

Tel Aviv University
Raymond and Beverly Sackler Faculty of Exact Sciences
The Blavatnik School of Computer Science

*Using Iterative Ridge Regression to
Explore Associations Between
Conditioned Variables*

Thesis submitted in partial fulfillment of the requirements
for the M.Sc. degree of Tel-Aviv University
by:
Nimrod Bar-Yaakov

The research work for this thesis has been carried out at Tel-Aviv University under the
supervision of
Prof. Nathan Intrator

February 2011

Acknowledgments

I would like to thank Prof. Nathan Intrator for his professional guidance and patience along the research years.

To Prof. Zehava Grossman of the Central Virology Laboratory at Tel Hashomer for the initial problem presentation and the joint work and collaboration toward a shared solution.

To my bosses at Orbotech for their cooperation and patience.

And finally to my wife and kids for their love and support.

Abstract

Correlations and associations between variables tend to form or dissolve as a result of external effect. The effect can be temporal, or episodically – where it is related to a specific context or event.

In some cases we can mark this specific context as a variable or set of variables.

In this study, we want to map the associations that form with the appearance of the context variable, which we term as *condition variable*. The associations that will appear with the appearance of the condition variable will be termed as *conditioned associations*.

Specifically, we studied a specific case of joint probability mapping, where the information presented is the probabilistic associations of random variables under a certain condition variable.

Bayesian and Dependency networks graphically map the joint probabilities of random variables, though both networks may identify associations that are independent of the condition (background associations). Since the background associations have the same topological features as conditioned associations, it is difficult to discriminate between conditioned and non-conditioned associations, which results in a major increase in the search space.

We introduce a modification of the dependency network method, which produces a directed graph, containing only condition-related associations. The graph nodes represent the random variables and the graph edges represent the associations that arise under the condition variable. This method is based on ridge-regression, where one can utilize a numerically robust and computationally efficient algorithm implementation.

We illustrate the method's efficiency in the context of a medically relevant process, the emergence of drug resistant variants of HIV in drug-treated, HIV-infected people. Our mapping was used to discover associations between variants that are conditioned by the initiation of a particular drug treatment regimen. We have demonstrated that our method can recover known associations of such treatment with selected resistance mutations as well as documented associations between different mutations. Moreover, our method revealed novel associations that are statistically significant and biologically plausible.

Table of Contents

ACKNOWLEDGMENTS.....	II
ABSTRACT.....	III
CHAPTER 1: INTRODUCTION	1
1.1 Problem Example	1
1.2 Naïve testing of a single association.....	2
1.3 Using unsupervised learning for locating conditioned patterns	3
1.4 Using Bayesian Networks to infer about conditioned patterns.....	3
1.5 Dependency Networks	7
1.6 Summary of current state of the art and the proposed method.....	8
CHAPTER 2: METHOD – ITERATIVE RIDGE REGRESSION (IRR)	9
2.1 Problem Definition	9
2.2 Selection or regularized regression method	10
2.3 Iterative interaction discovery	11
2.4 Analytical proof of the IRR method.....	12
2.5 Complexity of the IRR algorithm.....	14
CHAPTER 3: COMPARING THE IRR ALGORITHM WITH BAYESIAN NETWORKS.....	15
3.1 Data set construction	15
3.2 Simulation overview	15
3.3 Results.....	16
3. 4 Summary of the comparison results.....	17

3. 5 Conclusions	18
CHAPTER 4: APPLICATION OF THE IRR FOR PREDICTION OF HIV RESISTANCE MUTATION PATTERNS.....	20
4.1 Data and Method	21
4.2 Results.....	22
4.3 Comparison to previous findings.....	23
4.4 Analysis of inconsistencies with previous findings.....	23
4.5 IRR new findings	23
CHAPTER 5: SUMMARY AND CONCLUSIONS	25
APPENDIX	26
APPENDIX A.....	26
IRR Pseudo code.....	26
APPENDIX B.....	28
Detailed Algorithm demonstration –.....	28
APPENDIX C	31
Exact details of the IRR – BNT simulation	31
BIBLIOGRAPHY	34

Chapter 1: Introduction

The search methods of associations, dependencies, and correlations between random variables have been thoroughly studied during the last five decades, and in some cases, there is a need to recover a specific type of associations: those that result from the existence of a joint event or context. For example, dependencies between variables can change with the appearance of another variable; different pattern of associations are created with the appearance of this variable. This variable will herein be referred to as **Condition**, and the associations generated by this condition – **Conditioned associations**.

Mathematically speaking, we want to find a subset S out of a set of random variables $\{x_1, x_2, \dots, x_n\}$ where $p(S/c)$ significantly differs from $p(S)$.

Exploring the conditioned associations can recover hidden mechanisms or processes that operate only when the condition exists. Such mechanisms can be quantitative and qualitative modifications of elements of the immune system in response to pathogens, emergency services recruitment upon a catastrophic event, or DNA mutations in an infecting virus arising under the selection pressure of a novel treatment.

Pearl [1] specifies the benefits of using a graph representation of joint probability mapping, including clarity, convenience and economical representation. Graphical notation and terminology will be used further in this paper.

1.1 Problem Example

Assume there are three random binary variables: Car rental shortage (x_1), Airport closure (x_2), Western wind (x_3), and a Condition – Icelandic volcano eruption (c).

Exploring the correlations between x_1 , x_2 and x_3 will show that x_1 correlates with x_2 while none of them correlates with x_3 . Thus, normally there is no apparent correlation between western wind and airport closure or car rental shortage, or $p(x_1, x_2/x_3) = p(x_1, x_2)$. Yet, when an Icelandic volcano erupts, a western wind can carry the ash cloud over the airport, causing it to shut down, in which case $p(x_2, x_3/c) \gg p(x_2, x_3)$. In a graph representation, retrieving the $x_3 \rightarrow x_2$ edge will suggest that a new association was created under the Condition c (see figure 1).

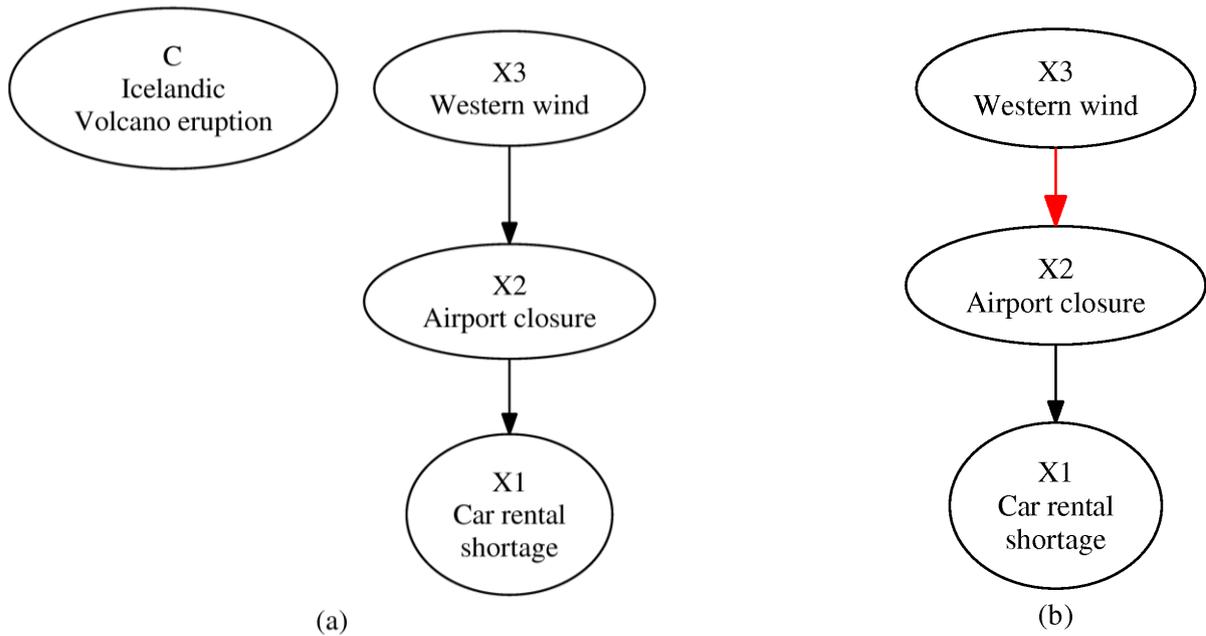


Figure 1 – Graphical representation of the dependencies between three random variables. (a) is the general case and (b) cases under the condition c (Icelandic Volcano eruption) . The edge $x_3 \rightarrow x_2$ is unique to cases under Condition c.

1.2 Naïve testing of a single association

Traditional hypothesis testing methods can be used to check whether a single association is conditioned. In the example above, the cases in which x_2 and x_3 correlate under c , should be found to be significantly different from the cases in the general population.

A new variable can be defined, x_{23} , that receives a true value in cases where $x_2 = x_3$. The hypothesis testing will check whether $p(x_{23}/c) \gg p(x_{23})$. This can be done using a X^2 test [2]; a positive answer may suggest that this association indeed arise uniquely under the condition.

Edwards [3] suggests a stepwise selection method for building a graphical model of the data. Beginning in a fully connected graph, in each step the algorithm excludes edges based on assessment of their significance value resulting from hypothesis testing. Alternatively, edges are added to initially sparse graphs based on their tested significance.

This simple tool becomes inefficient when the number of variables, n , is large. Exploring all the possible associations between any two variables requires $n^2/2$ tests. To test different patterns or subsets of variables, 2^n tests are required, which may prove computationally difficult for n larger than a few dozens.

However, this tool can be used to validate the results of an efficient method that locates conditioned associations. Given the resulting simpler graph, each edge can be tested to verify the significance of the association. Thus, after greatly reducing the search space, the robust hypothesis testing tool can be applied to assess the remaining edges.

1.3 Using unsupervised learning for locating conditioned patterns

Methods of unsupervised learning, such as Clustering [4], can be adapted to locate conditioned associations. In such a case, we compare the clustering results of data under the condition against the data in the general population. Unique clusters that appear in the conditioned population may imply of a conditioned association.

This approach holds a major drawback: if the condition frequency is large enough, or the clustering tree is deep enough, conditioned associations will also appear in the general population clusters. In that case, conditioned associations in the conditioned population may be overlooked.

Another drawback of this approach is that the clusters are mutually exclusive – each variable can only relate to one cluster. This will prevent the discovery of associations between several clusters of variables, which may appear in empirical data.

1.4 Using Bayesian Networks to infer about conditioned patterns

Bayesian Networks (BNT) represent the conditional dependencies between random variables, using a Directed A-cyclic Graph (DAG) [5], and are commonly used in similar problems to ours, mapping the associations between genes [6] and the associations between HIV DNA mutations [7].

The condition variable is usually referred to as a variable with a predefined value. Predefining the value of a variable is commonly termed as intervention or perturbation of the observed data.

Pearl [1] uses the term Atomic Intervention to describe an external manipulation of a network variable value. He suggests [8] adding to the condition variable a parent variable that activates the induced state.

Pe'er et al [9] explored the causal relationships between genes' expression level. In their paper, they note that external perturbations such as medical treatment, that have no direct influence over other variables (in this case gene expression level) but indirectly effect the values of many other variables, should be regarded as "indicator variables", and added to the Network with the constraint that they cannot have other variables as network parents.

Note that the above examples require manipulating or placing constraints over the discovered network structure.

The result of inducing the condition variable to our data will be a joint probability map, containing a node representing the condition. In order to deduce the conditioned associations, one can traverse through the nodes connected to the condition node, or review the whole connected component containing the condition node.

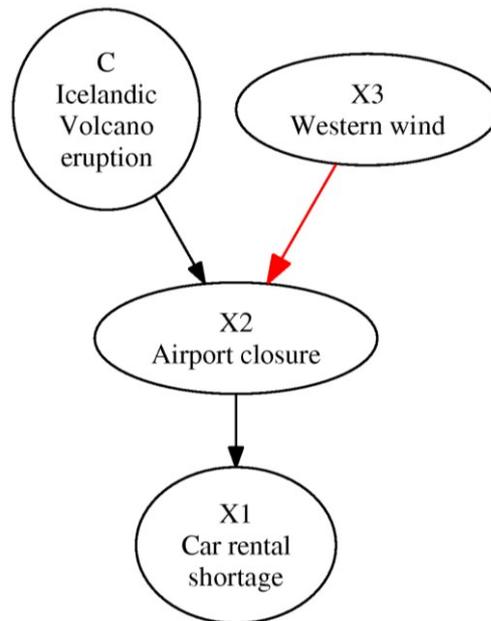


Figure 2 – Bayesian Network representing the joint distribution mapping of X1, X2, X3 and the condition C. The edge $x_3 \rightarrow x_2$ was retrieved using the BNT.

Figure 2 displays the joint probability mapping of our example, represented by a Bayesian Network. It is clear that the new edge, $x_3 \rightarrow x_2$ which appears in the resulting network, suggests that this unique association has been discovered.

However, this case demonstrates the ambiguity encountered when trying to discover conditioned associations using BNT. The $x_3 \rightarrow x_2$ association can result from existing correlation between (x_3) and (x_2) in the general population and is not specific to the conditioned case, in which $c=true$. When this graph is the only source of information, there is no way to distinguish between Background associations (in the general case) and Conditioned associations (in the conditioned case).

This matter could be resolved by exploring the differences between the BNT without the condition variable and the BNT containing the condition variable. Yet in many cases, the proportion of the samples with positive value of the condition variable may be large enough for the conditioned associations to appear in the BNT without the condition variable. In other cases, when the proportion of the samples

with positive condition value is small enough, the conditioned associations will not be discovered in the BNT with the condition variable. In fact, the former ambiguity was clearly visible in the simulation described later in this paper (see paragraph 5e).

Deforche et al. [7] overcomes this issue by preexamining the candidate variables for the BNT, and selecting variables that show significant correlation with the condition. This way the chance that background associations will arise is lowered; conversely, conditioned associations with variables that are not significantly correlated with the condition may be lost. In the above example, since western wind (x_3) is not correlated with the volcanic eruption (c), (x_3) will not be used in our BNT, therefore the $x_3 \rightarrow x_2$ association will be lost.

Another drawback might be caused by the stochastic nature of searching the BNT model and its Directed A-Cyclic Graph (DAG) structure. Reconstructing BNT structure from given observational data requires searching through all possible models, looking for the model that maximizes the log-likelihood of the data. Since this problem is NP-Hard [10], cases with more than a few variables require heuristic search methods [6]. Since prior causal knowledge is missing in many cases, our model search can result in structures that satisfy the observed statistical dependencies, but cannot accurately recover the underlying "real" structure or suggest an equivalent model that may hide the conditioned associations.

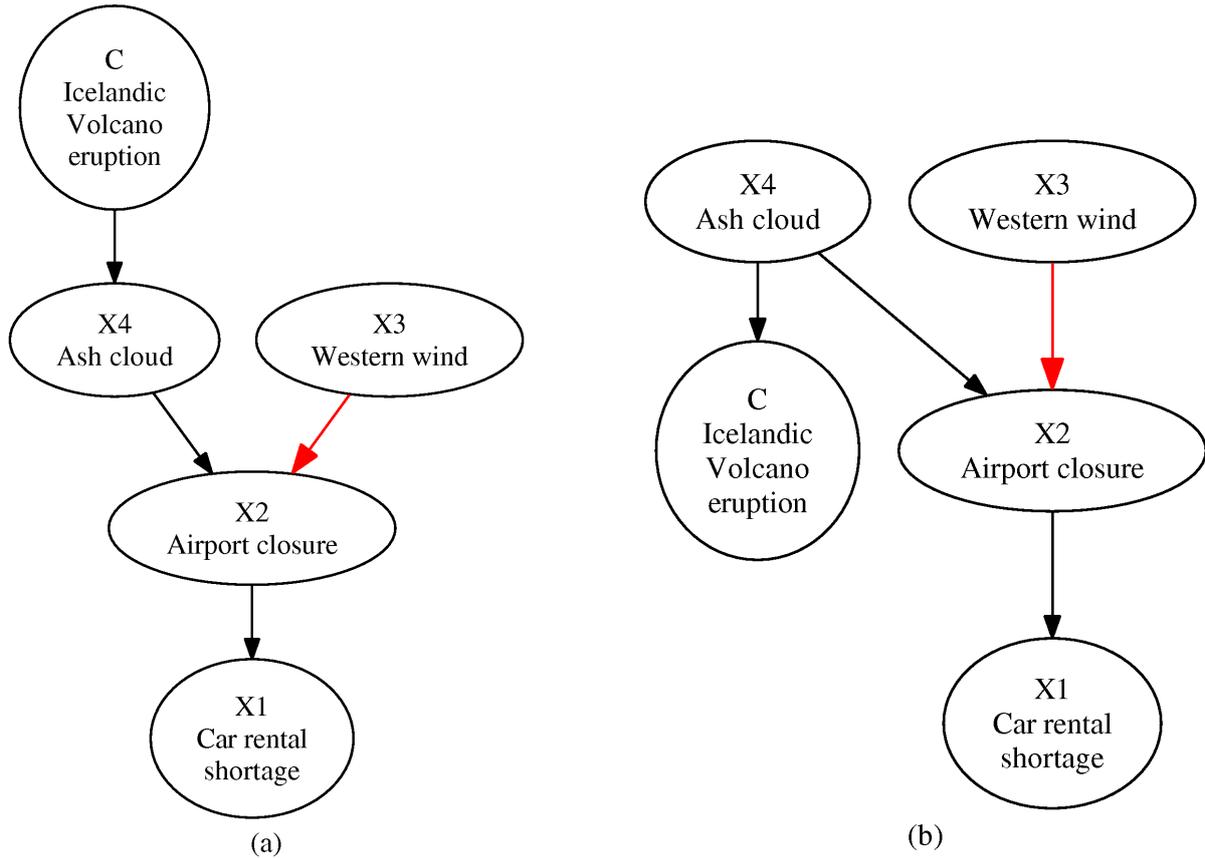


Figure 3 – Alternate Bayesian Network representation of our data. These networks maintain the d -separation criteria between the variables, though now the novel conditioned association ($x_3 \rightarrow x_2$) should be searched within the whole connected component containing C .

Consider the case displayed at figure 3a, where an additional variable is added to our network – Ash cloud (x_4): this DAG can have *observationally equivalent* [1] DAGs (3b) where the statistical dependencies between the variables (herein, d -separation [1]) are still maintained, i.e., (c) and (x_2) are independent given (x_4). In the case displayed in (3b), the causal relationship is obviously wrong.

This example shows the ambiguity faced when trying to draw conclusions concerning the conditioned association, while exploring the graph structure of the BNT. In cases where the network structure is unknown, a single criterion cannot be applied in order to locate the conditioned associations, since we can alternate between two different graph structures that represent a single conditioned association.

This ambiguity effects the search procedure applied in order to locate the novel conditioned associations. To ensure that the conditioned association edges are covered by our search space, the edge directions should be ignored and the edges of the connected component containing the condition should be added to our search space. However, our search space will be greatly obscured by background association added to our result. Applying the methods described above, such as adding an activating variable [8] to the condition variables, or removing the edges from the parents of the condition variable,

while increasing the inference capabilities of the network, still does not eliminate the ambiguity regarding other variables.

1.5 Dependency Networks

Heckerman et al. [11] suggest a method that deals with some of the problems that arise, when interpreting a resultant BNT, especially the tight topology constraints, such as Directed A-cyclic Graph, and the dependencies implied from the d-separation criteria.

This method iteratively regresses, using probabilistic decision trees. Each variable uses the rest of the model variables, and uses the prediction capabilities of the regressor variables as criteria for adding an edge between the regressor variables and the predicted variable. This method retains the "Markov Blanket" property of the BNT – given the parents' values for a network variable; it renders this variable independent of all other variables.

Dependency Networks provide another advantage over other modeling methods – since the computation is done locally over the variable's immediate neighborhood, the computational effort is polynomial over the number of variables, and as such, more efficient.

Meinshausen et al. [12] suggest a method for reconstructing the variables' graphical neighborhood using an estimation of Tibshirani's [13] Lasso method, applying penalty over the regression coefficients norm, and applying a zero value to coefficients with negligible prediction values. While regressing a variable using all the other variables as predictors. Non-zero coefficients will add an edge between the predictor and the predicted variables, that is, enabling a robust and computational efficient method for reconstructing a dependency network, especially in high dimensional cases.

Friedman et al. [14] later enhance this model, suggesting an efficient exact solution to the lasso peer regression problem, using the Banerjee et al. [15] blockwise coordinate descent approach. This method, herein graphical lasso, is demonstrated on cell signaling data from 11 different proteins.

Dependency networks efficiently solve the above problems, providing a straightforward interpretation and usage of a robust numerical mechanism. Yet the ambiguity between the background and conditioned associations still needs to be solved, since the condition variable is still part of the resulting network.

1.6 Summary of current state of the art and the proposed method.

The above sections display the inherent limitations of locating conditioned associations using BNT modeling, mainly the DAG structure that constrains the network structure and observational equivalency that may produce different graphical models for similar variable dependencies. Both limitations can cause the conditioned associations to appear far, or not on the path of the condition variable in the result graph, and so, largely increase the number of associations that need to be tested as candidates for conditioned associations.

Dependency Networks and their recent adaptations solve this problem by addressing the local dependency neighborhood of the variable. Therefore, at least one of the conditioned association vertices should be a neighbor of the condition variable.

Yet, both Bayesian and Dependency networks may identify associations that are independent of the condition (background associations). Since the background associations have the same topological features as conditioned associations, it is hard to discriminate between conditioned and non-conditioned associations, which results in a major increase in the search space.

We introduce a modification of the dependency network method, which obtains a graph containing only conditioned related associations.

We suggest a tool, based on an efficient and numerically robust implementation of ridge regression, which maps the associations of variables under the condition. These properties render it invariant to the proportion of the conditioned samples in the population. The resulting graph identifies the associations between variables that are correlated specifically under the condition variables, and so solving the ambiguity between associations in the general population (background associations) and conditioned associations.

Chapter 2: Method – Iterative ridge regression (IRR)

The previous chapter has demonstrated how current state-of-the-art algorithms, utilize graphical notation for better representation of associations between random variables. Yet, these algorithms hold certain limitations when trying to locate associations between conditioned variables.

One major limitation is their inability to distinguish between background and conditioned associations, which appear in the same result graph.

Bayesian Networks also have DAG structure constraints, which increase the search space needed to locate the conditioned association in the result graph.

Dependency Networks, which solves this constraint by addressing the local dependency neighborhood of the variable, still shares the ambiguity between background and conditioned associations, in the result graph.

We suggest a method that graphically maps conditioned associations between variables. Each association can be illustrated as a directed edge between the variables. The edges can be weighted according to the assumed prediction strength between the variables under the condition. The result will be a network structure (built as a directed Graph) that will map the variable associations under the condition.

Exploring the network structure is done by regressing the condition variable using the remaining variables, observing the weight change of the regressor variables when omitting one, and then again regressing the condition variable by the remaining variables.

We will show that the weight change in the remaining variables is relative to their predictive value over the omitted variable under the condition.

Prediction capabilities of one variable by another, under the condition, can be used as a cue for conditioned association, which can later be tested separately by a simple hypothesis testing tool.

2.1 Problem Definition

Let $S = \{x_1, x_2, \dots, x_n\}$ be a set of random binary variables (features), random binary variable c (Condition), and a data sample set D , where each sample represents a vector for the value of x_1, x_2, \dots, x_n and their corresponding c value. Next, generate a graph (network) $G = \{V, E\}$ where each node corresponds to a variable of $\{x_1, x_2, \dots, x_n\}$; add an edge $x_j \rightarrow x_k$, if the association $\{x_j, x_k\}$ under the condition c is significantly better than in the general population. The direction of the edge is dependent

upon the prediction direction between \mathbf{x}_j and \mathbf{x}_k if \mathbf{x}_j can predict the value of \mathbf{x}_k under the condition c , there will be an edge directed from \mathbf{x}_j to \mathbf{x}_k , and vice versa.

2.2 Selection or regularized regression method

In a classic regression problem, we want to find a weight set W that satisfies the equation:

$$\operatorname{argmin}_W E \left(Y - \sum_{i=1 \rightarrow n} w_i x_i \right)^2 \quad (1)$$

In order to handle the problem of regressing variables with high covariance (typical to the problems explored by the IRR), which cause instability when inverting the sample matrix, a statistical method called Ridge Regression or Tikhonov regularization [16] is utilized. This method adds a penalty over the weight vector norm.

$$\operatorname{argmin}_W E \left(Y - \sum_{i=1 \rightarrow n} w_i x_i \right)^2 + \lambda \|W\|^2 \quad (2)$$

Where the λ parameter stands for the ridge parameter.

Another method, called The Lasso regularization, suggested by Tibshirani [13], replaces the ridge regression's L2 norm penalty of the weight vector, with L1 norm penalty –

$$\lambda \|W\|_{\ell_1}$$

While it also has an efficient solution of the weight vector, it also tends toward a zero value for negligible coefficients when using a large enough λ value.

As was recently suggested, the Elastic nets [17] use a mix of L2 and L1 regularization over the weight norm –

$$\lambda((1 - \alpha) \|W\|^2 + \alpha \|W\|_{\ell_1})$$

where the α parameter value determines whether the penalty leans toward L2 (which displays the ridge regression behavior) or L1 (Lasso).

In their most recent paper, Friedman et al. [18], display a distinction between the above three regularization methods – Ridge regression, Lasso, and elastic nets – in regards to the effect of the weight change of regressor variables after omitting one of them.

Since the L2 norm penalty amplifies an uneven weight distribution, Ridge regression tends to split the coefficient weight of the omitted variable between the highly correlated variables. Lasso, on the

other hand, tends to pick one correlated variable to receive the weight of the omitted variable. Elastic-nets behavior depends on the value of the α parameter.

In our case, the inter-relations between the variables are more relevant than the precision of the model's prediction. Therefore the ridge regression will be more suitable for our needs, while there is a need for a threshold value to filter the negligible interaction.

2.3 Iterative interaction discovery

Once the regularization method is chosen, the conditioned interactions are discovered while iterating over the model variables.

At the first stage, the regression of the condition variable will be done, using the variables in S:

$$W^B = \underset{W}{\operatorname{argmin}} E \left(C - \sum_{x_i \in S} w_i x_i \right)^2 + \lambda \| W \|^2 \quad (3)$$

Later, in each iteration j , the values of variable x_j will be omitted from the samples' data and the condition variable will again be predicted using the remaining variables.

$$W^j = \underset{W}{\operatorname{argmin}} E \left(C - \sum_{x_i \in S/x_j} w_i x_i \right)^2 + \lambda \| W \|^2 \quad (4)$$

This results in a new set of weights for the remaining features – W^j , that will be compared with the original set of weights. The weight difference of each feature can be computed: $\Delta W^j = W^B - W^j$

As proved in subchapter d, ΔW^j is the coefficients vector that gives the best regularized estimation of $W_j^B x_j$.

$$\Delta W^j = \underset{W}{\operatorname{argmin}} E \left(W_j^B x_j - \sum_{x_i \in S/x_j} w_i x_i \right)^2 + \lambda \| W^B - W \|^2 \quad (5)$$

At this point, the benefits of ridge regression come in hand. The best estimation of $W_j^B x_j$ is biased towards the selection of ΔW_j , which evenly distribute the weights between variables that are correlated with x_j under C.

For instance, if \mathbf{x}_1 and \mathbf{x}_2 are fully correlated with \mathbf{x}_j , the result will be $[\Delta W^j]_1 = [\Delta W^j]_2 = \frac{1}{2}$, while in L1 norm penalty, the result may be $[\Delta W_j]_1 = 1$ and $[\Delta W_j]_2 = 0$, or vice versa.

In this case, one can expect all of the variables that highly correlate with \mathbf{x}_j under c , to receive significant value in the corresponding ΔW^j index.

Therefore, each element i in ΔW^j represents the prediction level of \mathbf{x}_j using \mathbf{x}_i under the condition c . This element can be used as a cue for the conditioned association level between \mathbf{x}_i and \mathbf{x}_j .

If $[\Delta W^j]_i$ passes a known threshold t , the association $\mathbf{x}_j \rightarrow \mathbf{x}_k$ can be tested using an independent χ^2 test, whether it is statistically significant under the condition c . The χ^2 test results are adjusted for false discovery rate using the Hochberg-Benjamini correction [19].

Passing both criteria, an edge $\mathbf{x}_j \rightarrow \mathbf{x}_k$ will be added to our graph G .

(See appendix A and B for Pseudo code and demonstration of the method).

2.4 Analytical proof of the IRR method

Given a random variable set $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ and a condition variable c , denote W^B as the basis weight vector of ridge regressing c using S :

$$W^B = \underset{W}{\operatorname{argmin}} E \left(c - \sum_{\mathbf{x}_i \in S} w_i \mathbf{x}_i \right)^2 + \lambda \|W\|^2$$

Let W^j be the resulting weight vector, after the ridge regression of c using S excluding \mathbf{x}_j :

$$W^j = \underset{W}{\operatorname{argmin}} E \left(c - \sum_{\mathbf{x}_i \in S/\mathbf{x}_j} w_i \mathbf{x}_i \right)^2 + \lambda \|W\|^2$$

In the case where $W_j^B \neq 0$, i.e. \mathbf{x}_j has significant part in estimating c , we receive:

$$E \left(c - \sum_{\mathbf{x}_i \in S} W_j^B \mathbf{x}_i \right)^2 + \lambda \|W^B\|^2 < E \left(c - \sum_{\mathbf{x}_i \in S/\mathbf{x}_j} W_j^j \mathbf{x}_i \right)^2 + \lambda \|W^j\|^2$$

Lemma 2.4. Denote ΔW^j to be the weight difference between the two results: $\Delta W^j = W^B - W^j$.

We wish to prove that ΔW^j is the coefficients vector that gives the best estimation of $W_j^B x_j$, while maintaining the regularization expression small:

$$\Delta W^j = \underset{W}{\operatorname{argmin}} E \left(W_j^B x_j - \sum_{x_i \in S/x_j} w_i x_i \right)^2 + \lambda \| W^B - W \|^2$$

Proof. Suppose there is an alternative set of weights: $\widetilde{W}^j \neq \Delta W^j$ that offers better regularized estimation of $W_j^B x_j$ than ΔW^j , hence:

$$\begin{aligned} E \left(W_j^B x_j - \sum_{x_i \in S/x_j} \widetilde{W}_i^j x_i \right)^2 + \lambda \| W^B - \widetilde{W}^j \|^2 \\ < E \left(W_j^B x_j - \sum_{x_i \in S/x_j} \Delta W_i^j x_i \right)^2 + \lambda \| W^B - \Delta W^j \|^2 \end{aligned}$$

Denote $\widetilde{\Delta W}^j$ as their weight difference: $\widetilde{\Delta W}^j = \widetilde{W}^j - \Delta W^j$. After decomposing \widetilde{W}^j :

$$\begin{aligned} E \left(W_j^B x_j - \sum_{x_i \in S/x_j} (\widetilde{\Delta W}_i^j + \Delta W_i^j) x_i \right)^2 + \lambda \| W^B - \widetilde{W}^j \|^2 \\ < E \left(W_j^B x_j - \sum_{x_i \in S/x_j} \Delta W_i^j x_i \right)^2 + \lambda \| W^B - \Delta W^j \|^2 \end{aligned}$$

Hence:

$$\begin{aligned} E \left(W_j^B x_j - \sum_{x_i \in S/x_j} \Delta W_i^j x_i + \sum_{x_i \in S/x_j} \widetilde{\Delta W}_i^j x_i \right)^2 + \lambda \| W^B - \Delta W^j + \widetilde{\Delta W}^j \|^2 \\ < E \left(W_j^B x_j - \sum_{x_i \in S/x_j} \Delta W_i^j x_i \right)^2 + \lambda \| W^B - \Delta W^j \|^2 \end{aligned}$$

Since:

$$E \left(c - \sum_{x_i \in S} W_i^B x_i \right)^2 + \lambda \| W^B \|^2 < E \left(c - \sum_{x_i \in S/x_j} W_i^j x_i \right)^2 + \lambda \| W^j \|^2$$

After decomposing:

$$E \left(c - \sum_{x_i \in S/x_j} W_i^B x_i + W_j^B x_j \right)^2 + \lambda \| W^B \|^2 < E \left(c - \sum_{x_i \in S/x_j} W_i^B x_i + \sum_{x_i \in S/x_j} \Delta W_i^j x_i \right)^2 + \lambda \| W^j \|^2$$

Since we have a better regularized estimation for $W_j^B x_j$:

$$\begin{aligned} E \left(c - \sum_{x_i \in S/x_j} W_i^B x_i + \sum_{x_i \in S/x_j} \Delta W_i^j x_i + \sum_{x_i \in S/x_j} \widetilde{\Delta W}_i^j x_i \right)^2 + \lambda \| W^B - \Delta W^j + \widetilde{\Delta W}^j \|^2 \\ < E \left(c - \sum_{x_i \in S/x_j} W_i^B x_i + \sum_{x_i \in S/x_j} \Delta W_i^j x_i \right)^2 + \lambda \| W^B - \Delta W^j \|^2 \end{aligned}$$

Therefore it can use $\widetilde{\Delta W}^j$ for better minimization of the condition variable estimation without using x_j :

$$E \left(c - \sum_{x_i \in S/x_j} W_i^j x_i + \sum_{x_i \in S/x_j} \widetilde{\Delta W}_i^j x_i \right)^2 + \lambda \| W^j + \widetilde{\Delta W}^j \|^2 < E \left(c - \sum_{x_i \in S/x_j} W_i^j x_i \right)^2 + \lambda \| W^j \|^2$$

And this contradicts with the argmin property of W^j . ■

In the above procedure, we proved that by simple reduction of W^j from W^B we receive the best regularized estimation of the omitted variable by its co-variables.

2.5 Complexity of the IRR algorithm

Since the IRR algorithm uses both Ridge regression and X^2 test, both have many robust and efficient implementations, its calculation time is a linear function over the sample size.

Overall, using an N size variable set and an M size sample set, the IRR regresses the sample set N times, then tests again within the set all of the N variables for significant associations.

In this case the IRR's asymptotic complexity is $O(N^2M)$.

Chapter 3: Comparing the IRR algorithm with Bayesian Networks

The following chapter provides a performance comparison of the IRR algorithm, with state of the art Bayesian network algorithm. Both algorithms try to locate a-priory induced associations between variables that correlate under a condition.

3.1 Data set construction

The sample set contains 1000 samples with 20 variables. Each variable can hold a binary value of -1 or 1.

Each sample is assigned with a binary condition value; the total size of the condition vector is 1000 samples. All sample values were initialized to -1, afterwards 0.2 of the samples had their values inversed to simulate background noise.

Out of the 20 variables, 4 variables were randomly selected to hold the conditioned pattern – their values will be correlated under the condition and random otherwise.

Out of the condition vector, a predefined ratio (0.8) of the samples were set as a condition and their values were set to 1. Out of the conditioned samples, a predefined ratio (0.2) was randomly selected to hold the pattern. These samples will have a positive (1) value in the conditioned pattern variables.

After the construction of the sample set, sampling noise was induced by inverting the values of a predefined ratio (Noise Level) of the sample set.

3.2 Simulation overview

Both the IRR and the BNT algorithms were used to extract the conditioned pattern variables. We used Kevin P. Murphy's BNT Matlab package [20] for BNT construction. The BNT was constructed using the K2 algorithm [21] that outperformed other network construction algorithms [22].

As described in the chapters above, The BNT result DAG, was searched for the Connected Component containing the condition variable. All the variables in this Connected Component (ignoring edge directions) were tested as the result pattern.

The BNT and the IRR result patterns were tested against the induced pattern, and scored using the Jaccard Index. For each noise level, 100 independent simulation executions were performed, and the

mean and standard deviation of the Jaccard score were used as a basis for performance comparison between the IRR and the BNT (see exact simulation details in appendix C).

3.3 Results

Noise level	IRR	K2
0	0.9340	0.7220
0.02	0.8980	0.6327
0.04	0.9163	0.6750
0.06	0.9180	0.7153
0.08	0.8828	0.6020
0.1	0.8923	0.6173
0.12	0.9012	0.6504
0.14	0.8593	0.5330
0.16	0.8418	0.5280
0.18	0.8160	0.6558
0.2	0.8001	0.3687
0.22	0.7030	0.3755
0.24	0.6382	0.3135
0.26	0.5423	0.2052
0.28	0.4298	0.2015
0.3	0.3153	0.1360
0.32	0.2223	0.0965
0.34	0.1480	0.0492
0.36	0.1093	0.0827
0.38	0.0470	0.0329
0.4	0.0295	0.0170

Table 1: Mean Jaccard index scoring over 100 iterations of the IRR and Bayesian Network K2 algorithms along various noise levels

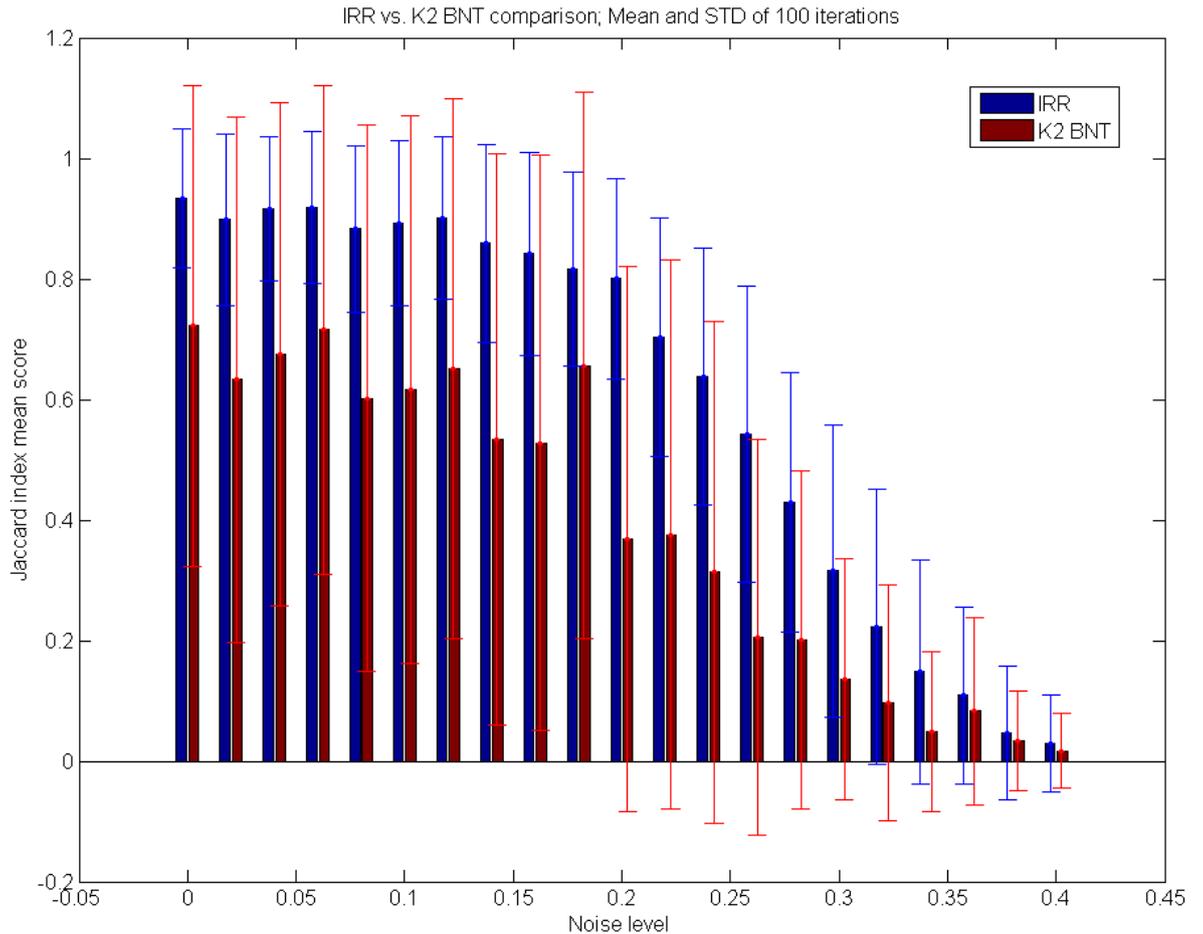


Figure 4: Graphic representation of the Jaccard index scoring mean and STD of the IRR and Bayesian Network K2 algorithms, over 100 iterations. The IRR algorithm significantly outperforms the Bayesian Network K2 algorithm ($p < 0.01$). The STD values show more stable performance by the IRR as compared with the K2 algorithm.

3. 4 Summary of the comparison results

The simulation results show that the IRR performs well when trying to locate conditioned patterns in data that have up to 25% noisy (i.e. reversed) values. At these noise levels, the IRR significantly ($p < 0.01$) outperforms known state of the art algorithms such as the K2 BNT algorithm.

These results emphasize the effect of association ambiguity of the BNT, as shown in the following simulation case:

In this case, a 4 sized pattern that includes variables number 1, 16, 17 and 20, was induced to a 20 random variables data set. These variables associate only under the condition; otherwise, they are randomly correlated.

Figure 5 shows that the IRR result is a fully connected graph, which includes the pattern variables. The BNT graph, which by nature is a DAG (directed a-cyclic graph), contains the pattern variables but also two other variables (var12 and var15) that were randomly associated with the pattern variables, regardless of the condition.

As described before, we can only assume about the conditional nature of the BNT result by exploring the connected component of the graph containing the condition, and ignoring the direction of the edges, thus include variables 12 and 15 in the suggested pattern result.

In this case the Jaccard index score of the IRR graph will be 1 (fully compatible with the pattern), while the BNT result will be 0.67.

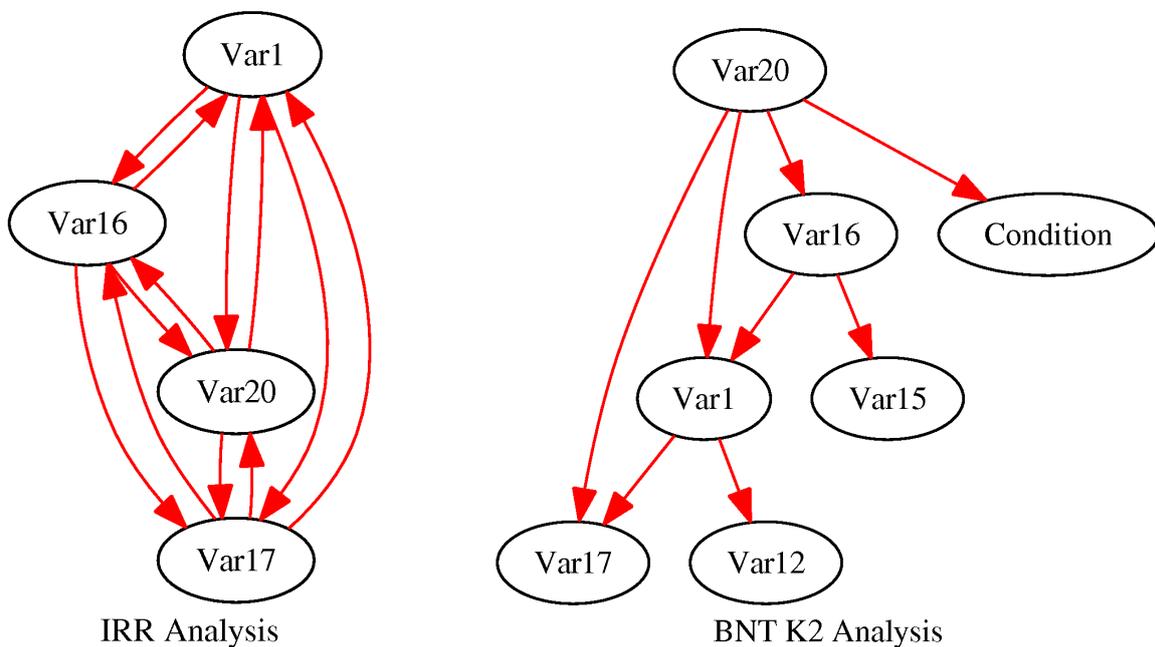


Figure 5: Graphical result of pattern locating. The pattern contains 4 variables (Var1, Var16, Var17, Var20) in a 20 random variables data set. The IRR has fully located the induced pattern, while the BNT has also located two randomly correlated variables (Var12, Var15), not associated in the pattern under the condition.

3.5 Conclusions

The simulation above shows that the IRR algorithm performs well in various noise levels, when trying to locate conditioned associations. The IRR significantly ($p < 0.01$) outperforms a state-of-the-art Bayesian Network algorithm, yields better accuracy scoring (measured using Jaccard Index) and smaller variance of the results. The major cause of performance degradation of the BNT, is association

ambiguity, where background associations correlate with conditioned associations. Since the BNT does not differentiate between the two, it harms the specificity scoring of the BNT results.

Chapter 4: Application of the IRR for prediction of HIV resistance mutation patterns

One possible usage of the IRR is to explore patterns of HIV DNA mutations that appear after treatment admittance; in fact, this was the trigger for the IRR development.

The susceptibility of the HIV RNA to natural variation increases the probability that with time, RNA variants will appear. These variants might translate into a virus protein structure immuned to the admitted treatment. These DNA variants carry changes, termed as **resistance mutations**, and are a major cause for treatment failure in HIV patients.

In most cases, the treatment is targeted to interfere with the activity of specific HIV proteins, such as viral protease, reverse-transcriptase and integrase. The mutation mapping is done in parts of these protein sequences, which translate into the protein active sites. These are the sites that are affected by the HIV treatment. Mapping the resistance mutations associations can be beneficial in the selection and optimization of a given treatment. It can be used to predict the appearance of resistance mutation or to explore the functional behavior of the mutant protein.

Deforche et al. [7], have explored the resistance mutations interaction using a Bayesian Networks modeling. Their data set contained amino-acid samples of the protease site from HIV patients and their treatment history (specifically – Nelfinavir or NFV, a Protease Inhibitor (PI)). At the first stage, a set of mutations was chosen, using a X^2 test, to select the mutations that significantly correlate with the treatment.

At the second stage, a new data set was created, where each variable was assigned with a Boolean value associated with a specific mutation appearance. An additional variable was assigned with the NFV treatment admittance of this sample.

BNT modeling was used to explore the associations between the resistance mutations variables and between the resistance mutations NFV treatment variable and the resistance mutations variables.

The result network has recovered some of the known interactions between the NFV mutations. It also points to strong associations later found to be biologically correlated.

4.1 Data and Method

A total of 1745 sample protease DNA Sequences were obtained from the National HIV Reference Laboratory (NHRL). 1261 samples were of subtype C, of which 170 were treated with NFV, and 454 samples were of subtype B, of which 29 were treated with NFV.

DNA sub typing was done using the Stanford HIV reference center Database (<http://hivdb.stanford.edu/>) [23,24].

DNA sequences were aligned using the ClustalW [25] program, and then translated to the matching Amino-Acid sequences.

Since the C and B subtypes can display different resistance mutation patterns [26,27,28] , we separated the IRR analysis to each subtype's samples.

The amino acid sequences were compared to the subtype consensus sequences. The subtype consensus sequences were equal in naïve and in the general population. After the consensus comparison, a mutations variable set was established by including every diversion from consensus, which occurred at least twice

For each mutation, a data set binary variable was created, stating whether the mutation has appeared in this sample (positive value) or not. The condition variable value holds the NFV treatment status for each sample.

The binary frequency input matrix (non positive values were set to -1) of the mutation variables and the condition variable were used as an input for the IRR algorithm.

The IRR algorithm setup used a value of 50 as the ridge parameter, and a $p < 0.05$ as the X^2 test significance threshold.

4.2 Results

Graphical representation of the IRR analysis of the HIV mutation data are displayed in figures 6a (subtype B analysis) and 6b (subtype C analysis). Associations that showed exceptionally significant X^2 scoring ($p < 0.00000$) and IRR scoring > 0.01 are marked in red bold edges.

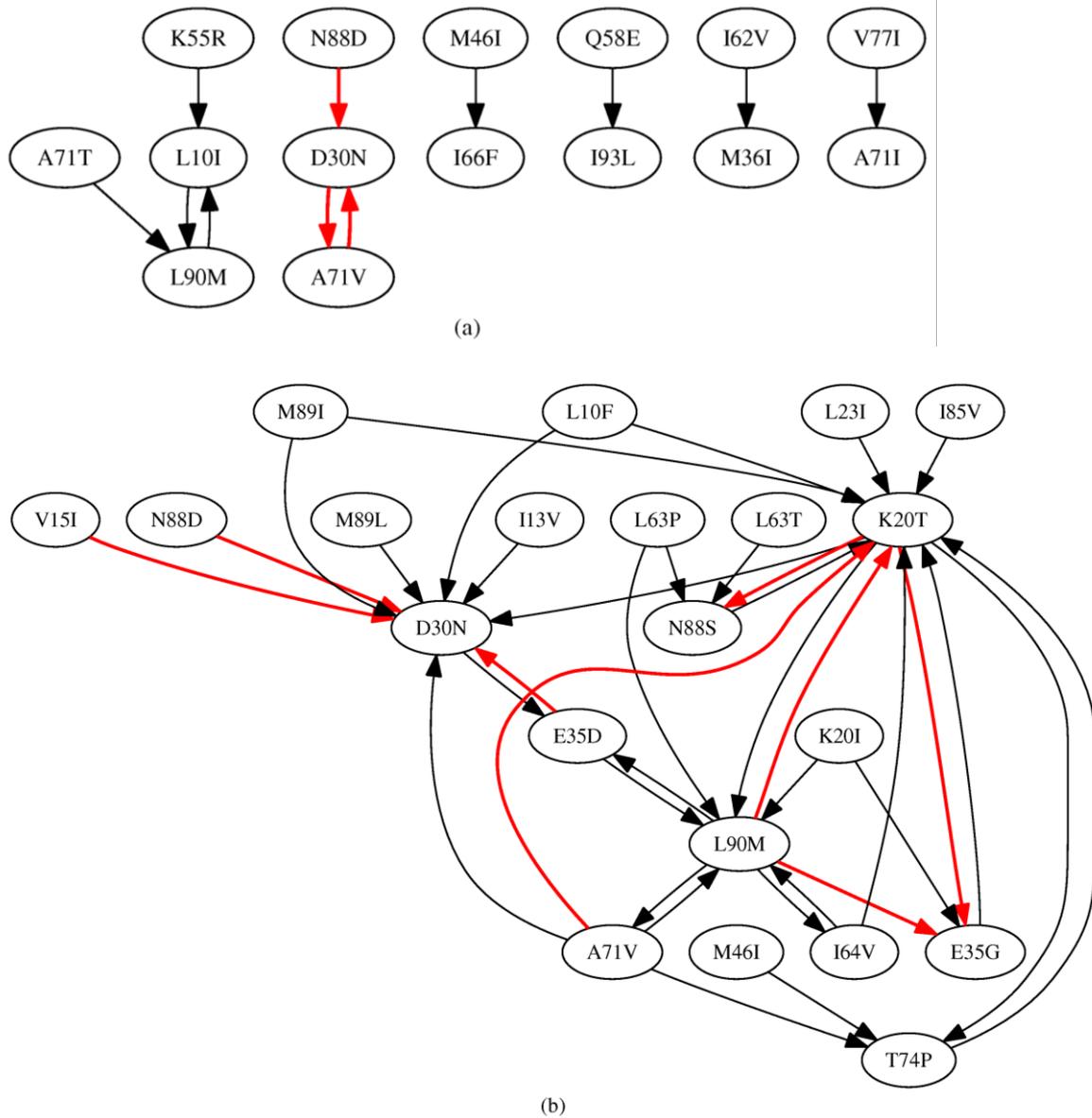


Figure 6: IRR graphical result of amino acid mutations associations found in subtype B (figure 6a) and subtype C (figure 6b) HIV protease sequences, conditioned by Nelfinavir (NFV) treatment. Red edges suggest exceptionally significant associations with high IRR and X^2 scoring. Edge directions are such that the presence of the edge source will increase the probability of the edge target.

4.3 Comparison to previous findings

A comparison to Stanford HIV reference center Database [23] shows that out of the 31 mutations in the IRR result networks, 13 have been previously discovered as resistance mutations (L10F, L23I , D30N, E35G, M46I, Q58E, A71V/T/I, T74P, N88D, N88S, L90M). The IRR network contains directed paths that match known resistance mutation patterns (D30N-N88D, M46I-L90M, M46I-N88S, D30N-N88D-L90M, N88S-L90M, M46I-N88S-L90M, D30N-M46I-N88D-L90M).

Compared to Deforche et al.'s Bayesian network analysis [7], the IRR network shared 24 mutations with the Bayesian network. It also recovered the different pathways of N88D (directly connected to D30N) and N88S. It also reiterated the M89L-D30N path, which is specific to subtype C samples.

4.4 Analysis of inconsistencies with previous findings

The results above display some inconsistencies with previous publications. An in depth statistical analysis of these cases in the sample set shows that in most cases the correlation between the mutation and the treatment was not significant. Such cases are mutations L33F, I54V, I84V and V82A, which appear in the Stanford HIV DB [23]. Many of the mutation patterns have low correlation with NFV treatment (V82A has even small negative correlation with NFV treatment). This, together with their low frequency in the treated samples, makes them hard to discover via pure statistical means (see appendix 7e for their distribution in the sample set).

Another inconsistency is the undirected path from mutation D30N and L90M that are known *not* to appear together at the same case [26,29,30]. In this case, their network neighbor, E35D, is highly correlated separately with D30N and L90M, under treatment, this although there are 0 cases where D30N and L90M appear together. This appears to be an artifact of the locality property of dependency networks. In subtype B samples, E35D does not appear as an NFV related mutation; therefore, in the subtype B networks D30N and L90M appear separately in different connected components.

4.5 IRR new findings

In subtype C IRR network, mutation V15I appears with strong conditioned association to D30N resistance mutation. An analysis of subtype B patients samples shows that 15I appears as a wild type form (the consensus DNA sequence), as opposed to subtype C where it appears as a mutation. As was previously shown, 83.8% of patients with viral rebound on NFV, who had resistance mutations, had D30N [31]. It was also shown that the prevalence of D30N in subtypes C or A under NFV treatment is significantly lower than in subtype B under this treatment [4,29]. Our work may provide an explanation

for this phenomenon. We assume that the appearance of D30N requires the support of two amino acids: 15I and 89L. Both mutations appear as the wild-type B virus. In subtype C, both mutations have to appear in order to support and stabilize the appearance of D30N. In the BNT analysis, Deforche et al. already found the 89L-D30N correlation. Our method strengthens this finding and adds 15I-D30N as an additional NFV-related association.

Mutation K20T has highly significant correlation with the NFV treatment in the subtype C sample set. The network analysis shows its high association ratio with other resistance mutations under NFV treatment. This property, known as Network Hub, indicates a possible significant role of this mutation in many resistance pathways. Since it also appears in Deforche et al.'s analysis, it suggests that a phenotypic survey of this mutation (along with its associate's mutations) can indicate a significant biological role of this mutation in resistance mechanism.

Chapter 5: Summary and Conclusions

This thesis displayed the problem of locating conditioned association. Where we want to recover specific type of associations between variables, which occur in the presence of another variable – the Condition.

Current state-of-the-art algorithms, that graphically map the joint probability of random variables, lack the ability to differentiate between the conditioned and non-conditioned associations. Some algorithms, such as Bayesian Networks, apply structural constraints over the result network which may cause the conditioned association to appear distant from the condition variable, thus largely increase the search space for these associations.

The thesis introduced IRR, a modification of the dependency network method, which produces a directed graph containing only condition related associations. We have described a robust and computationally efficient estimator algorithm used to uncover such a network.

We showed that the IRR algorithm performs better than current state of the art Bayesian networks when the purpose is to identify conditioned associations.

We also describe, here and in a forthcoming publication [32,33], a real life application of the IRR method to the analysis of HIV mutation patterns that emerge in HIV infected patients treated with a particular antiretroviral drug. We have demonstrated that our method can recover from a pool of viral sequences known associations of such treatment with selected resistance mutations as well as documented associations between different mutations. Moreover, the IRR method revealed novel associations that are statistically significant and biologically plausible.

The IRR can be utilized as a simple and straight forward framework for locating conditioned association in various empirical studies, such as molecular biology, physiology and earth studies.

Appendix

Appendix A

IRR Pseudo code

A pseudo-code for describing the iterative ridge regression (IRR) construction of variable correlation under condition. See text for details of the algorithm steps.

```
/* variables initialization*/
```

```
K = number of variables
```

```
N = number of samples
```

```
Variable List = 1..K variables
```

```
Sequence Samples = 1..N sequences
```

```
Condition Vector = 1..N boolean (index i is true is sample i was treated)
```

```
Variable interaction matrix = KxK real
```

```
Variable Matrix = NxK matrix
```

```
/* build the variables frequency matrix */
```

```
Foreach Sequence Sample i = 1..N
```

```
  foreach Variable j = 1..K
```

```
    if Sequence Sample i contains Variable j
```

```
      Variable Matrix(i,j) = 1
```

```
    else
```

```
      Variable Matrix(i,j) = -1
```

```
    endif
```

```
  endforeach
```

```
endforeach
```

```
/* build the Variables interaction matrix */
```

```
Base Weight Vector = apply Ridge Regression to estimate Condition Vector with
```

```
Variable matrix as data;
```

```

for i = 1..K
  Omitted VariableMatrix = NxK-1
  Omitted Variable Matrix(1..N,1..i-1) = Variable Matrix(1..N,1..i-1)
  Omitted Variable Matrix(1..N,i..k-1) = Variable Matrix(1..N,i+1..k)
  Omitted Weight Vector = apply Ridge Regression to estimate Condition
  Vector with Omitted Variable matrix as data;
  Weight Diff Vector = Omitted Weight Vector - Base Weight Vector
  Variables interaction matrix(1..k,i) = Weight Diff Vector
end

threshold = get a p value threshold from the distribution of all interactions

/* network buildup */

Variables Interaction Graph = Create empty Graph matrices with variables as vertices
foreach i = 1..K , j =1..K
  if Variables interaction matrix(i,j) > threshold and
    Variable I and j is significant (FDR corrected) under condition
    add directed arc from variable i to j
  endif
endforeach

return Variables Interaction Graph

```

Appendix B

Detailed Algorithm demonstration –

Follows a simulated example of finding a conditioned association. The left column shows the condition variable value, applied to each of the samples, while the columns to its right show the existence of several mutations in each of the samples.

Condition	Var. D	Var. C	Var. B	Var. A
0	1	1	0	1
0	0	0	1	0
0	1	1	1	1
0	0	0	0	1
0	1	1	1	0
0	0	0	0	0
1	1	1	0	0
1	0	0	1	1
1	0	0	1	1
1	1	1	1	1
1	0	0	1	1
1	1	1	0	0

Table 2: Example data set of four random variables (Variables A, B, C, D) and a condition variable, over 12 samples. Positive sample value is marked with gray.

At first stage, the 0 values are replaced with -1 for more robust calculations.

Variables C and D are fully correlated, but with low correlation with the condition (-0.169). In the full sample set, variables A and B have low correlation between themselves (0.314) and with the Condition (0.169). Yet it is clearly visible that under the condition, variables A and B are highly correlated, hence creating a pattern. When we omit Variable D we will receive the following weights values after ridge regression:

Variable	Ridge Reg. Weights	Weights After Omission of Variable D	Weight Difference
A	0.0302	0.0302	0.0000
B	0.0302	0.0302	0.0000
C	0.0016	0.0018	0.0002
D	0.0016	N.A.	N.A.

Table 3: Weights of the variables after ridge regressing the condition variable, and after ridge regressing the condition variable without variable D.

The change in the weights of the other variables is relatively small, and not affected by the initial weight of the feature.

However, after omitting one of the pattern's members we will receive the following weights:

Variable	Ridge Reg. Weights	Weights After Omission of Variable A	Weight Difference
A	0.0302	N.A.	N.A.
B	0.0302	0.0319	0.0017
C	0.0016	0.0008	-0.0008
D	0.0016	0.0008	-0.0008

Table 4 : Weights of the variables after ridge regressing the condition variable, and after ridge regressing the condition variable without variable A.

The weight change in the other pattern member (Variable B) is significantly bigger than the change in the other weights, and positively directed.

Iteratively applying the same step over the whole set of variables results in an NxN table consisting of the weight change of each of the features when a different feature omitted.

Ridge Regression Weights	Weight Diff				
All Variables	Variable	Var. A omitted	Var. B omitted	Var. C omitted	Var. D omitted
0.0302	A	0	0.0016677	0	0
0.0302	B	0.00167	0	0	0
0.0016	C	0	0	0	0.000279
0.0016	D	0	0	0.0002792	0

Table 5: Weights of the variables after ridge regressing the condition variable after omitting each of the variables. IRR Significant associations are marked in gray.

Pattern validation using χ^2 test: after a filtered interaction table, each cell holds a value that describes an “IRR significant” directed interaction between two features (mutations). These filtered interactions will comprise our Region of Interest (ROI), which will be the basis for an independent statistical test – this test will filter out the significant interaction from the previous “IRR significant” interaction.

For each $a \rightarrow b$ interaction, we will test whether the $a \rightarrow b$ association frequency under the condition is significantly different from its frequency in the general sample's population – i.e. $p(a, b/c) \gg p(a, b)$. Resulting P values will be adjusted according to the Benjamini-Hochberg algorithm for P value adjustment. We will sort the interactions that pass our threshold P value.

Network construction – we will now construct an interaction network out of the filtered interactions, each mutation will become a vertex, while every filtered $a \rightarrow b$ interaction will add a directed edge between the a and b vertices. Since the significant interactions found are between variables a and b the result network will be: $a \leftrightarrow b$.

Appendix C

Exact details of the IRR – BNT simulation

i) Data Simulation

(1) Parameters –

- (a) Number of Features – **N**
- (b) Number of Samples – **M**
- (c) **Condition ratio out of samples** – ratio of samples that are conditioned.
- (d) Pattern description –
 - (i) **Pattern size** – number of features in the pattern.
 - (ii) **Pattern-Condition ratio** – the ratio of the conditioned samples which contains the pattern.
- (e) **Background noise level** – ratio of random data members which have positive value (before pattern induction).
- (f) **Sampling noise level** – ratio of random data members, which will have their value inverted.

(2) Data creation

- (a) Allocate an **N** by **M** data matrix (Samples).
- (b) Allocate an **M** size vector (Condition vector).
- (c) Reset the sample matrix and the condition vector values to -1.
- (d) Randomly select cells out of the sample's matrix, until selecting total cells ratio of **background noise level**. Inverse their value to 1.
- (e) Randomly select cells out of the condition vector, until selecting an amount of cells corresponding to the **Condition ratio** multiplied by the samples set size. These are the **condition indices**, Inverse their value to 1.
- (f) Pattern induction –
 - (i) Randomly select total of **pattern size** indices out of the features indices. These are the **Pattern indices**.
 - (ii) Randomly select samples, until reaching total of (**Pattern-Condition ratio** * size of **condition indices**) , which their corresponding indices in the Condition vector are positive. These are the **Conditioned samples** associated with the condition vector.
 - (iii) Change the values of the **Conditioned samples** in the **Pattern indices** to 1.

(g) Sampling noise induction –

- (i) Randomly select values until reaching a ratio of sampling **noise level** out of the samples values, and inverse their values.

ii) Single Simulation execution

(1) IRR execution and scoring

- (a) Use the sample matrix and the condition vector as an input to the IRR algorithm.
- (b) IRR result is the estimated conditioned pattern; use Jaccard Index to evaluate the similarity between the original induced pattern and the estimated IRR result pattern.

(2) Bayesian network (**BNT**) execution and scoring

- (a) We used the BNT package [20] for executing a Bayesian network analysis over the simulated data.
- (b) We compared the IRR with the K2 Bayesian network algorithms [21], showed to outperform or match other BNET algorithms [22]
- (c) Input for the BN was created by concatenating the sample matrix and the condition vector – the condition vector becomes another feature in the sample features.
- (d) We used the DAG result of the Bayesian Network Power Constructor (BNPC) algorithm [5], for topological sorting used to deduct the variable order, needed as an input for the K2 algorithm.
- (e) The scoring was done by searching the connected component of the DAG result that contains the Condition feature.
- (f) We assemble a variable set containing the variables in the connected component.
- (g) The set is regarded as the result estimated pattern of the algorithm.
- (h) We use Jaccard Index to evaluate the similarity between the original induced pattern and the estimated Bayesian Network result pattern.

iii) Comparison method and parameters

(1) In each comparison iteration, we simulated an input data matrix with parameters values as follows –

- (a) Number of Features – $\mathbf{N} = 20$
- (b) Number of Samples – $\mathbf{M} = 1000$
- (c) Condition ratio out of samples = 0.8
- (d) Single Pattern induced–
 - (i) Pattern size = 4
 - (ii) Pattern-Condition ratio = 0.2.
- (e) Background noise level = 0.2

- (f) Sampling **noise level** = variable across the simulation
- (2) We used the simulated data as an input to the IRR, BNPC and K2 algorithms.
- (3) For each set of parameters, we used 100 executions to get an average score on each sampling noise level.

Bibliography

- [Judea Pearl, *Causality: Models, Reasoning, and Inference*. London: Cambridge University
1] Press, 2009.
- [M.S. Nikulin, "Chi-square test for continuous distributions with shift and scale parameters,"
2] *Theory of Probability and Its Applications*, vol. 18, pp. 559–568, 1973.
- [D. M. Edwards, *Introduction to Graphical Modelling, 2nd ed.* New York: Springer, 2000.
3]
- [A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A Review," *Computing Surveys*,
4] vol. 31, pp. 264–323, 1999.
- [G. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic
5] networks from data," *Machine Learning*, no. 9, pp. 309-347, 1992.
- [N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian Networks to Analyze
6] Expression Data," *Journal of Computational Biology*, vol. 7, pp. 601-620, 2000.
- [K. Deforche et al., "Analysis of HIV-1 pol sequences using Bayesian Networks:
7] implications for drug resistance," *Bioinformatics*, vol. 22, pp. 2975-2979, 2006.
- [J. Pearl, "Comment: Graphical Models, Causality and Intervention," *Statistical Science*, vol.
8] 8, pp. 266-269, 1993.
- [D. Pe'er, A. Regev, G. Elidan, and N. Friedman, "Inferring subnetworks from perturbed
9] expression profiles," *Bioinformatics*, vol. 17, pp. S215-S224, 2001.
- [D. M. Chickering, D. Heckerman, and C. Meek, "Large-Sample Learning of Bayesian
10] Networks is NP-Hard," *Journal of Machine Learning Research*, vol. 5, pp. 1287-1330, 2004.
- [David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, and
11] Carl Kadie, "Dependency Networks for Inference, Collaborative Filtering, and Data
Visualization," *Journal of Machine Learning Research*, vol. 1, pp. 49-75, 2000.
- [N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the
12] Lasso," *Annals of Statistics*, vol. 34, pp. 1436-1462, 2006.
- [Robert Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal
13] Statistical Society. Series B*, vol. 58, pp. 267-288, 1996.
- [J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the

- 14] graphical lasso," *Biostatistics*, vol. 9, pp. 432-441, 2008.
- [Onureena Banerjee, LE Ghaoui, and Alexandre d'Aspremont, "Model Selection Through
- 15] Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data," *Journal of Machine Learning Research*, vol. 9, 2008.
- [A. N. Tychonoff and V. Y. Arsenin, "Solution of Ill-posed Problems," Washington, 1977.
- 16]
- [Hui Zou and Trevor Hastie, "Regularization and variable selection via the elastic net,"
- 17] *Journal of the Royal Statistical Society: Series B*, vol. 67, no. 2, pp. 301-320, 2005.
- [J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear
- 18] models via coordinate descent," *Journal of Statistical Software*, vol. 33, pp. 1–22, 2010.
- [Yoav Benjamini and Yosef Hochberg, "Controlling the False Discovery Rate: A Practical
- 19] and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society. Series B*, vol. 57, pp. 289-300, 1995.
- [Kevin Murphy. (2001) Bayes Net Toolbox for Matlab. [Online].
- 20] <http://people.cs.ubc.ca/~murphyk/Software/BNT/bnt.html>
- [J. Cheng, D.A. Bell, and W Liu, "An algorithm for Bayesian belief network construction
- 21] from data.," *Proceedings of AI and STAT'97*, pp. 83–90, 1997.
- [P. Leray and O. Francois, "BNT structure learning package: documentation and
- 22] experiments," 2004.
- [Robert W. Shafer, "Rationale and Uses of a Public HIV Drug-Resistance Database," *Journal*
- 23] *of Infectious Diseases*, vol. 194, pp. s51-s58, 2006.
- [Soo Yon Rhee et al., "Human immunodeficiency virus reverse transcriptase and protease
- 24] sequence database.," *Nucleic Acids Research*, vol. 31, no. 1, pp. 298-303, 2003.
- [M. A. Larkin et al., "Clustal W and Clustal X version 2.0," *Bioinformatics*, vol. 23, no. 21,
- 25] pp. 2947-2948, 2007.
- [Z. Grossman et al., "Mutation D30N is not preferentially selected by human
- 26] immunodeficiency virus type 1 subtype C in the development of resistance to nelfinavir," *Antimicrobial agents and chemotherapy*, vol. 48, pp. 2159–2165, 2004.
- [R. Kantor et al., "Impact of HIV-1 Subtype and Antiretroviral Therapy on Protease and
- 27] Reverse Transcriptase Genotype: Results of a Global Collaboration," *PLoS Medicine*, vol. 2, pp. 325-337, 2005.
- [S.Y. Rhee et al., "HIV-1 pol mutation frequency by subtype and treatment experience:

- 28] extension of the HIVseq program to seven non-B subtypes," vol. 20, no. 5, pp. 643-652, 2006.
- [K. Ariyoshi et al., "Patterns of Point Mutations Associated With Antiretroviral Drug
29] Treatment Failure in CRF01_AE (Subtype E) Infection Differ From Subtype B Infection,"
Journal of Acquired Immune Deficiency Syndromes, vol. 33, no. 3, pp. 336-342, 2003.
- [W. Sugiura et al., "Interference between D30N and L90M in Selection and Development of
30] Protease Inhibitor-Resistant Human Immunodeficiency Virus Type 1," *Antimicrobial Agents
and Chemotherapy*, vol. 46, pp. 708-715, 2002.
- [B. Clotet et al., "Prevalence of HIV Protease Mutations on Failure of Nelfinavir-Containing
31] HAART: A Retrospective Analysis of Four Clinical Studies and Two Observational Cohorts,"
HIV Clinical Trials, vol. 3, pp. 316-323, 2002.
- [Nimrod Bar-Yaakov, Nathan Intrator, and Zehava Grossman, "Interactions Among PI-
32] Induced Resistance Mutations Revealed by Iterative Ridge Regression ," *preprint*, 2011.
- [Nimrod Bar-Yaakov, Zehava Grossman, and Nathan Intrator, "Using Iterative Ridge
33] Regression to Explore Associations Between Conditioned Variables," *Journal of Computational
Biology*, 2011, submitted.
- [T. Sing et al., "Characterization of novel HIV drug resistance mutations using clustering,
34] multidimensional scaling and SVM-based feature ranking," in *Knowledge Discovery in
Databases: PKDD 2005*. New York: Springer, 2005, pp. 285:296.
- [J. Pearl, "Fusion, propagation, and structuring in belief networks," *Artificial Intelligence*,
35] vol. 29, pp. 241–288, 1986.
- [Florian Markowetz and Rainer Spang, "Inferring cellular networks – a review," *BMC
36] Bioinformatics*, vol. 8, p. S5, 2007.
- [FV Jensen, *An Introduction to Bayesian Networks*. Berlin: Springer, 1996.
37]
- [AE Hoerl, "Application of ridge analysis to regression problems," *Chemical Engineering
38] Progress*, no. 58, pp. 54-59, 1962.
-

תקציר

בתזה זו זה נתייחס לבעיה ייחודית של מיפוי הסתברויות משותפות, כאשר המידע הנדרש הינו הקשרים ההסתברותיים בין משתנים מקריים שונים, המתרחשים תחת משתנה מקרי אחר – התנאי. קשרים אלו יכוונו - קשרים מותנים. רשתות בייסיאניות ורשת תלויות (Dependency Networks) ממפות בצורה גרפית את הקשרים בין משתנים מקריים. אך רשתות אלה ימפו גם קשרים בין משתנים שאינם תלויים בתנאי. מכיוון שמבנה הרשת אינו מאפשר להבדיל בין סוגי הקשרים השונים, נאלץ לחפש בכל הרשת אחר הקשרים המותנים, דבר שיגדיל את מרחב החיפוש בצורה משמעותית. אנו נציג שיטה המבוססת על רשת תלויות, המייצרת גרף מכוון המכיל רק קשרים התלויים בתנאי. צמתי הגרף ייצגו את המשתנים המקריים והקשתות את הקשרים הנוצרים תחת התנאי. שיטה זו מבוססת על רגסיית Ridge, המאפשרת לנו להשתמש במימושים יעילים ויציבים הקיימים לאלגוריתם זה. נדגים את יעילות שיטה זו לתחום הרפואי, לאיתור מוטציות DNA עמידות לטיפול תרופתי, בחולי איידס המטופלים בתרופות אלה. השתמשנו בשיטה זו בכדי למפות קשרים בין מוטציות שהותנו במתן טיפול תרופתי מסויים. שיטה זו מגלה קשרים ידועים בין מוטציות עמידות לבין טיפול תרופתי מסויים, וכן קשרים ידועים בין המוטציות לבין עצמן תחת טיפול זה. בנוסף, שיטה זו זיהתה קשרים חדשים בין מוטציות, שהינם מובהקים סטטיסטית ובעלי משמעות ביולוגית.