.

České vysoké učení technické v Praze
fakulta Elektrotechnická

Czech Technical University in Prague
faculty of Electrical Engineering

Ing. Filip Železný, Ph.D.

Genomické Aplikace Relačního Strojového Učení

Genomic Applications of Relational Machine Learning

Summary

This talk reviews selected applications of relational machine learning in genomic data analysis. The machine learning task relevant to the applications is that of learning classification functions from examples. In conventional machine learning, examples are described through attribute tuples. In domains where the relational structure of examples is important, attribute-value descriptions are inappropriate. To this end, relational machine learning deals with learning from structural example descriptions. In its most extensively elaborated subfield, inductive logic programming, first-order predicate logic is used as the representation language. Examples are usually expressed as first-order clauses and the learned classifier is a clausal theory (set of clauses). A further clausal theory, describing relevant background knowledge, may be supplied to and exploited by the learner. In the first application of relational machine learning, we aim at learning classifiers predicting gene groups that will be differentially expressed over given phenotypes. These classifiers are learned from gene expression measurement data and the gene groups are characterized by the learner in terms of relational background knowledge pertaining to the gene ontology and to known gene-gene interactions. In the second application, we want to learn classifiers predicting whether or not a given protein is able to bind DNA. The classifiers are learned from structural (3D) descriptions of proteins. Here we combine the first-order logic based approach with the conventional attribute-based approach, obtaining predictive accuracies that improve upon the state of the art. We conclude by briefly reviewing a few other applications of relational machine learning in genomics, including studies by other authors.

Souhrn

Přednáška představuje vybrané aplikace relačního strojového učení v analýze genomických dat. Úloha strojového učení relevantní k těmto aplikacím se týká učení klasifikační funkce z příkladů. V tradičním strojovém učení jsou příklady popsány n-ticemi příznaků. V oblastech, kde je pro klasifikaci důležitá relační struktura příkladů, není příznakový popis vhodný. Za tímto účelem pracuje relační strojové učení se strukturními popisy příkladů. V jeho nejšířeji prozkoumané podoblasti, induktivním logickém programování, je jako reprezentační jazyk využita predikátová logika prvého řádu. Příklady jsou obvykle vyjádřeny jako prvořádové logické klauzule a naučený klasifikátor je klauzální teorií (tedy množinou klauzulí). Učící se algoritmus může také využít další klauzální teorii popisující relevantní apriorní (předem danou) znalost. V první aplikaci relačního strojového učení je naším cílem naučit se klasifikátory předpovídající, které genové skupiny budou rozdílně exprimovány mezi zadanými fenotypy. Tyto klasifikátory jsou natrénovány z měřených dat genové exprese a genové skupiny jsou popsány algoritmem v termínech relační apriorní znalosti vyplývající z genové ontologie a známých vzájemných interakcí genů. V druhé aplikaci chceme natrénovat klasifikátory predikující, zda je zadaná bílkovina schopna vázat DNA. Klasifikátory jsou natrénovány na strukturních (3D) popisech bílkovin. V tomto případě kombinujeme metody založené na predikátové logice s konvenčním příznakovým učením, čímž dosahujeme prediktivních přesností překonávajících dosud používané klasifikátory. Přednášku uzavíráme krátkým přehledem několika dalších aplikací relačního strojového učení v genomice, včetně studií jiných autorů.

Klíčová slova

Umělá inteligence, relační strojové učení, induktivní logické programování, analýza dat, prediktivní klasifikace, bioinformatika, molekulární genomika, strukturní proteomika, genová exprese, transkripční faktory

Keywords

Artificial intelligence, relational machine learning, inductive logic programing, data analysis, predictive classification, bioinformatics, molecular genomics, structural proteomics, gene expression, transcription factors

Contents

1 Background

1.1 Machine Learning

To introduce relational machine learning, we will set off from the conventional statistical attribute-value machine learning setting [7] since the latter is a framework familiar to many. Within this framework, we specifically focus on *supervised learning* algorithms that are used to learn a classification function from a set of classified examples. More precisely, we consider a finite set of n random variables $a_1, a_2 \ldots a_n$ called *attributes*. An attribute-value description of a learning example is a particular assignment

$$\mathbf{x} \in X = dom(a_1) \times dom(a_2) \times \ldots \times dom(a_n)$$

of values to these attributes where, for simplicity, we consider all the domains discrete (countable). Further we consider a random *class* variable y taking values from a finite set Y, and a joint probability distribution $P_{X,Y}$. A *training set* is a finite multiset drawn i.i.d. from $P_{X,Y}$ and its elements are called *training examples*. A learning algorithm receives a training set and outputs a representation of a function $f : X \to Y$. In a widely adopted Bayesian view, f would ideally minimize the *risk*[1]

$$R(f) = \sum_{\mathbf{x} \in X} \sum_{y \in Y} L\left(f(\mathbf{x}), y\right) P_{X,Y}(\mathbf{x}, y)$$

involving an apriori defined *loss function* $L(.,.)$ quantifying the importance of all possible misclassifications, i.e. $L(y, y) = 0$ for all $y \in Y$. If $L(y, y') = 1$ whenever $y \neq y'$, then $R(f)$ is called the *classification error* of f; in what follows we will always assume this to be the case. The risk $R(f)$ usually cannot be computed exactly since $P_{X,Y}$ is typically not known. We therefore work with empirical estimates $E(f, S)$ of $R(f)$ computed on an i.i.d. sample S of m elements from $P_{X,Y}$

$$E(f, S) = \frac{1}{m} \sum_{(\mathbf{x},y) \in S} L\left(f(\mathbf{x}), y\right)$$

If S is the training set used for learning f, then $E(f, S)$ is called the *training error*. If S is a sample independent of the training set, then S is called a *testing set* and $E(f, S)$ is called the *testing error*, which is an unbiased estimate of $R(f)$.

Given a training set T, a learning algorithm seeks a suitable classifier f from among a predefined class of functions $\mathcal{F}$. $\mathcal{F}$ is said to define a *hard bias*

[1] In all sums over possibly infinite domains, we silently assume these converge.

of the algorithm; depending on the respective domains of the n attributes, it may e.g. be the class of linear discrimination functions in R^n, functions representable as decision trees, propositional-logic rules, neural networks, etc. In choosing a function from $\mathcal{F}$, the algorithm may proceed simply by minimizing the training error $E(f, T)$, however, this approach would typically result in *overfitting*. So termed is the situation where $R(f)$ is large despite $E(f, T)$ being small. Overfitting is the more eminent the larger ('more flexible') the hard bias $\mathcal{F}$ is.[2] Often therefore, learners minimize $E(f, T) + \lambda\rho(f)$ where $\lambda \in R$ and $\rho(f)$ is a *regularization term*, defining the *soft bias*. One often chooses $\rho(f)$ such that it assigns lower values to *simple classifiers* (usually measured by their description length) and thus penalizes the complex ones. This way, flexibility of $\mathcal{F}$ is reduced by an extent parameterized by λ. The art of designing a machine learning experiment rests in the good choice of the hard and soft biases, depending on the background of the data.

1.2 Relational Machine Learning

Conventional machine learning is suitable when there is a natural way to describe data through the values of attributes. In some important domains, this is not the case. Consider e.g. that data are organic chemical molecules classified as carcinogenic or non-carcinogenic. Adhering to the attribute-tuple representation, we could easily represent properties such as charge, weight, number of carbon elements and so forth, for each molecule. However, carcinogenicity is mostly determined by the structure of the molecules rather than the mentioned properties. We would thus like to be able to learn classifiers predicting through relational reasoning, such as *a molecule is carcinogenic if it contains a benzene ring that in turn contains an element connected to an oxygen atom through a double bond*. In principle, we could manually formulate relational conditions such as the above, check their truth values for each data instance, and present them as Boolean-domain attributes to a conventional learner. However, the number of possible statements of the exemplified kind is combinatorially vast and usually we have no clue to judge which ones are relevant for classification. Therefore, we want the learning algorithm itself to be able to construct such logical assertions as part of learning.

Relational machine learning algorithms aim at solving this problem. This talk focuses on the family of relational learners based on the framework of *inductive logic programming* (ILP). As the name suggests, this framework uses formal logic for data and classifier representation as well as for inference. Though in scope of current vital research, this talk does not cover recent ex-

[2] Entire branches of machine learning theory elaborate this simple statement into precision, see e.g. [22, 23].

tensions of ILP dealing with probabilistic representation of uncertainty [3] or approaches to relational learning based on grounds different from logic [5].

Data, from which ILP systems learn, are relational structures. Specifically, in the *normal setting* of ILP, they are first-order predicate clauses. Using the clausal representation, a fragment of a training example in the chemical domain would e.g. be

$$\begin{aligned}\mathsf{Carc}(\mathsf{M1}) \leftarrow &\mathsf{Atom}(\mathsf{M1},\mathsf{A1}) \wedge \mathsf{Type}(\mathsf{A1},\mathsf{Carbon}) \wedge \mathsf{Atom}(\mathsf{M1},\mathsf{A2}) \\ &\mathsf{Type}(\mathsf{A2},\mathsf{Oxygen}) \wedge \mathsf{Bond}(\mathsf{Double},\mathsf{A1},\mathsf{A2}) \wedge \ldots\end{aligned}$$

that is to be interpreted as *Molecule M1 is carcinogenic as a result of containing a carbon atom and an oxygen atom connected by a double bond* ... (the full example would contain the description of the entire molecular structure). The training examples are partitioned into the set of *positive examples* E^+ (here, carcinogenic molecules) and *negative examples* E^- (non-carcinogenic). An ILP algorithm seeks a *first-order clausal theory* (set of clauses, also called a *hypothesis* in the learning context) H that explains the training data. That is to say, $H \models e$ for as many as possible $e \in E^+$ but for as few as possible $e \in E^-$. Here, the sign $\models$ denotes *logical entailment*. For example, the clause shown above is entailed by the single-clause theory

$$\forall x, y : \mathsf{Carc}(x) \leftarrow \mathsf{Atom}(x, y) \wedge \mathsf{Type}(y, \mathsf{Carbon})$$

stipulating that any molecule is carcinogenic if it contains a carbon atom. Obviously, such an overly general hypothesis would be eliminated since it would likely entail many negative examples as well. In what follows, we will omit the universal quantification in the clauses and always assume all variables to be universally quantified in the clause.

The entailment relation $\models$ is in general undecidable even if the theory H is a single clause. This comes at little surprise given the high expressiveness of the first-order predicate logic. Therefore, $\models$ is usually approximated by the decidable relation $\preceq_\theta$ called θ-subsumption, that is only defined between single clauses. If relying on $\preceq_\theta$, multi-clause theories must be learned iteratively (adding a clause at a time), e.g. using the *covering strategy* well known from rule learning [13]. The relation $\preceq_\theta$ is verified by syntactical inspection of the two clauses and this verification is known to be NP-complete.

A more general formulation of the normal ILP setting additionally involves a clausal theory B acting as an input to the learning task. Through B, one can express *background knowledge* relevant to classification. For instance, by the following background knowledge clause

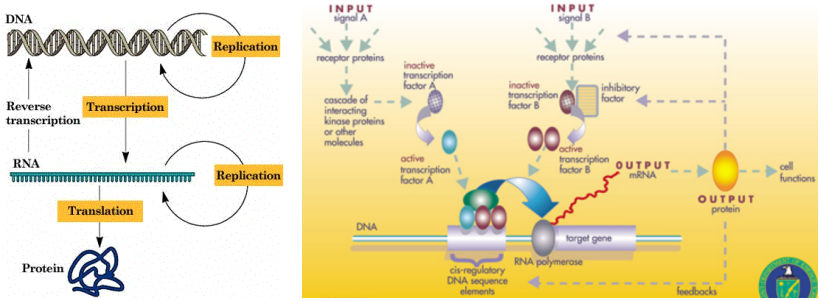

Fig. 1: Left: The central dogma of molecular biology describing the flow of information from DNA to proteins. In this talk we are not interested in the processes of reverse transcription or RNA replication. **Right:** An simple scheme of a gene expression regulatory network. (From Wikimedia Commons)

$$\mathsf{BRing}(x_1, x_2, \ldots, x_6) \leftarrow \mathsf{Type}(x_1, \mathsf{Carbon}) \wedge \mathsf{Bond}(\mathsf{Single}, x_1, x_2) \\ \wedge \mathsf{Type}(x_2, \mathsf{Carbon}) \wedge \mathsf{Bond}(\mathsf{Double}, x_2, x_3) \ldots$$

we would define that six carbon atoms in a particular bonding constitute a *benzene ring*. The learner can then exploit the predicate BRing in forming H, e.g. by plugging it as a literal in a considered clause. To check entailment, the learner considers $H \cup B \models e$ instead of just $H \models e$. If $\preceq_\theta$ is used instead of $\models$, computational tricks must be employed to approximate the multi-clause theory $H \cup B$ by a single clause.

Relating ILP to the conventional machine learning framework we explained initially, we see that ILP considers binary classification, i.e. $Y = \{0, 1\}$. This comes without loss of generality. Instead of the attribute-value tuples $\mathbf{x}$ representing data, ILP assumes clauses e. The learned hypothesis H represents a classifier f_H such that $f_H(e) = 1$ iff $H \models e$. The hard bias of the learner is given by B and a specification of the particular language for expressing H. The soft bias usually penalizes syntactically complex theories H. Given these bridges, all the statistical rationale explained for conventional machine learning translates also to ILP.

1.3 Molecular Genomics

Here we briefly introduce the aspects of molecular genomics[3] relevant to the relational learning applications addressed subsequently. Hereditary infor-

[3]Molecular genomics is the intersection of genomics and molecular biology. Other branches of genomics study e.g. the Mendelian inheritance principles.

mation prescribing the construction of an organism is stored in a *deoxyribonucleic acid* (DNA). Eucaryotes store one copy of the same DNA template in the nucleus[4] of each of its cells. From the information-theoretic viewpoint, the DNA is a sequence of symbols drawn from a 4-symbol alphabet. In humans, it is about 3.10^9 symbols long. The symbols are called *bases* and correspond to the respective molecules guanin, cytosin, thymin and uracil. Most of the time, the DNA in fact consists of two parallel sequences (*strands*, follow left panel in Fig. 1) of the said length, which are however *complementary* in that a symbol at a position of one strand uniquely determines the symbol at the same position of the other strand. This parallelism serves DNA-replication purposes in processes such as cell multiplication.

The DNA contains distinguishable regions known as *genes* (about 20 thousand genes in human DNA), which are subject to the process of *gene expression* conducted by a cellular machinery whose details are out of the scope of this talk. As an effect of this process, a gene is first transcribed into a *ribonucleic acid* (RNA), which also is a 4-symbol alphabet just as the DNA and serves for passing the information content from the cellular nucleus to the cytoplasm.[5] Here, the RNA is *translated* into a *protein*. To describe a protein, we consider two perspectives. In a *primary structure* perspective, a protein is a sequence of symbols drawn from an alphabet of about 20 symbols; these correspond to various *amino acids*. To constitute a protein, amino acids lose a water molecule and as such are called *residues*. From an expressed gene, a protein primary structure is formed by following the gene's sequence; each three consecutive DNA bases (commonly called a *codon*) determine which residue to attach to the protein under construction. Since $3^4 = 81 > 20$, multiple codons may map to a single amino acid. Codons mapping to the same amino acid usually have similar base sequence and this contributes to resistance against translation errors. From a *higher-order structure* perspective,[6] a protein folds into a spatial form uniquely defined by its primary structure and determining the protein's physiological function such enzymatic activity, cellular scaffolding, or cell to cell signaling. In effect, the genes expressed into proteins in a tissue specify the structure and function of the tissue, modulo external influences. The principles explained so far are informally referred to as the central dogma of molecular biology.

Cell functioning depends on which genes get expressed in what situations. Proteins known as *transcription factors* (TF) regulate the expression of genes. A TF is able to physically *bind* to a region on the DNA, called the

[4] Small pieces of DNA are also located in mitochondria. We ignore them here.

[5] area bounded by the cellular membrane, outside the nucleus and other organelles.

[6] Secondary, ternary, and quaternary structures are distinguished. Here we treat them collectively.

promoter region (PR) of a gene, located in the vicinity of that gene. This binding is *specific* in that each TF binds to the PR's of certain *target* genes only, although multiple TF's can bind to a single gene's PR. Whether or not binding occurs is given by the spatial conformation of the TF and the sequential content of the PR. A TF may *catalyze* the expression of a gene by 'dragging-and-dropping' it to the transcription machinery, or *inhibit* it by merely binding to the gene's PR and thus blocking the access of any catalyzing TF. Through TF's, a cell is able to react to external stimuli and produce situation-specific proteins. This is because TF's floating in the cytoplasm are, by their default structure, usually inactive (cannot bind to a PR) and only activate upon interaction with a signal carrier[7] usually coming through the membrane from outside the cell. Since TF's are proteins, they are also regulated by other TF's or even by themselves. This gives rise to an extremely complex network of regulatory interactions including omnipresent feedback loops (see the right panel of Fig. 1). Consequently, the concentrations of proteins in a cell can be seen as a state-space vector in a mass-dimensional non-linear dynamic system. Steady states of the cell have been shown to correspond to attractors in this system, and their transitions are a result of external perturbations combined with intrinsic stochastic fluctuations [12].

2 Applications

2.1 Learning Descriptions of Gene Groups

Here we employ relational machine learning to characterize which families of genes are expressed in given situations. This application was enabled by recent progress in biotechnology which made it possible to measure expression of genes on a massive scale. Traditional techniques for measuring gene expression are laborious and provide estimates relating to only a few apriori selected genes. In the 1990's, however, *expression chips* (also called DNA chips, gene chips, microarrays) emerged, manufactured using technologies derived from computer-chip production (see Fig. 2, left panel). These can measure the expression of thousands of genes simultaneously, under different conditions.

Expression measurements are performed on a sample of RNA extracted from the investigated tissue. The amount of RNA corresponding to a given gene is considered a surrogate measure for the amount of protein made from that gene (recall Fig. 1, left panel). The RNA sample is colored and spread on the surface of a DNA chip that is an array of DNA probes (follow Fig. 2,

[7] Physically, activation corresponds e.g. to adding a phosphorylation or binding a ligand molecule.

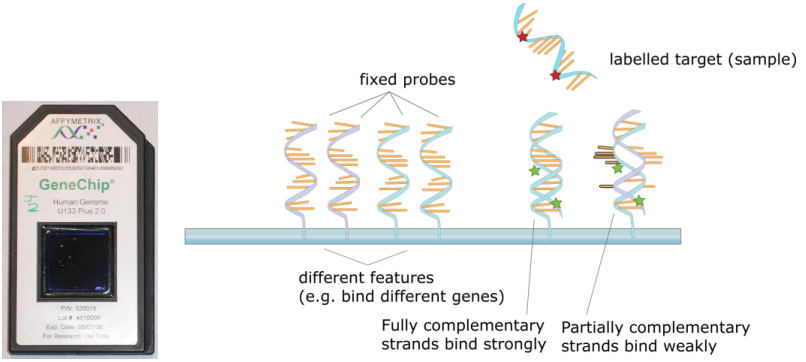

Fig. 2: Left: An expression chip produced by Affymetrix. **Right:** The principle of its operation. (From Wikimedia Commons)

right panel). A DNA probe is a string of about 20 bases, complementary to a substring of a gene of interest. Currently, microarrays may encompass probes for up to tens of thousands of genes, i.e. entire genomes. If the applied sample contains RNA of a particular gene, it will *hybridize* (attach) to the corresponding probe due to the complementarity principle (recall Section 1.3). Due to the RNA coloring, these spots are then easily identified optically (see Fig. 3, left panel). Microarrays in fact contain multiple probes for each gene interrogated; the more RNA is present for a gene, the more of probes for that gene are hybridized. In effect, the measured color intensity of the gene's probe set is a growing function of the level of the gene's expression.

The output of a series of microarray experiments is a matrix with genes spread along one dimension and RNA samples (relating to different conditions) along the other dimension. Often, only few different conditions are considered (e.g. cancerous vs. control) and multiple samples are taken in each of the conditions. As a result, collected gene expression measurements acquire the form of classified attribute-value data wherein genes are attributes and the respective conditions represent classes. Therefore machine learning can be applied to induce classifiers predicting the sample class from expressions of genes in that sample. The utility of such classifiers, at least for diagnostic purposes, is unquestionable [6]. Unfortunately, the cost of a single microarray experiment is high ($100's). Thus in most real-life lab experiments the number of attributes is much larger than the number of samples (usually tens of thousands of genes against units or tens of samples). This causes an extremely high risk of overfitting and often prevents the algorithm from learning a reliable classifier.

Due to the described obstacle to machine learning application, the most common way of analyzing gene expression data is to rely on standard statistical techniques to identify a set of 'suspicious' genes. Suspicious genes are usually those exhibiting largely different expression across the different classes. The problem of this approach is that long lists of genes are produced, usually with no apparent mutual relationships. These are then very difficult to interpret in terms of biological processes and link them to any concise phenomena underlying the differential expression.

The two problems described above are so characteristic of expression data analysis they have earned their aliases in the folklore of the field (the *large n / small m syndrom*, and the *gene list syndrom*). In [21], we presented a relational machine learning application that contributes to solving both of them simultaneously. In particular, we aimed at learning compact characterizations of differentially expressed gene sets in terms of background knowledge. Viewed as a classification task, we wanted to learn classifiers predicting whether or not a gene will be differentially expressed between given conditions, according to background knowledge about that gene.

Note that in this formulation of the learning task, genes correspond to learning examples rather than to attributes. While their sheer number was a problem in the above described conventional application of machine learning, in our approach it is a benefit. A transition from attribute-value learning to relational learning techniques is however required here, since available genomic background knowledge is relational.

In particular, we used two sources of background knowledge. The first is the Gene Ontology (GO, www.geneontology.org), which provides a controlled vocabulary for the description of cellular components, molecular functions, and biological processes. Terms in this vocabulary are linked through two kinds of binary relations ('part of' and 'is a', see Fig. 3, right panel). A GO-based annotation of a gene is a subset of terms of the GO. The second source of relational background knowledge we used is the database of reported gene-gene interactions sourced from the US National Center of Biotechnology Information.

The input list of gene sets was first extracted using gene expression data from previous research. In particular we considered the studies [6, 16, 15]. For each of the three gene expression datasets, we extracted both a set of positive examples (gene differentially expressed) and negative examples (other genes) according to a statistical test. See [21] for details on the preliminary gene selection.

Learning of relational descriptions of gene groups was then formulated as a task that technically slightly differs from the normal ILP setting as introduced in 1.2 (see [21] for details) but the difference is not important in this talk.

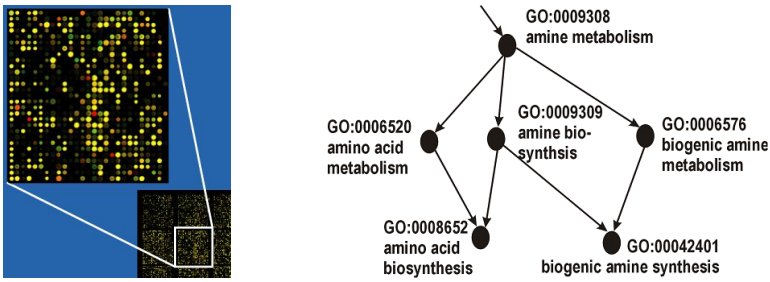

Fig. 3: **Left:** A scan of a microarray with hybridized probes. **Right:** The Gene Ontology provides a structured vocabulary to describe genes.

As a result, we obtained characterizations such as

$$\begin{aligned}\mathsf{Diff}(x) \leftarrow &\mathsf{Interaction}(x, y) \wedge \mathsf{Process}(y, \mathsf{Phosphorylation}) \wedge \\ &\mathsf{Interaction}(x, z) \wedge \mathsf{Process}(z, \mathsf{Negative\ regulation\ of\ apoptosis}) \\ &\wedge \mathsf{Component}(z, \mathsf{intracellular\ membrane\text{-}bound\ organelle})\end{aligned}$$

describing genes differentially expressed between the central nervous system cancer class on one hand and other classes on the other hand. The reliability of the learned characterizations was tested through cross-validation and the results demonstrated an acceptable decay from the training to the testing set in terms of classification error. See [21] for performance details as well as for biological comments on the learned characterizations.

2.2 Predicting Protein-DNA Interactions

Whereas the study above did yield some concise characterizations of expressed gene groups, it admittedly had an air of a fishing expedition due to the high generality and—to some extent—vagueness of the information sources used as background knowledge. In our latest experiments [18, 10], we wanted to study the expression phenomenon in a more constrained and sharply defined manner. Recall from Section 1.3 that gene expression needs the assistance of transcription factors, which are proteins able to physically bind the DNA (see Fig. 4). We wanted to be able to classify whether a protein is able to bind DNA given the spatial structure of the protein. To this end, we decided to learn classifiers using data about proteins previously reported as DNA binding.

Solving this problem is important for several reasons. In particular, current art is far from knowing all proteins acting as DNA binders and their identi-

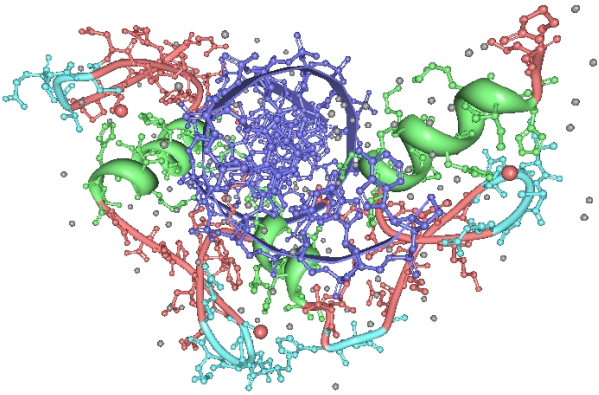

Fig. 4: A DNA (blue in the middle) bound to a protein (other colors). Balls represent atoms. Spatial motifs within the protein are distinguished by color (green – helices, red – sheets, cyan – turns).

fication may help in hypothesizing about yet-unknown expression regulatory networks. Of equal importance, DNA binding proteins are currently in scope of excited research in the area of gene therapy where they represent an instrument for DNA editing [8]. One of the challenges in this stream of research is to build a library of DNA binders.

We were not the first to try to predict DNA binding proteins. Previous approaches include learning with neural networks [17, 1], support vector machines [2], or logistic regression [19]. What they had in common is that classifiers were learned using the attribute-value data representation. The involved attributes of proteins had a coarse-grained nature, relating to protein's properties such as overall electric charge, amino acid composition distribution, the asymmetry of the spatial distribution of specific residues and the dipole moment of the protein. The study [19] reported a ranking of such features by their power to predict DNA binding.

In contrast to these studies, we wanted to predict DNA binding directly from the three-dimensional conformation of the proteins. For comparative purposes we decided to work with the same set of protein examples as [19]. We downloaded the atom-level structural descriptions of 54 DNA binding proteins and of 110 non-DNA-binding proteins from the Protein Data Bank (www.pdb.org) to act as positive (negative, respectively) examples. For computational feasibility we first recalculated the descriptions from the atom level to the residue level. Eventually, each protein description consisted of facts pertaining to two predicates that respectively described the presence of a resi-

due, and the spatial distance between two residues in Angstroms. For example, this beginning of a clause

$$\mathsf{Binds}(\mathsf{P1}) \leftarrow \mathsf{Res}(\mathsf{P1}, \mathsf{R1}, \mathsf{His}) \wedge \mathsf{Res}(\mathsf{P1}, \mathsf{R2}, \mathsf{Arg}) \wedge \mathsf{Dist}(\mathsf{R1}, \mathsf{R2}, 10.0) \wedge \ldots$$

asserts that protein P1 contains the amino acids Histidine and Arginine that are 10 Angstroms apart. A complete description of a protein addresses all involved residues, and their all pairwise spatial distances that do not exceed 40 Angstroms as computed from coordinates of the residues' *alpha carbons* (distinguished atoms representing residue centroids). The full description of a single protein contained up to tens of thousands of literals. A possible theory is e.g. one assuming three specific residues and two pairwise distances

$$\begin{aligned}\mathsf{Binds}(x) \leftarrow & \mathsf{Res}(x, y, \mathsf{Arg}) \wedge \mathsf{Res}(x, z, \mathsf{Gln}) \wedge \mathsf{Dist}(y, z, 10.0) \\ & \wedge \mathsf{Res}(x, w, \mathsf{Leu}), \mathsf{dist}(y, w, 10.0) \end{aligned} \quad (1)$$

Through experimentation, we determined that to obtain good predictive accuracies, we need to slightly deviate from the standard ILP framework. In particular, it was not appropriate in the current domain to assign only truth values to each pair of a theory and example, as standard in ILP. For class discrimination, it turned out important to *count the number of occurrences* of the spatial pattern defined by the theory in the exemplary protein. Technically, we thus proceeded as follows. We constructed a large number of single-clause theories. Then we considered each of them to represent an attribute, calling it a *spatial feature*. For each example, the occurence count for a given spatial feature was assigned to it as the value for that example. Thus we derived an attribute-value description of each example, based on its spatial structure. We also explored the option where coarse-grained attributes suggested in previous research [19] were also computed and added alongside the spatial features. In the described protocol, we employed our recently published algorithm [9] since it can scale to rather large structures corresponding to proteins, which would be prohibitively large for mainstream inductive logic programming algorithms. This algorithm exhaustively constructs a set of relational features which are not *redundant*, comply with a user-defined language bias and have frequency higher than a given threshold. As a result, we maintained about 1500 spatial features. The final attribute-value representations were then passed to seven different state-of-the-art machine learning algorithms.

Analyzing predictive accuracy results over 10 folds of cross-validation and the 7 employed learning algorithms, the design where our spatial features were combined with existing coarse-grained features significantly improved

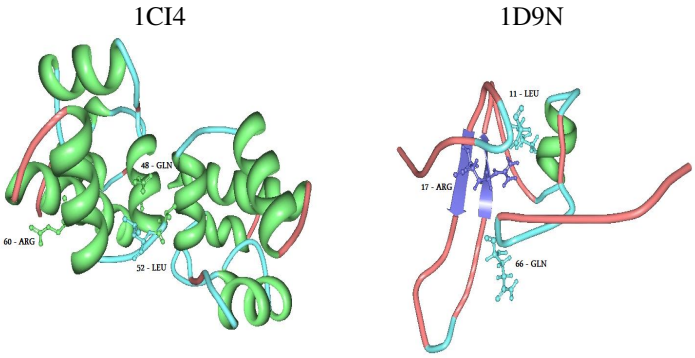

Fig. 5: Example proteins containing the most discriminative spatial feature. Residues assumed by the feature are indicated. Their pairwise distances (not shown) are conserved over the positive examples of proteins.

on state-of-the-art accuracies. When the two sets of attributes were isolated from each other, our spatial count-based approach outperformed the state-of-the-art approach based on coarse-grained attributes.

Besides outperforming the state-of-the-art approach, another advantage of our method is that its results have a visual representation that can be interpreted. To show an example, we consider the most discriminative spatial feature according to the χ^2 criterion. This feature was in fact already presented as an example in Eq. 1 and is graphically shown in two different proteins in Fig. 5.

2.3 Other Applications

Lastly, we briefly review a few more significant applications of relational machine learning, particularly inductive logic programming, in genomics and proteomics. We do so chronologically.

In the pioneering work [14], protein secondary structure was predicted from the primary structure and background knowledge. Recall from Section 1.3 that primary structure (i.e., the sequence of specific residues) determines the secondary structure (i.e., the 3D motifs of the protein). Remarkably, this deterministic mapping has never (even to date) been exactly deciphered. In the early 90's, the ILP system Golem learned classifiers discriminating helix motifs from sheet motifs (see Fig. 4) from the sequence data and background knowledge pertaining to chemical properties of amino acids such as polarity or hydrophobicity. The accuracy of the method was, at the time, unmatched by other prediction methods.

More recent efforts concentrated on the reconstruction of *metabolic networks* from data. Metabolic networks are principally similar to the expression regulation networks we have reviewed but encompass a wider set of entities and relations that act in the processing of energy. Besides enzyme proteins, the entities include substrates, metabolites and ligand molecules and the relations include e.g. signaling instruments such as protein phosphorylation. In [20], missing parts of metabolic networks were completed through ILP-based learning.

The study [4] presented an application of ILP, where the goal was to predict the expression regulation of a gene from information relating to the promoter site, state of transcription factors and from additional information. This application thus had a goal similar to that in Section 2.2 but it did not exploit protein 3D structural information.

In our most recent experiments [11], we contributed to the task of estimating covariance matrices of random variables corresponding to expressions of genes. This task is routine in systems biology. Due to the scarcity of expression samples, such estimated matrices are unstable and the undesirable estimation variance is compensated through regularization. The standard approach is to bring the initially estimated matrix closer (w.r.t. a suitable matrix metric) to the diagonal unit matrix. In [11] we have proposed a knowledge-based way to matrix regularization using biological rules learned by ILP.

References

[1] S. Ahmad and A. Sarai. Moment-based prediction of DNA-binding proteins. *Journal of Molecular Biology*, 341(1):65–71, 2004.

[2] N. Bhardwaj, R. E. Langlois, G. Zhao, and H. Lu. Kernel-based machine learning protocol for predicting DNA-binding proteins. *Nucleic Acids Research*, 33(20):6486–93, 2005.

[3] L. De Raedt. *Logical and Relational Learning*. Springer, 2008.

[4] S. Frohler and S. Kramer. Inductive logic programming for gene regulation prediction. *Machine Learning*, 70(2-3):225–40, 2006.

[5] L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning*. The MIT Press, 2007.

[6] T.R. Golub et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–7, 1999.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[8] A.S. Hirsh and J.K. Joung. Designer zinc finger proteins for gene therapy: progress and challenges. *Gene Therapy and Regulation*, 2:191–206, 2004.

[9] O. Kuzelka and F. Zelezny. Block-wise construction of acyclic relational features with monotone irreducibility and relevancy properties. In *ICML 2009: The 26th International Conference on Machine Learning*, 2009.

[10] O. Kuzelka and F. Zelezny. Prediction of dna-binding proteins from structural features. In *MLSB 2010: 4th International Workshop on Machine Learning in Systems Biology*, 2010.

[11] O. Kuzelka and F. Zelezny. Shrinking covariance matrices using biological background knowledge. In *MLSB 2010: 4th International Workshop on Machine Learning in Systems Biology*, 2010.

[12] B.D. Macarthur, A. Ma'ayan, and I.R. Lemischka. Systems biology of stem cell fate and cellular reprogramming. *Nature Reviews on Molecular Cell Biology*, 10(10):672–81, 2009.

[13] T. M. Mitchell. *Machine learning*. McGraw Hill, 1996.

[14] S.H. Muggleton, R. King, and M. Sternberg. Protein secondary structure prediction using logic-based machine learning. *Protein Engineering*, 5(7):647–57, 1992.

[15] S. Ramaswamy et al. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences*, 98(26):15149–54, 2001.

[16] M. E. Ross et al. Classification of pediatric acute lymphoblastic leukemia by gene expression profile. *Blood*, 102(8):2951–2959, 2003.

[17] E. W. Stawiski, L. M. Gregoret, and Y. Mandel-Gutfreund. Annotating nucleic acid-binding function based on protein structure. *Journal of Molecular Biology*, 326(4):1065–79, 2003.

[18] A. Szaboova, O. Kuzelka, F. Zelezny, and J. Tolar. Mining frequent spatial docking patterns in zinc finger - dna complexes. In *ESBME 2010: 7th European Symposium on Biomedical Engineering*, 2010.

[19] A. Szilágyi and J. Skolnick. Efficient prediction of nucleic acid binding function from low-resolution protein structures. *Journal of Molecular Biology*, 358(3):922–33, 2006.

[20] A. Tamaddoni-Nezhad, R. Chaleil, A. Kakas, and S.H. Muggleton. Application of abductive ilp to learning metabolic network inhibition from temporal data. *Machine Learning*, 64:209–30, 2006.

[21] I. Trajkovski, F. Zelezny, N. Lavrac, and J. Tolar. Learning relational descriptions of differentially expressed gene groups. *IEEE Trans. Sys Man Cyb C*, 38(1):16–25, 2008.

[22] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.

[23] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

3 Ing. Filip Železný, Ph.D.

Filip Železný received his Ph.D. in Artificial Intelligence and Biocybernetics from the Czech Technical University in Prague (CTU) in 2003. In 2003-2004 he was a post-doctoral researcher at the University of Wisconsin in Madison and subsequently a visiting professor at the State University of New York in Binghamton. Currently, he is assistant professor and head of the Intelligent Data Analysis Research Group at CTU. His main research interests are machine learning and data mining, and their applications in bioinformatics. Filip was the local coordinator of the European research project SEVENPRO, served on program committees of conferences such as the International Conference on Machine Learning, chaired the 18th International Conference on Inductive Logic Programming, has been a project evaluator for the European Commission and member of the editorial boards of the Machine Learning Journal and the journal Advances in Artificial Intelligence.

Journal Papers

- Zahalka J., Zelezny F.: An Experimental Test of Occam's Razor in Classification. Accepted to **Machine Learning** pending minor revisions.
- Kuzelka O., Zelezny F.: Block-Wise Construction of Tree-like Relational Features with Monotone Reducibility and Redundancy. **Machine Learning** online first DOI 10.1007/s10994-010-5208-5 2010.
- Zakova M., Kremen P., Zelezny F., Lavrac N.: Automatic Knowledge Discovery Workflow Composition through Ontology-Based Planning. **IEEE Trans. Autom. Sci. and Eng.** online first DOI 10.1109/TASE.2010.2070838 2010
- Zelezny F., Lavrac N.: Guest editors' introduction: Special issue on Inductive Logic Programming (ILP-2008). **Machine Learning** 76(1):1-2, 2009
- Kuzelka O., Zelezny F.: A Restarted Strategy for Efficient Subsumption Testing. **Fundamenta Informaticae** 89(1):95-109, 2008
- Trajkovski I., Zelezny F., Lavrac N., Tolar J.: Learning Relational Descriptions of Differentially Expressed Gene Groups . **IEEE Trans. Sys Man Cyb C** 38(1):16-25, 2008
- Klema J., Novakova L., Karel F., Stepankova O., Zelezny F.: Sequential Data Mining: A Comparative Case Study in Development of Atherosclerosis Risk Factors . **IEEE Trans. Sys Man Cyb C** 38(1):3-15, 2008
- Zelezny F., Srinivasan A., Page D.: Randomized Restarted Search in ILP. **Machine Learning** 64(1-2):183–208, 2006
- Zelezny F., Lavrac N.: Propositionalization-Based Relational Subgroup Discovery with RSD. **Machine Learning** 62(1-2):33-63, 2006
- Gamberger D., Lavrac N., Zelezny F., Tolar J.: Induction of comprehensible models for gene expression datasets by subgroup discovery methodology. **Journal of Biomedical Informatics** 37(4):269-284, 2004.
- Zelezny F.: Efficiency-conscious Propositionalization for Relational Learning **Kybernetika** 4(3):275-292, 2004

Selected Conference Papers

- Kuzelka O., Zelezny F.: Block-Wise Construction of Acyclic Relational Features with Monotone Irreducibility and Relevancy Properties. **ICML 2009**: the 26th International Conference on Machine Learning (acceptance rate $\sim 25\%$)
- Kuzelka O., Zelezny F.: Fast Estimation of First-Order Clause Coverage through Randomization and Maximum Likelihood. **ICML 2008**: the 25th International Conference on Machine Learning (acceptance rate $\sim 25\%$)
- Zakova M., Zelezny F.: Exploiting Term, Predicate, and Feature Taxonomies in Propositionalization and Propositional Rule Learning. **ECML/PKDD 2007**: 18th European Conference on Machine Learning / 11th European Conference on Principles and Practice of Knowledge Discovery (acceptance rate $\sim 20\%$)

Scientometrics

- Citations: 99 by WoS (non-auto), 334 by Harzing's 'Publish or Perish'
- h-index: 5 by WoS, 8 by Harzing's 'Publish or Perish'