Gaussian Logic for Predictive Classification

Ondřej Kuželka, Andrea Szabóová,
Matěj Holec, Filip Železný

Czech Technical University in Prague

Outline

- Introduction – statistical relational learning
- Gaussian Logic
- An experiment with proteins
- An experiment with gene-expression data
- Conclusions and future work

INTRODUCTION

Statistical Relational Learning (with Continuous Random Variables)

- **SRL combines relations (first-order logic) with probability**
- Some SRL frameworks are able to handle continuous random variables (e.g. Hybrid MLN, Bayesian Logic Programming)
- **Unfortunately**: Learning the structure (i.e. the features) is very hard in these frameworks in the presence of continuous variables

GAUSSIAN LOGIC – A SIMPLE SRL APPROACH

Gaussian Logic

- Let us assume that examples (possible worlds) are sampled from a distribution

$$P(S, \Omega) = \int_{\Omega} f_S(\theta|S)\, P(S) d\theta$$

relational structure with wild-cards (variables) for real numbers

subset of R^n

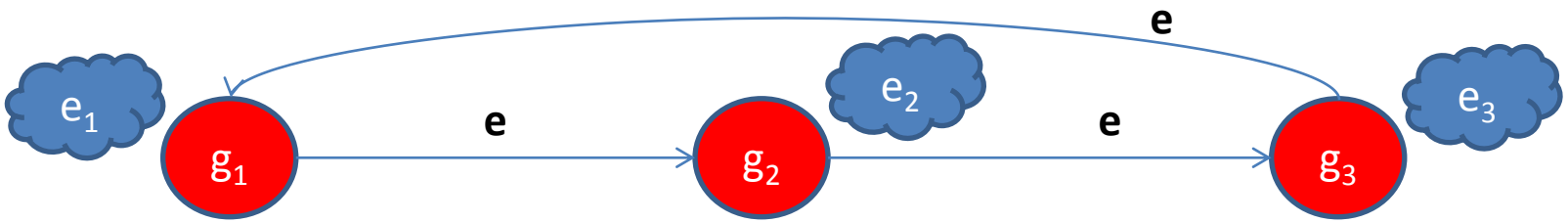
probability of the relational structure S

multivariate normal distribution conditioned on the relational structure

More Intuition – How a Possible World Could be Drawn from the Distribution

$$P(S,\Omega) = \int_{\Omega} f_S(\theta|S)\, P(S) d\theta$$

1. Draw a relational structure S from $P(S)$, e.g.



2. Get the corresponding multivariate normal distribution $f_S(\theta|S)$, e.g.

$$\boldsymbol{X} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix}, \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 \\ 0.2 & 0.2 & 1 \end{bmatrix}, \qquad \boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

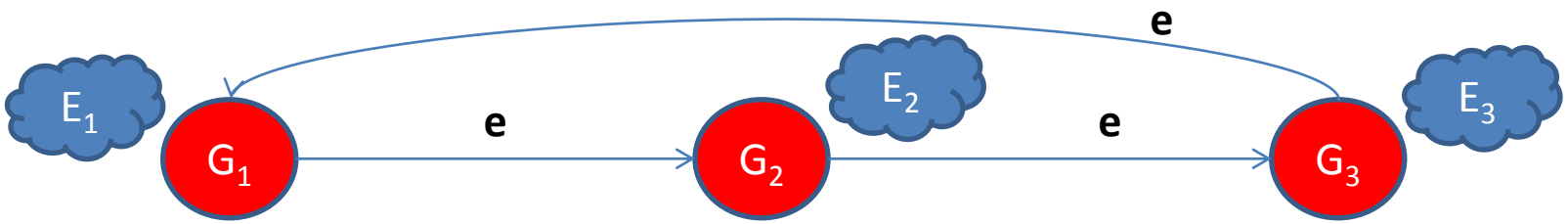
3. Sample a real vector from the distribution

Modeling the Distributions

- $P(S)$ can be modeled using established SRL methods (e.g. MLNs) – it does not contain numerical variables
- We need to be able to model also $f_S(\theta|S)$ for any S – we need to be able to estimate $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$
- We assume that there are relational sub-structures – ***Gaussian features*** with the same multivariate joint distributions

Gaussian Features

$$F = g(G_1, E_1), expresses(G_1, G_2), g(G_2, E_2)$$

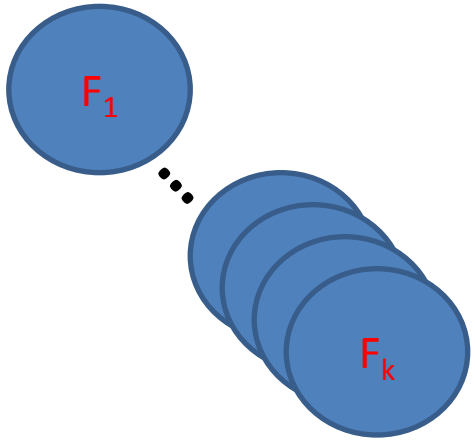


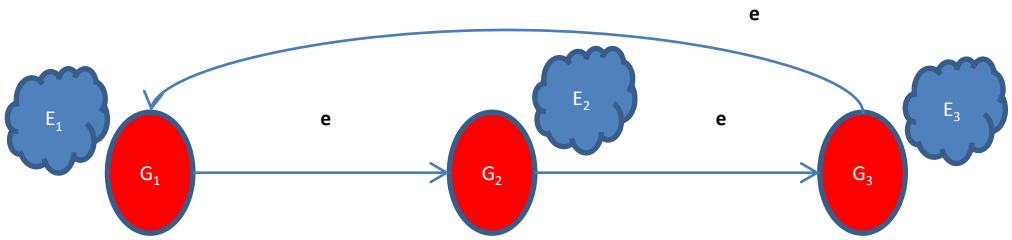
- If we assume that F is a Gaussian feature, we assume that the marginal distributions of G_1, G_2, G_1, G_3 and G_2, G_3 are the same (this is true for the example from one of the previous slides):

$$\boldsymbol{X} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix}, \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 \\ 0.2 & 0.2 & 1 \end{bmatrix}, \qquad \boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

- For each Gaussian feature we have its mean $\boldsymbol{\mu_F}$ and covariance $\boldsymbol{\Sigma_F}$

Construction of $f_S(\theta|S)$ from Gaussian Features



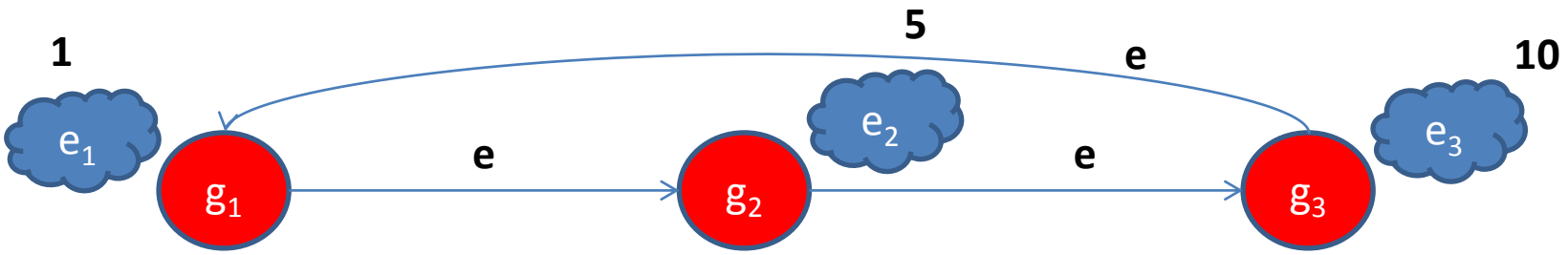


$$\begin{bmatrix} v_1 & c_{1,2} & \cdots & c_{1,n-} \\ c_{1,2} & v_2 & & c_{2,n-} \\ \vdots & & \ddots & \\ c_{1,n-1} & c_{2,n-1} & & v_{n-1} \\ c_{1,n} & c_{2,n} & \cdots & c_{n-1,n} \end{bmatrix}$$

Estimation (1): Gaussian Features and Sample Sets

- A feature in our framework is a first-order-logic formula containing a set of free distinguished variables $R = \{R_1, \ldots, R_k\}$

$$F = \exists G_2 : g(G_1, R_1), expresses(G_1, G_2), expresses(G_2, G_3), g(G_3, R_2)$$



- A sample set $S(F, e)$ is a multi-set of vectors extracted by *F* from *e*, so e.g. here:

$$S(F, e) = \{(1,10)^T, (5,1)^T, (10,5)^T\}$$

Estimation (2): Estimating the Mean and Covariance from Sample Sets

- For a Gaussian feature F and each example e_i we can compute sample sets $S(F, e_i) = \{\overrightarrow{s_1}, \ldots, \overrightarrow{s_{n_i}}\}$
- **The samples $\overrightarrow{s_1}$ are not independent,** we can't compute $\boldsymbol{\mu_F}$ and $\boldsymbol{\Sigma_F}$ simply from $\bigcup_i S(F, e_i)$

- **We can take one sample from each sample set and compute the mean and covariance – this is a little wasteful**

Estimation (3): Estimating the Mean and Covariance from Sample Sets

$$\mu(F,e) = \frac{1}{|S(F,e)|} \sum_{s_i \in S(F,e)} s_i$$

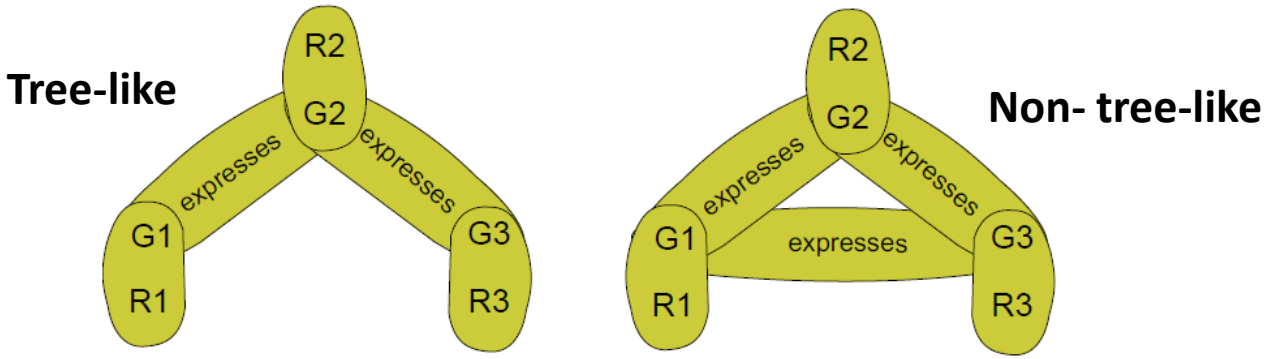
$$\Sigma(F,e) = \frac{1}{|S(F,e)|} \sum_{s_i \in S(F,e)} (s_i - \mu(F,e))(s_i - \mu(F,e))^T$$

$$\hat{\mu}_F = \frac{1}{|E|} \sum_{e_i \in E} \mu(F,e_i)$$

$$\hat{\Sigma}_F = \frac{1}{|E|} \sum_{e_i \in E} (\Sigma(F,e_i) + \mu(F,e_i)\mu(F,e_i)^T) - \hat{\mu}_F \hat{\mu}_F{}^T$$

Note: Complexity of Estimation of Mean and Covariance of Gaussian Features

- We can estimate μ_F and Σ_F of a Gaussian feature F - **but it is NP-hard**, so it can take time
- For existentially quantified tree-like features we can do this **in polynomial time** (and possibly also for bounded-tree-width features)

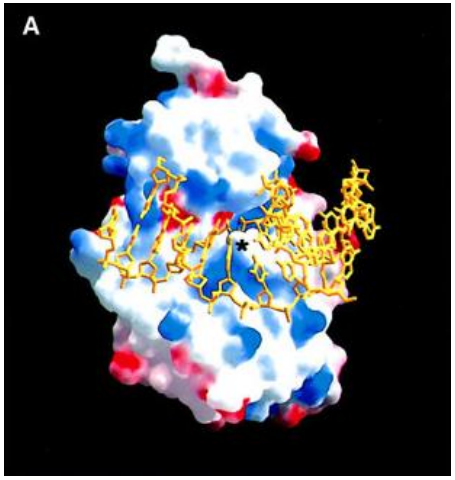


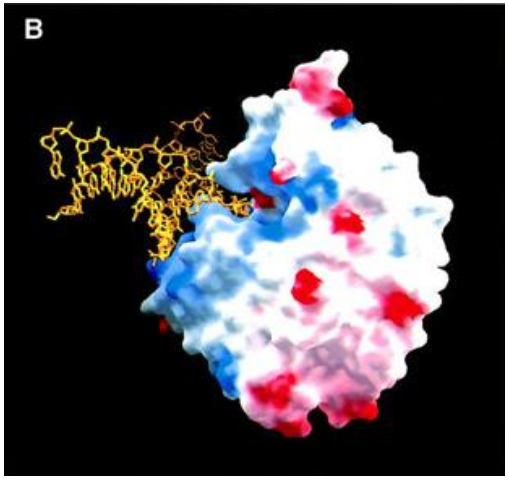
Estimation of DNA-binding Propensity of Proteins

AN EXPERIMENT WITH PROTEINS

Predicting DNA-binding Propensity

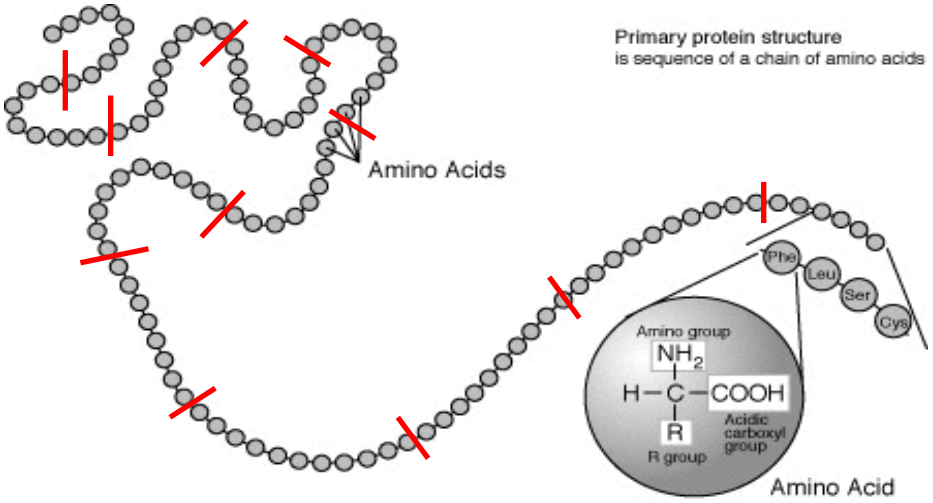
- **Motivation:** Positively charged amino acids tend to be clustered in DNA-binding proteins – in 3D space. **Can we see this effect in sequences? - We will model distribution of positive charge in sequences using Gaussian Logic.**





DNA-binding Propensity –Method

1. Split the protein sequences to non-overlapping windows and compute fraction of positive residues in each of them:



2. Construct a set of discriminative Gaussian features

3. Estimate two models – binding/non-binding

4. Use these models to predict propensity of new proteins (by comparing ratio of likelihoods of the two models to a threshold)

DNA-binding Propensity - Results

- The model based on Gaussian features was able to achieve the same accuracy (81.9% vs 81.4%) as a state-of-the-art method based on physicochemical properties of proteins (Szilágyi et al., 2006) while using much less information

Estimation of DNA-binding Propensity of Proteins

AN EXPERIMENT WITH GENE EXPRESSION DATA

Automatic Invention of Gene Sets for Set-level Predictive Classification

- Gaussian features are able to capture correlations => **We can use it to find descriptions of sets of genes on average highly correlated with the class**
- These sets may be then used in set-level based classification

Set-level Classification - Results

- We compared our automatically generated gene sets with so-called *fully-coupled fluxes* and obtained similar results while using about half the genes (the number of sets (= attributes) being equal)

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

- Example feature which exhibited high correlation with the class-label on a training dataset:

$$F = g(A, R_1), \textit{phosphorylates(A,B)},$$
$$\textit{g(B,}R_2\textit{), phosphorylates(A,C), g(C,}R_3\textit{)}$$

Conclusions

- Despite its simplicity, Gaussian logic obtained promising results in two real-world domains

- This indicates the potential of statistical relational learning methods capable to work with continuous variables – **but we may possibly need something more sophisticated than Gaussian logic**

THANK YOU!