Multi-class Protein Fold Recognition Through a Symbolic-Statistical Framework

Marenglen Biba, Floriana Esposito, Stefano Ferilli,
Teresa M.A. Basile, and Nicola Di Mauro

Department of Computer Science, University of Bari, Italy
{biba,esposito,ferilli,basile,ndm}@di.uniba.it

Abstract. Protein fold recognition is an important problem in molecular biology. Machine learning symbolic approaches have been applied to automatically discover local structural signatures and relate these to the concept of fold in SCOP. However, most of these methods cannot handle uncertainty being therefore not able to solve multiple prediction problems. In this paper we present an application of the symbolic-statistical framework PRISM to a multi-class protein fold recognition problem. We compare the proposed approach to a symbolic-only technique and show that the hybrid framework outperforms the symbolic-only one in terms of predictive accuracy in the multiple prediction problem.

1 Introduction

Proteins form the very basis of life. They are responsible for regulating a variety of activities in all known organisms, from replication of the genetic code to transporting oxygen or regulating the cellular machinery. Proteins accomplish their task by three-dimensional tertiary and quaternary interactions between various substrates such as DNA and RNA, and other proteins. Therefore knowing the structure of a protein is an essential prerequisite to gain a thorough understanding of the protein's function. However, once the protein sequence has been determined, deducing its unique three-dimensional native structure is a very hard task. For this reason, many efforts have been made to develop methods for predicting proteins' structure given their amino acid sequence. Important competitions such as CASP and CAFASP2 [1] have given rise to many computational methods for the protein structure prediction problem. Despite the large amount of effort expended, the protein folding or protein structure prediction problem remains largely unsolved. Thus, there is strong motivation to continue working on the many remaining open problems that the protein structure modeling area poses.

Protein folding is the process by which a protein assumes its characteristic functional shape or tertiary structure, also known as the native state. All protein molecules are linear heteropolymers composed of amino acids and this sequence is known as the primary structure. Most proteins can carry out their biological functions only when folding has been completed, because three-dimensional

F. Masulli, S. Mitra, and G. Pasi (Eds.): WILF 2007, LNAI 4578, pp. 666–673, 2007.

shape of the proteins in the native state is critical to their function. A particular fold is adopted by a certain protein sequence/structure following several constraints which can be local or global. Local signatures, which are those dealt with in this paper, relate to a short region that may involve a particular sequence or arrangement of secondary structures. Structural signatures are hard to classify and although several automated methods have been proposed, knowledge about structural signatures depends primarily on human expertise. However, with the increase of the number of protein structures, intensive efforts have been made for the development of automated methods.

In the field of machine learning, approaches such as artificial neural networks or hidden markov models have been applied successfully to several problems of molecular biology [2]. However most of these techniques, being not able to model long range interactions, have had their best results on sequence data, while the problem of dealing with the three-dimensional structure has not been tackled very much. On the other side, symbolic approaches based on first-order logic representations have the power to deal with such complex domains and are very suitable to model rich structures and relations between objects. One of these approaches is ILP (Inductive Logic Programming) [3] that learns rules from examples and background knowledge. This technique, being able to model relations, has been applied successfully to some problems in structural molecular biology [4]. However, a major drawback of this symbolic approach is the limited ability to handle uncertainty. Rules in ILP are deterministic and there is no way to handle the uncertainty that may characterize a certain problem.

In this paper, we consider a previous study [5] in the protein folding area that uses ILP to automatically discover structural signatures of protein fold and function. A problem that arises in this previous work is that of multiple predictions, i.e. an example which represents a protein domain is predicted to be in several folds. We apply to the same problem the symbolic-statistical framework PRISM [6] in order to solve the multi-class classification problem and show that the hybrid approach outperforms the symbolic one in terms of predictive accuracy.

The paper is organized as follows. In Section 2 we report a brief introduction of ILP and its application on the structural signatures performed in [5]. Section 3 presents PRISM as a symbolic-statistical framework. Section 4 presents the modeling of the problem in [5] in the framework PRISM and the experiments. Section 5 contains conclusions and future work.

2 Multi-relational Learning for Structural Signatures of Proteins

Multi-relational data mining applications in biological domains [4] have exploited the expressive power of logic to represent complex structures. As pointed out in [5], since structures consist of interactions among objects and sub-structures and since ILP is suitable to learn logical representations, it can be applied to problems encountered in protein structure. Moreover, one of the most powerful

advantages of ILP is that of using background knowledge and since great amount of knowledge has been gathered during years of research on protein structure, all this expert knowledge can be used in ILP to discover principles of protein fold. Another advantage of ILP is that rules are amenable to human interpretation.

In ILP, the model learned from the data is a set of rules. The data consist of the examples, while the background knowledge expresses what the expert already knows about a certain problem. An application of ILP to automatically discover the structural signatures of protein folds and function has been presented in [5]. In this work sets of rules were learned for each protein fold, in particular 59 signatures (rules) were learned from 20 populated folds. Positive examples were derived from SCOP [7] by selecting representative domains for the fold under study while the negative examples were derived by selecting domains from different folds of the same class where the classes are all-α, all-β, α/β and $\alpha + \beta$. For each positive and negative example, it was derived structural information (attributes such as total number of residues), relational information (adjacency of the secondary structure) and local information (such as average hydrophobicity of each secondary structure element and the presence of proline residues). In the following, we show part of the background knowledge that is used in the experiments to represent the three-dimensional structure information of the protein domains which represent the examples for the learning task.

adjacent(D, A, B, Pos, TypA, TypB): this predicate indicates that the secondary structures A and B are consecutive. Furthermore, their respective types are TypA and TypB each of which can be one of the known types of secondary structure. Pos is the serial number of the secondary structure element A. Helices and strands are numbered separately.

coil(A,B,Length): bounds Length to the length of the loop between secondary structures A and B or is true if loop has Length ±50%. A brief description of two of the signatures (rules) learned is given below, consisting of the Prolog representation and the corresponding translation in English where the symbol ":-" stands for "if... then...".

Rule (lambda repressor): The protein is between 53 and 88 residues long. Helix A at position 3 is followed by helix B. The coil between A and B is about six residues long.

fold('lambda repressor', X) :- total_length ($53 < X < 88$), adjacent(X, A, B, 3, h, h), length_loop(A, B, 6).

Rule(Rossman fold): Strand A at position 1 is followed by helix B. Strand C at position 6 is followed by helix D. The length_loop between A and B is about one residue long.

fold('NAD(P)-binding Rossmann-fold', X) :- adjacent(X, A, B, 1, e, h), adjacent(X, C, D, 6, e, h), length_loop(A, B, 1).

Since protein folding is a complex phenomenon, a problem that arises in [5] are multiple predictions. Many examples are predicted to be in different folds i.e. signatures of different folds explain the same example. For instance, the protein domain "d1hslb_" is predicted to be in three folds: "DNA-binding 3-helical bundle", "Periplasmic binding protein-like II" and "beta-Grasp" while in fact it belongs only to the fold "Periplasmic binding protein-like II". A large number of examples are involved in the multiple prediction problem hence a ranking mechanism is needed so that different folds can have different importance towards an example. We decided to use probability to model the uncertainty that arises when multiple predictions exist. On the other side, we want to preserve the expressive power of logical representations. Therefore, we need a framework that is able to provide expressive power and uncertainty handling. PRISM provides both the logic language and the ability to incorporate probability in logical descriptions. Moreover, providing learning capabilities for estimating parameters from observations (examples), it represents a suitable framework to deal with uncertainty when classifying examples with multiple potential predictions.

3 The Symbolic-Statistical Framework PRISM

PRISM (PRogramming In Statistical Modeling) [6] is a symbolic-statistical modeling language that integrates logic programming with learning algorithms for probabilistic programs. PRISM programs are not only just a probabilistic extension of logic programs but are also able to learn from examples through the EM (Expectation-Maximization) algorithm which is built-in in the language. PRISM represents a formal knowledge representation language for modeling scientific hypotheses about phenomena which are governed by rules and probabilities. The parameter learning algorithm [8], provided by the language, is a new EM algorithm called graphical EM algorithm that when combined with the tabulated search has the same time complexity as existing EM algorithms, i.e. the Baum-Welch algorithm for HMMs (Hidden Markov Models), the Inside-Outside algorithm for PCFGs (Probabilistic Context-Free Grammars), and the one for singly connected BNs (Bayesian Networks) that have been developed independently in each research field. Since PRISM programs can be arbitrarily complex (no restriction on the form or size), the most popular probabilistic modeling formalisms such as HMMs, PCFGs and BNs can be described by these programs.

PRISM programs are defined as logic programs with a probability distribution given to facts that is called basic distribution. Formally a PRISM program is $P = F \cup R$ where R is a set of logical rules working behind the observations and F is a set of facts that models observations' uncertainty with a probability distribution. Through the built-in graphical EM algorithm the parameters (probabilities) of F are learned and through the rules this learned probability distribution over the facts induces a probability distribution over the observations. As an example, we present a hidden markov model with two states slightly modified from that in [8]:

```
values(init,[s0,s1]).          % State initialization
values(out(_),[a,b]).          % Symbol emission
values(tr(_),[s0,s1]).         % State transition

hmm(L) :-                      % To observe a string L
        str_length(N),         % Get the string length as N
        msw(init,S),           % Choose an initial state randomly
        hmm(1,N,S,L).          % Start stochastic transition (loop)

hmm(T,N,_,[ ]) :- T > N,!.     % Stop the loop
hmm(T,N,S,[Ob | Y]) :-         % Loop: current state is S, current time is T
        msw(out(S),Ob),        % Output Ob at the state S
        msw(tr(S),Next),       % Transit from S to Next.
        T1 is T+1,             % Count up time
        hmm(T1,N,Next,Y).      % Go next (recursion)
str_length(10).                % String length is 10
set_params :- set_sw(init, [0.9,0.1]), set_sw(tr(s0), [0.2,0.8]), set_sw(tr(s1),
[0.8,0.2]), set_sw(out(s0),[0.5,0.5]), set_sw(out(s1),[0.6,0.4]).
```

The most appealing feature of PRISM is that it allows the users to use random switches to make probabilistic choices. A random switch has a name, a space of possible outcomes, and a probability distribution. In the program above, msw(init,S) probabilistically determines the initial state from which to start by tossing a coin. The predicate set_sw(init, [0.9,0.1]), states that the probability of starting from state s0 is 0.9 and from s1 is 0.1. The predicate learn in PRISM is used to learn from examples (a set of strings) the parameters (probabilities of init, out and tr) so that the ML (Maximum-Likelihood) is reached. For example, the learned parameters from a set of examples can be: switch init: s0 (0.6570), s1 (0.3429); switch out(s0): a (0.3257), b (0.6742); switch out(s1): a (0.7048), b (0.2951); switch tr(s0): s0 (0.2844), s1 (0.7155); switch tr(s1): s0 (0.5703), s1 (0.4296). After learning these ML parameters, we can calculate the probability of a certain observation using the predicate prob: prob(hmm([a,a,a,a,a,b,b,b,b,b]) = 0.000117528. This way, we are able to define a probability distribution over the strings that we observe. Therefore from the basic distribution we have induced a probability distribution over the observations.

4 PRISM Modeling of Structural Signatures

What we need to model in PRISM the structural signatures of protein domains is a set of rules and a set of facts with a probability distribution over them. The set of 59 rules learned in [5] can be used without any changes. We have to define the random switches and learn for them a probability distribution which models the uncertainty about the protein domains for their classification. In the predicate adjacent(D, A, B, Pos, TypA, TypB) that is used as background

knowledge, we define a random switch that probabilistically assigns the values e or h (strand or helix) to TypA. What we have modeled in this way is a probability distribution over secondary structures that are of type e or h. Therefore after learning the parameters for this random switch we have two values that represent the probability that a secondary structure is of type e or h in the dataset of training. Another random switch that we define is that based on the length of the secondary structure. This represents the probability that a certain secondary structure has a certain length. The possible values of the length of the secondary structure define the space of possible outcomes for this second random switch.

We used as training data the dataset used in [5] and performed the experiments in PRISM version 1.10 through a 5-fold cross-validation on 381 examples. After learning the parameters for the two random switches we calculated the probability for each of the observations. Now we explain how these probabilities can be used to solve the problem of multiple predictions. In multi-relational data mining, cases of multi-class classifications are treated by assigning a test example with multiple predictions to the fold which covers the maximum number of examples. For example, if for the fold "DNA-binding 3-helical bundle" have been learned 4 rules which together cover (explain) 74 training examples and for the fold "Periplasmic binding protein-like II" have been learned 3 rules which together cover 30 examples, then the protein domain "d1hslb_" is assigned to the fold which covers more examples. In this case the prediction is wrong since the protein domain "d1hslb_" in reality belongs to the fold "Periplasmic binding protein-like II". If the number of the examples covered by the folds is equal, the example is assigned randomly. This has proven to be not an optimal solution and generally has produced low predictive accuracy in multi-class classification problems. In order to model the uncertainty of which fold to choose in case of multiple predictions we use the probabilities of the observations that we compute in PRISM. We sum the probabilities of the observations (training examples) that belong to the same fold. In this way we rank the folds with a probability instead of the number of the examples covered and in case of a multiple prediction for an example of testing we assign the example to the fold with a greater probability.

We have performed two experiments. In the first we used as a classification criterion the number of covered examples for each fold, i.e. examples with multiple predictions (covered by rules belonging to different folds) were assigned to the fold with the greatest number of covered examples. While in the second experiment we used the probability of each fold to solve multiple predictions, i.e. an example with multiple predictions was assigned to the fold with the highest probability. Table 1 contains the results of these experiments. Each column corresponds to one of the datasets in the 5-fold cross-validation and contains for each row the test results, i.e. number of correct classified examples towards the number of all the examples of testing. The number of examples with multiple predictions is about 63 % of the total number of examples.

As we can see from the table, in the Experiment 2 where we used the system PRISM and modeled the uncertainty with the probabilities of the observations, we obtained a predictive accuracy of 65,35 % towards 49,6 % of the Experiment

Table 1. Results of the 5-fold cross-validation

	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Overall
Exp. 1	37/76	34/76	45/76	40/76	33/77	189/381
Exp. 2	54/76	50/76	51/76	52/76	42/77	249/381

1 where we do not use probabilities. The difference in predictive accuracy is significant at the 0,005 level in a paired t-test. Analyzing the experiments' results, we observed that the significant difference in accuracy among the two experiments is due to the fact that for many examples with multiple predictions, their classification in the fold with the greatest probability was correct. This shows that fold's probability provides a more principled and robust method for handling the uncertainty of multiple predictions against the fold's number of covered examples.

The experiments validate our approach of handling the uncertainty of multiple predictions through fold probabilities. Using PRISM it was possible to learn fold probabilities from observations (training examples) and therefore better identify the most probable fold for a test example with multiple predictions. What we have learned from this application is that hybrid symbolic-statistical approaches can solve problems for which single symbolic approaches fail, such as problems where uncertainty must be dealt with.

5 Conclusions and Future Work

In this paper we have applied the symbolic-statistical framework PRISM to a multi-class protein fold recognition problem. We have exploited the ability of PRISM to represent proteins' three-dimensional structures through logic programs and to model the uncertainty about observations through learning switch probabilities. In dealing with a multi-class prediction problem we have used probability of protein folds to correctly classify test examples with multiple predictions. We have shown that the proposed method outperforms the symbolic-only approach in terms of predictive accuracy. This is to the best of our knowledge the first application of the framework PRISM to a problem of protein folding and multi-class prediction.

As future work we intend to apply PRISM to other datasets for protein fold recognition problems. We believe that PRISM, having the expressive power of a logic-based language and the ability to deal with uncertainty in a robust manner through EM based learning algorithms, provides a valid framework for dealing with structural domains with intrinsic uncertainty. Moreover, we intend to evaluate the performance of our approach towards other methods that have been applied to multi-class classification problems such as support vector machines and neural networks [9] which are among the state-of-the-art discriminative methods that have produced accurate results for the multi-class protein fold recognition problem.

References

1. Moult, J.: Rigorous Performance Evaluation in Protein Structure Modeling and Implications for Computational Biology. Phil. Trans. R. Soc. B 361, 453–458 (2006)
2. Baldi, P., Brunak, S.: Bioinformatics: The Machine Learning Approach, 2nd edn. MIT Press, Cambridge (2001)
3. Muggleton, S.H., De Raedt, L.: Inductive logic programming: Theory and methods. Journal of Logic Programming 19(20), 629–679 (1994)
4. Page, D., Craven, M.: Biological Applications of Multi-Relational Data Mining. Appears In: SIGKDD Explorations, special issue on Multi-Relational Data Mining (2003)
5. Turcotte, M., Muggleton, S.H., Sternberg, M.J.E.: Automated discovery of structural signatures of protein fold and function. Journal of Molecular Biology 306, 591–605 (2001)
6. Sato, T., Kameya, Y.: PRISM: A symbolic-statistical modeling language. In: Proceedings of the 15th International Joint Conference on Artificial Intelligence, pp. 1330–1335 (1997)
7. LoConte, L., Ailey, B., Hubbard, T.J.P., Brenner, S.E., Murzin, A.G., Chothia, C.: SCOP: a structural classification of proteins database. Nucl. Acids Res. 28, 257–259 (2000)
8. Sato, T., Kameya, Y.: Parameter learning of logic programs for symbolic-statistical modeling. Journal of Artificial Intelligence Research 15, 391–454 (2001)
9. Ding, C.H., Dubchak, I.: Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics 17(4), 349–358 (2001)