Gaussian Logic for Proteomics and Genomics

Ondřej Kuželka, Andrea Szabóová, Matěj Holec, and Filip Železný

Czech Technical University, Prague, Czech Republic
{kuzelon2,szaboand,holecmat,zelezny}@fel.cvut.cz,

1 Introduction

We describe a statistical relational learning framework called Gaussian Logic capable to work efficiently with combinations of relational and numerical data. The framework assumes that, for a fixed relational structure, the numerical data can be modelled by a multivariate normal distribution. We show how the Gaussian Logic framework can be used to predict DNA-binding propensity of proteins and to find motifs describing novel gene sets which are then used in set-level classification of gene expression samples[1].

2 A Probabilistic Framework

We address the situation where training examples have both *structure* and *real parameters*. One example may e.g. describe a measurement of the expression of several genes; here the structure would describe functional relations between the genes and the parameters would describe their measured expressions. Note that we allow different structures in different examples. In the genomic example, a training set thus may consist of measurements pertaining to different gene sets, each giving rise to a different structure of mutual relations between the genes.

To describe such training examples as well as learned models, we use a conventional first-order logic language $\mathcal{L}$ whose alphabet contains a distinguished set of constants $\{\mathsf{r}_1, \mathsf{r}_2, \ldots \mathsf{r}_n\}$ and variables $\{R_1, R_2, \ldots R_m\}$. Any substitution in our framework must map variables (other than) R_i only to terms (other than) r_j. The structure of an example is described by a (Herbrand) interpretation H, in which the constants r_i represent uninstantiated real parameters. The parameter values are then determined by a real vector $\boldsymbol{\theta}$. Thus each example is a pair $(H, \boldsymbol{\theta})$. Examples are assumed to be sampled from the distribution

$$P\left(H, \Omega_H\right) = \int_{\Omega_H} f_H\left(\boldsymbol{\theta}|H\right) P\left(H\right) \mathrm{d}\boldsymbol{\theta}$$

which we want to learn (where $\Omega_H \subseteq R^n$). Here, $P(H)$ is a discrete probability distribution on the countable set of Herbrand interpretations of $\mathcal{L}$. $f_H\left(\boldsymbol{\theta}|H\right)$

[1] A longer version of this paper appears at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2011) under the title: "Gaussian Logic for Predictive Classification".

are the conditional densities of the parameter values. The advantage of this definition is that it cleanly splits the possible-world probability into the discrete part $P(H)$ which can be modeled by state-of-the-art approaches such as Markov Logic Networks (MLN's) [3], and the continuous conditional densities $f_H(\boldsymbol{\theta}|H)$ which we elaborate here. In particular, we assume that $f(\boldsymbol{\theta}|H) = N(\boldsymbol{\mu}_H, \Sigma_H)$, i.e., $\boldsymbol{\theta}$ is normally distributed with mean vector $\boldsymbol{\mu}_H$ and covariance matrix Σ_H. The indexes H emphasize the dependence of the two parameters on the particular Herbrand interpretation that is parameterized by $\boldsymbol{\theta}$.

To learn $P(H, \Omega_H)$ from a sample E, we first discuss a strategy that suggests itself readily. We could rely on existing methods (such as MLN's) to learn $P(H)$ from the multi-set $\mathcal{H}$ of interpretations H occurring in E. Then, to obtain $f(\boldsymbol{\theta}|H)$ for each $H \in \mathcal{H}$, we would estimate $\boldsymbol{\mu}_H, \Sigma_H$ from the multi-set $\hat{\Omega}_H$ of parameter value vectors $\boldsymbol{\theta}$ associated with H in the training sample E. The problem of this approach is that, given a fixed size of the training sample, when $\mathcal{H}$ is large, the multi-sets $\hat{\Omega}_H, H \in \mathcal{H}$ will be small, and thus the estimates of $\boldsymbol{\mu}_H, \Sigma_H$ will be poor.

Our strategy is instead to discover *Gaussian features* of the training examples. A Gaussian feature is logic formula which, roughly said, extracts some of the parameter values for each example into a vector such that this vector is approximately normally distributed across the training sample. For example, the intentionally simplistic feature

$$\exists G_1, G_2 \quad \mathrm{expr}(G_1, R_1) \wedge \mathrm{expr}(G_2, R_2) \wedge \mathrm{regulates}(G_1, G_2)$$

contains two standard FOL variables G_1, G_2 and two distinguished variables R_1, R_2, and indicates that expressions of any two genes (G_1, G_2) in the regulation relation are co-distributed normally. The corresponding mean vector and covariance matrix are then estimated from all training examples whose structures contain one or more pairs of such related genes. The learned features then act as constraints on the target distribution $P(H, \Omega_H)$. By choosing the number of employed features, we are able to trade off between under- and over-constraining the target distribution model.

In general, the problem of estimating parameters of Gaussian features is an NP-hard problem. However, it is tractable for a class of features, *conjunctive tree-like features* for which we have devised also an efficient feature construction algorithm based on the feature-construction algorithm from [8]. It shares most of the favourable properties of the original algorithm like detection of redundant features.

3 Predictive Classification Applications

A straightforward application of the Gaussian-logic framework is in Bayesian classification. We address a case study involving an important problem from biology: prediction of DNA-binding propensity of proteins. Several computational approaches have been proposed for the prediction of DNA-binding function from protein structure. It has been shown that electrostatic properties of proteins are

good features for predictive classification (e.g. [1, 2]). A more recent approach is the method of Szilágyi and Skolnick [9] who created a logistic regression classifier based on 10 features also including electrostatic properties.

Here, we use Gaussian logic to create a model for capturing distributions of positively charged amino acids in protein sequences. We split each protein into consecutive non-overlapping *windows*, each containing l_w amino acids (possibly except for the last window which may contain less amino acids). For each window of a protein P we compute the value a_i^+/l_w where a_i^+ is the number of positively charged amino-acids in the window i. Then for each protein P we construct an example $e_P = (H_P, \boldsymbol{\theta}_P)$ where $\boldsymbol{\theta}_P = \left(a_1^+/l_w, a_2^+/l_w, \ldots, a_{n_P}^+/l_w\right)$ and $H_P = w(1, r_1)$, $next(1, 2)$, $\ldots$, $next(n_P - 1, n_P)$, $w(n_P, r_P)$. We constructed only one feature $F_{non} = w(A, R_1)$ for non-DNA-binding proteins since we do not expect this class of proteins to be very homogeneous. For DNA-binding proteins, we constructed a more complex model by selecting a set of features using a greedy search algorithm. The greedy search algorithm optimized classification error on training data. Classification was performed by comparing, for a tested protein, the likelihood-ratio of the two models (DNA-binding and non-DNA-binding) with a threshold selected on the training data. We estimated the accuracy of this method using 10-fold cross-validation (always learning parameters and structure of the models and selecting the threshold and window length l_w using only the data from training folds) on a dataset containing 138 DNA-binding proteins (PD138 [9]) and 110 non-DNA-binding proteins (NB110 [1]). The estimated accuracies (*Gaussian Logic*) are shown in Table 1. The method performs similarly well as the method of Szilagyi et al. [9] (in fact, it outperforms it slightly but the difference is rather negligible) but uses much less information. Next, we were interested in the question whether the machinery of Gaussian logic actually helped improve the predictive accuracy in our experiments or whether we could obtain the same or better results using only the very simple feature $F = w(A, R_1)$ also to model the DNA-binding proteins, thus ignoring any correlation between charges of different parts of a protein (*Baseline Gaussian Logic* in Table 1). Indeed, the machinery of Gaussian Logic appears to be helpful from these results.

Method	**Accuracy [%]**
Szilágyi et al.	81.4
Baseline Gaussian logic	78.7
Gaussian logic	**81.9**

Table 1. Accuracies estimated by 10-fold cross-validation on PD138/NB110.

It is interesting how well the Gaussian-logic model performed considering the fact that it used so little information (it completely ignored types of positively charged amino acids and it also ignored negative amino acids). The model that we presented here can be easily extended, e.g. by adding secondary-structure information. The splitting into consecutive windows used here is rather artificial

and it would be more natural to split the sequence into windows corresponding to secondary-structure units (helices, sheets, coils). The features could then distinguish between consecutive windows corresponding to different secondary-structure units.

Next, we used Gaussian logic to search for novel definitions of gene sets with high discriminative ability. This is useful in set-level classification methods for prediction from gene-expression data [5]. Set-level methods are based on aggregating values of gene expressions contained in pre-defined gene sets and then using these aggregated values as features for classification. We constructed examples $(H_S, \boldsymbol{\theta}_S)$ from gene-expression samples and KEGG pathways [7] as follows. For each gene g_i, we introduced a logical atom $g(g_i, r_i)$ to capture its expression level. Then we added all relations extracted from KEGG as logical atoms $relation(g_i,\ g_j,\ relationType)$. We also added a numerical indicator of class-label to each example as a logical atom $label(\pm 1)$ where $+1$ indicates a positive example and -1 a negative example. Finally, for each gene-expression sample S we constructed the vector of the gene-expression levels $\boldsymbol{\theta}_S$. Using our feature construction algorithm we constructed a large set of tree-like features involving exactly one atom $label(L)$, at least one atom $g(G_i, R_i)$ and relations *expression*, *repression*, *activation*, *inhibition*, *phosphorylation*, *dephosphorylation*, *state* and *binding/association*. After that we selected a subset of features according to the correlation of the average expression of the involved genes with the class label, which can be extracted from the estimated Gaussian-feature parameters.

Dataset	**GL**	**FCF**	**Dataset**	**GL**	**FCF**
Collitis	80.0	**89.4**	Pheochromocytoma	**64.0**	56.0
Pleural Mesothieloma	**94.4**	92.6	Prostate cancer	**85.0**	80.0
Parkinson 1	52.7	**54.5**	Squamus cell carcinoma	**95.5**	88.6
Parkinson 2	**66.7**	63.9	Testicular seminoma	58.3	**61.1**
Parkinson 3	62.7	**77.1**	**Wins**	**5**	4

Table 2. Accuracies of set-level-based classifiers with Gaussian-logic features and FCF-based features, estimated by leave-one-out cross-validation.

We have constructed features using a gene-expression dataset from [4] which we did not use in the subsequent predictive classification experiments. We have compared gene sets constructed by the outlined procedure with gene sets based on so called *fully-coupled fluxes (FCFs)* which are biologically-motivated gene sets used previously in the context of set-level classification [5]. We constructed the same number of gene sets for our features as was the number of FCFs. The accuracies of an SVM classifier (estimated by leave-one-out cross-validation) are shown in Table 2. We can notice that the gene sets constructed using our novel method performed equally well as the gene sets based on fully-coupled fluxes. Interestingly, our gene sets contained about half the number of genes as compared to FCFs and despite that they were able to perform equally well.

4 Conclusions and Future Work

We have introduced a novel relational learning system capable to work efficiently with combinations of relational and numerical data. The experiments gave us some very promising results, slightly outperforming methods based on features hand-crafted by biologists using only automatically constructed Gaussian features. Furthermore, there are other possible applications of Gaussian logic in predictive classification settings which were not discussed in this paper. For example, finding patterns that generally correspond to highly correlated sets (not necessarily correlated with the class) of genes may have applications with group-lasso based classification approaches [6].

Acknowledgement: We thank the anonymous reviewers of MLSB and ECML PKDD for their valuable comments. This work was supported by the Czech Grant Agency through project 201/09/1665 *Bridging the Gap between Systems Biology and Machine Learning* and project 103/11/2170 *Transferring ILP techniques to SRL*.

References

1. Shandar Ahmad and Akinori Sarai. Moment-based prediction of dna-binding proteins. *Journal of Molecular Biology*, 341(1):65 – 71, 2004.
2. Nitin Bhardwaj, Robert E. Langlois, Guijun Zhao, and Hui Lu. Kernel-based machine learning protocol for predicting DNA-binding proteins. *Nucleic Acids Research*, 33(20):6486–6493.
3. Pedro Domingos, Stanley Kok, Daniel Lowd, Hoifung Poon, Matthew Richardson, and Parag Singla. Probabilistic inductive logic programming. chapter Markov logic, pages 92–117. Springer-Verlag, 2008.
4. William A Freije et al. Gene expression profiling of gliomas strongly predicts survival. *Cancer Res*, 64(18):6503–10, 2004.
5. Matěj Holec, Filip Železný, Jiří Kléma, and Jakub Tolar. Integrating multiple-platform expression data through gene set features. In *Proceedings of the 5th International Symposium on Bioinformatics Research and Applications*, ISBRA '09, pages 5–17. Springer-Verlag, 2009.
6. Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 433–440. ACM, 2009.
7. M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori. The kegg resource for deciphering the genome. *Nucleic Acids Research*, 1, 2004.
8. O. Kuželka and F. Železný. Block-wise construction of tree-like relational features with monotone reducibility and redundancy. *Machine Learning*, 83:163–192, 2011.
9. András Szilágyi and Jeffrey Skolnick. Efficient prediction of nucleic acid binding function from low-resolution protein structures. *Journal of Molecular Biology*, 358(3):922 – 933, 2006.