Shrinking Covariance Matrices using Biological Background Knowledge

Ondřej Kuželka and Filip Železný

Czech Technical University, Prague, Czech Republic

Abstract. We propose a novel method for covariance matrix estimation based on shrinkage with a target inferred from biological background knowledge using methods of inductive logic programming. We show that our novel method improves on the state of the art when sample sets are small and some background knowledge expressed in a subset of first-order logic is available. As we show in the experiments with genetic data, this background knowledge can be even only indirectly relevant to the modeling problem at hand.

1 Introduction

An important problem in modelling of gene expression data is estimation of large covariance matrices. Only quite recently it has been realized that the vast amount of structured knowledge available in databases like KEGG [5] could be used to improve the estimation of covariance matrices. So far, all the approaches following this idea used biological knowledge only in a restricted way. For example in [3], shrinkage targets for covariance matrices have been constructed with non-diagonal entries being non-zero for genes from the same gene groups. We introduce a novel method that exploits structured knowledge in a non-trivial way and improves on a state-of-the-art covariance estimation method.

2 SGLNs: Simple Gaussian Logic Networks

We will be working with existentially quantified conjunctions of first-order logical atoms (*conjunctions*), which we will also treat as sets. We say that a conjunction C *θ-subsumes* a conjunction D (denoted by $C \preceq_\theta D$) iff there is a substitution θ such that $C\theta \subseteq D$. For example, if $C = a(B, C)$ and $D = a(x, y), b(y, z)$ then $C \preceq_\theta D$ because $C\theta \subseteq D$ for $\theta = \{A/x, B/y\}$. Next, we describe a framework termed *simple gaussian logic networks* (SGLNs) which borrows ideas from Markov logic networks and Bayesian logic programs [4].

Definition 1 (Simple Gaussian Logic Networks). *A Simple Gaussian Logic Network (SGLN) is a triple (G, R, N) where $G = (g_i)$ (gaussian atoms) is a list of ground first-order atoms, R (rules) is a set of conjunctions and N (network) is a ground conjunction. A normal distribution $N(\mu, \Sigma)$ is said to comply with a SGLN $S = (G, R, N)$ if for $P = (p_{ij}) = \Sigma^{-1}$ it holds $p_{ij} = 0$ whenever there is no rule $r \in R$ and substitution θ such that $\{g_i, g_j\} \subseteq r\theta \subseteq N$.*

It is well-known [7] that a multivariate normal distribution with covariance matrix Σ and precision matrix $P = \Sigma^{-1} = (p_{ij})$ can be viewed as a Markov network in which there are edges between any two variables (nodes) v_i and v_j for which $p_{ij} \neq 0$. Thus, SGLNs define an independence structure over the variables corresponding to the gaussian atoms g_i.

Example 1. Let us have a SGLN $S = (G, R, N)$ where $G = (g(a), g(b), g(c))$, $R = \{g(X), edge(X, Y), edge(Y, Z), g(Z)\}$ and $N = g(a), g(b), g(c), edge(a, b), edge(b, c)$. Then any normal distribution with covariance matrix Σ (below) complies with S.

$$\Sigma^{-1} = P = \begin{bmatrix} x & 0 & y \\ 0 & z & 0 \\ y & 0 & w \end{bmatrix}$$

We have not explained yet how to obtain a covariance matrix complying to a given SGLN S and maximizing likelihood on a set of training examples E. The problem of estimating a covariance matrix with a given pattern of zeros in its inverse is known as *covariance selection* [1]. Given an ordinary covariance matrix estimated from data, one can find the maximum-likelihood estimate with a prescribed zero-pattern by means of convex optimization. Very often one has too few data samples compared to the number of variables. In such a case, it is impossible to estimate the covariance matrix reliably as $\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(x_i - \overline{x})^T$. Instead, we have to apply a more advanced method, for example *shrinkage-based estimation* [6]. The basic idea of shrinkage is to obtain convex combinations of high-dimensional and lower dimensional models. The covariance selection method combined with the shrinkage based estimation of unconstrained covariance matrices gives us an effective tool for learning with SGLNs. First, we obtain an estimate of covariance matrix $\hat{\Sigma}$ using the shrinkage-based approach. Next, we use $\hat{\Sigma}$ as input together with the zero-pattern given by a given SGLN to the covariance selection procedure which gives us the estimate of the covariance matrix complying with the SGLN.

3 Estimating Covariance Matrices using SGLNs

In this section we briefly describe a simple method that uses SGLNs to improve covariance matrix estimation (Algorithm 1). It turns out that the covariance matrix obtained from an appropriate SGLN can be a very good shrinkage target. The rationale behind the SGLN-rule-learning part of the method is as follows. We assume that there are some rules which capture the dependency structure of the estimated distribution. First, we create a set of positive examples from unions of d-neighbourhoods[1] of *most correlated*[2] pairs of gaussian literals and a set of negative examples from the *least correlated* ones. We can expect that the rules which θ-subsume many positive examples and few negative examples would be good rules of the SGLN.

[1] the neighbourhoods of depth d in the (hyper)-graph theoretical sense

[2] Here, the word *correlation* refers to partial correlations, i.e. $p_{ij}/\sqrt{p_{ii}p_{jj}}$ where p_{ij} are entries of the inverse of the covariance matrix.

Algorithm 1 CovEstimate:

1: **Input:** Samples S, Set of gaussian atoms G, Conjunction N;
2: $\Sigma_0 \leftarrow$ Estimate covariance matrix from S using [6]
3: $P = (p_{ij}) \leftarrow \Sigma_0^{-1}$ /* P is the so-called *precision matrix* */
4: $PosExs \leftarrow \max\{k, \lfloor \frac{|V|}{2} \rfloor\}$ pairs $(i,j), i<j$ with highest $|p_{ij}/\sqrt{p_{ii}p_{jj}}|$
5: $NegExs \leftarrow \max\{k, \lfloor \frac{|V|}{2} \rfloor\}$ pairs $(i,j), i<j$ with lowest $|p_{ij}/\sqrt{p_{ii}p_{jj}}|$
6: $PosExs^*, NegExs^* \leftarrow$ Convert $PosExs$ and $NegExs$ to first-order clauses /*see the main text*/
7: $R \leftarrow$Construct a set of *good rules* using an ILP algorithm /*see main text*/
8: $\Sigma_1 \leftarrow$ Obtain an estimate of covariance matrix from Σ_0 complying with (G, R, N)
9: **return** $t \cdot \Sigma_0 + (1-t) \cdot \Sigma_1$ /* with t selected using internal cross-validation */

Example 2. Let us have the network N from Example 1 and a set of samples M. Let us suppose that we obtained the following covariance matrix by the shrinkage-based estimation method applied on samples from M.

$$\hat{\Sigma}^{-1} = \begin{bmatrix} 1 & 0 & -0.5 \\ 0 & 0.75 & 0 \\ -0.5 & 0 & 1 \end{bmatrix}$$

Now, let us construct the sets of positive and negative examples according to the recipe described in the preceding paragraphs. We set $d = 1$. Then $E^+ = \{e_1\}$ where e_1 corresponds to pair of gaussian literals $g(a), g(c)$ and $e_1 = (\{g(a), edge(a,b)\} \setminus \{g(b), g(c)\}) \cup (\{edge(b,c), g(c)\} \setminus \{g(a), g(b)\}) = g(a), edge(a,b), edge(b,c), g(c)$. Analogically, $E^- = \{e_2\}$ where $e_2 = g(a), g(b)$. It depends on the chosen language bias which rules would be induced. For example, if rules were restricted to connected clauses, the correct rule $g(X), edge(X,Y), edge(Y,Z), g(Z)$ from Example 1 would be one of them.

4 Experimental Results and Conclusions

In this section we show how SLGNs can be applied to covariance estimation of gene expression data. We used datasets from GEO database [2], namely GDS1209 and GDS1220. For each dataset, we generated 65 smaller datasets each corresponding to one pathway from KEGG database [5]. For each of these, we created a network N consisting of relations contained in KEGG, e.g. relation of *activation* or *phosphorylation* among proteins etc. Then we compared the baseline shrinkage-based method (Shr.) [6] with our novel method (SGLN). SGLN clearly outperformed the existing method (cf. Table 1). Nevertheless, one could still argue that we could obtain the same or even better improvement if we replaced the matrix on line 8 of Algorithm 1 by a matrix obtained with a different zero pattern which would have the same number K of non-zero elements but these non-zero elements would correspond to K most correlated pairs of variables. In other words, one could ask whether the use of background knowledge brings us any benefits. Therefore we performed experiments also with this suggested method (Top-K), but again SLGN outperformed it. Using cross-validation, we measured both likelihood on unseen data and RMSE of estimates of values with

Table 1. Results on the gene expression datasets. **Top:** Average ratios of likelihoods on test data obtained by the different methods (smaller is better). **Bottom:** Average ratios of RMSEs on test data obtained by the different methods (smaller is better). Numbers in parentheses correspond to number of wins/losses/ties.

Dataset	L - SGLN x Shr.	L - SGLN x Top-K	L - Top-K x Shr.
GDS1209_15	0.907 (62/3/0)	0.979 (39/25/1)	0.927 (64/1/0)
GDS1209_39	0.939 (63/2/0)	0.980 (53/12/0)	0.958 (64/1/0)
GDS1220_10	0.937 (64/1/0)	0.963 (55/10/0)	0.974 (54/5/6)
GDS1220_44	0.942 (63/2/0)	0.975 (59/6/0)	0.966 (54/2/9)
Dataset	**RMSE - SGLN x Shr.**	**RMSE - SGLN x Top-K**	**RMSE Top-K x Shr.**
GDS1209_15	0.899 (57/8/0)	0.968 (44/20/1)	0.928 (58/7/0)
GDS1209_39	0.983 (50/15/0)	0.996 (37/28/0)	0.987 (53/12/0)
GDS1220_10	0.983 (52/13/0)	0.991 (49/16/0)	0.991 (51/8/6)
GDS1220_44	0.981 (58/7/0)	0.994 (43/22/0)	0.987 (51/5/9)

a randomly selected half of the variables (genes) set to known values. A one-sided binomial test ($\alpha = 0.05$) on the *number of wins* has shown that SGLN was always significantly better than Shr. and that in all but two cases SGLN was also significantly better than Top-K.

In this paper, we have introduced a novel method that is able to exploit structured background knowledge for estimation of covariance matrices and outperforms an existing state-of-the-art method. An interesting fact is that even though most of the background knowledge was not directly related to gene co-expression as the pathways contain more relations regarding products of the respective genes, it increased accuracy. It would be therefore interesting to interpret some of the learned rules from the biological point of view.

Acknowledgements: OK has been supported by the Grant Agency of the Czech Technical University in Prague, grant SGS10/073/OHK3/1T/13. FŽ has been supported by the Czech Grant Agency through project 201/09/1665.

References

1. J. Dahl, L. Vandenberghe, V. Roychowdhury, Covariance selection for non-chordal graphs via chordal embedding. *Optimization Methods and Software*, 23 (4), 501-520, 2008.
2. R. Edgar, M. Domrachev, and A. Lash. Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.*, 30 (1), 2002.
3. Feng Tai, Wei, Pan, Incorporating prior knowledge of gene functional groups into regularized discriminant analysis of microarray data, *Bioinformatics*, 2007
4. L. Getoor, B. Taskar, Introduction to Statistical Relational Learning, The MIT Press, 2007.
5. M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori. The KEGG resource for deciphering the genome. Nucleic Acids Res, 32, 2004.
6. J. Schäffer, K.Strimmer, A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics, *Statistical Applications in Genetics and Molecular Biology*, 4 (1), 2005.
7. T. P. Speed, H. T. Kiiveri, Gaussian Markov Distribution over Finite Graphs, *The Annals of Statistics*, 14 (1), 1986.