This aggregation approach has two main advantages: it focuses on the *discriminative* information and it can be applied to arbitrary attributes including categoricals with many possible values and even identifiers. The main properties of the overall approach are:

- Domain independent general purpose approach,
- Full automation without declarative language,
- Aggregation of attributes with high cardinality including identifiers,
- Aggregation of known training class labels of related object.

The advantage of a fully automated general purpose approach without declarative language comes at the price of restrictive assumptions on the underlying concept:

- Class-conditional independence between bags of related objects,

- Class-conditional independence between attributes of related objects,
- Related objects are bags, i.e., random samples without internal order.

It is important to observe that based on the taxonomy in [8], ACORA is much less expressive than most ILP models. However, the approach has previously shown impressive generalization performances on noisy networked domains including direct marketing, terrorist identification, and citation-based document classification [9].

3 Data Representation

The genetic domain is originally provided in the form of PROLOG facts and can be readily represented in the form of a relational database consisting of 6 tables:

1. Target table of functional annotation (Gene ID, Class)
2. Gene homology (Gene ID, Gene ID, Similarity Score)
3. Protein homology (Gene ID, Protein ID, Similarity Score)
4. Protein description (Protein ID, Keyword, Category, Molecular Weight, Length)
5. Protein database references (Protein ID, Reference)
6. Gene structure (Gene ID, Order, Component, Length)

The two identifier attributes linking the entities in the tables are Gene ID and Protein ID. The target table contains the functional annotation classes based on a hierarchical scheme called *FunCat* [12]. Classes in *FunCat* are organized in a forest (multiple trees) and it is possible for genes to have multiple class labels. The two homology tables contain information on the inter-protein similarity for the Yeast genome and other proteins in the SwissProt database. In particular, a BLAST [1] distance score indicates the closeness of the match and the unlikeliness of the match being found by chance. It is important to note that the similarity scores are available only for some pairs. The absence of a score does not imply a zero or an infinite score. The SwissProt description table documents various properties of the proteins in the SwissProt collection. The secondary protein structure was predicted by PROF [7], a model that uses as input the primary gene sequence (which was not provided). The 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.

Given ACORA's assumptions we have to adjust this initial representation somewhat. Most importantly, ACORA assumes that matching identifiers reflect the existence of a relationship. This is not really the case in the homology tables. Each row captures a similarity relationship of certain strength. In particular, a score that is larger than 1e-06 is considered in the biological literature an indicator of no match. It would clearly be wrong to ignore the weight or to treat it as an independent attribute. Stochastic Logic Programming [6] is one of the approaches that can incorporate probabilistic information as weights into relational reasoning. We initially explored three different approaches to account for the weights:

1. Select only pairs where the score was below the 1e-06 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 1 minus similarity score as a probability and use it explicitly in the construction of the densities to weight the evidence accordingly.

Class	01	02	10	11	12	14	16	18
Prior	0.653	0.913	0.783	0.766	0.899	0.747	0.782	0.951
Accuracy	0.785	0.92	0.78	0.787	0.923	0.808	0.8	0.96
AUC	0.806	0.746	0.75	0.781	0.81	0.8	0.723	0.746

Class	20	30	32	34	38	40	42	43
Prior	0.767	0.939	0.872	0.884	0.973	0.943	0.839	0.908
Accuracy	0.832	0.938	0.875	0.88	0.973	0.942	0.839	0.908
AUC	0.792	0.793	0.641	0.663	0.93	0.785	0.673	0.731

Table 1. Generalization performances on a size 1000 test set for the 16 top classes.

Surprisingly, the second approach using only a small (10) and constant neighborhood performed significantly better than the other two approaches and we curtailed the homology tables accordingly. The secondary structure violates two assumptions, 1) the notion of an unordered bag of objects and 2) the class-conditional independence between the component attribute and the length. Clare used PolyFARM [3], an outgrowth of the relational learner WARMR [11], to extract 20,000 frequent patterns for this problem. After some inconclusive exploratory work we decided to ignore the length and extract sub-sequences of three components.

4 Experimental Results

We converted the multi-class task into 18 binary classification tasks. We present here only the results for 16 top level classes; two additional top classes had only 5 and 54 positive examples. ACORA constructed for each experiment a standard set of features including COUNT, and MEAN, and cosine and Euclidean distances between the bag and the class-conditional distribution for all attributes including the object identifiers. The stopping criterion for the breadth-first search is a depth limit of 3 allowing the aggregation of known class labels. ACORA estimates a logistic classification model on the selected features. The decision tree C4.5 [10] provided inferior results.

For the out-of-sample performances in Table 4 we used a subset of 3000 genes for training and 1000 for testing. We evaluate our models both in terms of accuracy (percent correct predictions) and area under the ROC [2] as a measure of the ranking performance. The AUC is bounded below by 0.5 for a random model and above by 1 for a perfect model. The score has a nice probabilistic interpretation; it corresponds to the probability that any pair of observations with different class labels is ranked in the correct order.

The results in Table 4 show that ACORA improves in terms of accuracy over the class prior in 9 out of 16 domains. Accuracy is a problematic measure of model performance for noisy domains with highly skewed class priors. The AUC results on the other hand show that ACORA finds predictive information for all tasks, the performances are consistently far above 0.5 and for 4 tasks reach or exceed 0.8. The final performance comparison for the ILP challenge will use a different metric across all classes and is beyond the scope of this report.

5 Summary and Conclusions

We presented a statistical propositionalisation approach on the task of gene classification for the ILP Challenge 2005. Our approach can take advantage of the identities of genes and proteins with high similarity scores and can also explicitly incorporate the known class labels. In particular, the cosine distance to the class-conditional distributions of the class labels of genes with high similarity was consistently one of the best predictors. Other predictive features include the cosine and Mahalanobis distances to the distributions of the very similar proteins and gene. However, no single protein or gene was ever sufficiently predictive to be chosen by feature selection. An example of a single predictive value was the keyword “transmembrane” of very similar proteins for predicting class 20 (Cellular transport). The interpretation of the learned models is often not straight forward and may involve the manual comparison of the class-conditional densities. We have started to develop visualization tools to support model interpretation. The performances on the first-level classification tasks are promising but require further comparative analysis against other relational modeling techniques.

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.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.
3. 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.
4. A. Knobbe, M. De Haas, and A. Siebes. Propositionalisation and aggregates. In *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*, pages 277–288, 2001.
5. S.A. Macskassy and F. Provost. A simple relational classifier. In *Proceedings of the Workshop on Multi-Relational Data Mining (KDD)*, 2003.
6. 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.
7. M. Ouali and R.D. King. Cascaded multiple classifiers for secondary structure prediction. *Protein Science*, (9):1162–1176, 2000.
8. 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.
9. C. Perlich and F. Provost. ACORA: Distribution-based aggregation for relational learning from identifier attributes. *Forthcoming in Journal of Machine Learning*, 2005.
10. J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Los Altos, California, 1993.
11. 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.
12. 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.