A Comparative Study on Feature Selection Methods and Their Applications in Causal Inference

Yubin Kuang

26th May 2009

Supervisor: Dr. Jacek Malec

Examensarbete för 30 hp
Institutionen för datavetenskap, Naturvetenskapliga fakulteten, Lunds universitet

Thesis for a diploma in Computer Science, 30 ECTS credits
Department of Computer Science, Faculty of Science, Lund University

Abstract

Feature selection is of crucial importance in model building process to eliminate variables that are not informative. Proper feature selection facilitates the model building by reducing noise and the dimensionality of the problem. However, the lack of correspondences between the features selected by two different feature selection methods raises the question of how to evaluate feature selection methods. Generally, for a given data set, feature selection methods are compared based on the prediction performance. Nevertheless, there are feature selection methods that output distinct subsets of features with statistically equivalent prediction performance. Incorporating causation into the diagram of feature selection presents a candidate to clarify such ambiguity.

The purpose of this report is to study different feature selection algorithms and their potential application in causal inference. This report first reviews methodologies in designing feature selection algorithms and their implications to causal inference. Then the performance of several widely used feature selection methods are studied on both simulated data and real data. Several modifications for the backward search algorithm by Nilsson et al. (2007) are also studied to see whether they will improve the performance.

Acknowledgements

First of all, I would like thank my supervisor Dr. Jacek Malec, who have read my thesis carefully and gave me helpful opinions from the side of machine learning. A special thank you goes to Prof. Krzysztof Nowicki, who had several inspiring and long discussions with me on the early version of the thesis. His suggestions on scientific writing and statistical subjects benefit me a lot. I also thank Dr. Ferenc Belik for arranging the practical details for my presentation.

I would also like to thank Doc. Rolf Karlsson, Doc. Thore Husfeldt and Prof. Andrzej Lingas for introducing me to the field of algorithms, and Prof. Johan Helsing for his guidance in numerical linear algebra, and Prof. Björn Holmquist for his help on my pursuit of studying statistics.

Further, I am grateful to my parents for continuous support on my pursuit of study and Xuan for always being understanding. I also thank my friends Yishi, Gan, Wang, Xiaohan, Hui for their helpful discussions and other things.

Contents

Chapter 1

Introduction

Feature selection is studied intensively in the theoretical field such as machine learning for its vast applications in gene expression microarray analysis, image analysis and text processing. Feature selection is of crucial importance in those areas, since it helps improve the prediction performance of machine learning models by eliminating noisy variables, provide simpler models that facilitate better interpretation of the complex stochastic process, save the cost of large amount of experimental measurements in practice, and detect subset of variables that can be studied closely for causal inference.

Feature selection methods, though based on results from statistical learning theory, rely on thinking in engineering perspective. Feature selection itself can be viewed as a model selection process. Therefore, like other methods in the area of data analysis, there is no feature selection method that is optimal for all datasets. The methodology of feature selection involves feature ranking criterion designs (filter methods), search strategies (feature subset selection), model selection (wrappers with proper classifiers) and assessment methods (statistical tests for comparison of two methods, controlling false discovery rate etc.). By combining different analysis tools, better feature selection methods can be created in terms of the performances of classifiers (learning machine) or causal inference.

The goals of feature selection have been divided into two major categories: i) identifying minimal subset of features that optimize prediction accuracy and ii) finding (all) features that are relevant to the target. It is essential to understand the distinctness between these two goals in practice. For instance, in biomedical fields, the first goal is to eliminate the effects of noise and improve the accuracy of prediction. This is common in diagnosis system implementation and experiment design. As for the second goal which is common in pharmaceutical research, identifying the causation between the drugs and the genes or the regulative paths of a disease is important to efficiently discover the effects of new drugs. Generally, discovering (causal) relevance is different from improving prediction accuracy. It is seen in most of previous study that features selected based on these two goals do not correspond to each other, i.e. the set of features that optimize classification does not necessarily include features that are strongly relevant to target (Kohavi and John, 1997). For example, some classes of genes of biological importance are usually present in small amount and their effects cannot be easily detected for a feature selection algorithm focusing only on classification performance. Therefore, the need to investigate the intrinsic differences of the two aspects in feature selection is crucial in application. Nilsson et al. (2007) addressed these differences in various constrained probability distributions and proposed several polynomial algorithms which in turn

shed light on the complexity of those problems.

Feature selection has been widely used to improve prediction accuracy of classifiers. The improvement in prediction is related to the redundant features or noisy features in the data which can be eliminated by feature selection. On the other hand, the prediction accuracy can also be the guideline for feature selection, which is basic idea of wrapper methods (subset of features that yields the best empirical risk is chosen). In image analysis and natural language processing where noise is common, feature selection is both necessary and crucial as the first step in building predictive models..

The causal discovery perspective of feature selection becomes more and more importance in the area of biomedicine and disease study. Bayesian network is a graphical model broadly used to model generating processes of the data. Features that are relevant to the target can be seen as connections in the Bayesian network representation, under some assumptions. Bayesian network provides a very complete view of the generating process regarding the target. However, to learn a Bayesian network structure is NP-hard (Chickering et al., 2004). More importantly, Bayesian network is heavily parameterized, which requires large size of data to learn a model with reasonable power of generalization. Therefore, Bayesian network is usually not practical for feature selection. Instead, methods focusing on the local area of the target variable- Markov blanket is a compromise way for causal inference, in terms of less parameterization and more efficient in space and time. Such methods are partially guided by causality which are helpful for identifying more reliable relevance between the target and other features.

In this report, we focus on the study of these two aspects of feature selection methods. Simulations are conducted with synthetic data and several real data. Several feature selection methods are compared regarding their power in prediction and causal inference. In Chapter 2, a preliminary study of methodologies of feature selection is summarized. The concepts of relevance, causality, Markov blanket and Bayesian network are reviewed in Chapter 3. In Chapter 4, several feature selection algorithms are studied and analyzed regarding their design and performance, followed by a set of implementations and simulations for comparisons of different feature selection algorithms in Chapter 5. Finally, Chapter 6 concludes with a summary of the work, its limitation and future work.

Chapter 2

Feature Selection Methodologies

Feature selection has its root in statistical learning theory, which ensures the selection process and results are statistically sound. Many methodologies have been derived for feature selection specifically and some from other areas. In the following, a review of feature selection methodologies is presented.

2.1 Definition of feature selection

In machine learning, predictive modeling is a central problem with feature selection as a preprocessing step. Given the training data (features) and the associated outcomes (targets), predictive modeling aims to construct models to predict the target from the features or describe the relationships between the features and the target. For instance, in clinical study, it might be of interest to study the relationships between a certain cancer and the physical conditions as well as the genomic expression of the patients. To be more concrete, given the observations of the patients and their health status: having the cancer or not, the goal is to predict the health status of a new patient (test data). The performance of a model on new test data is called generalization.

We introduce here some notations and assumptions that will be used in the further discussions. $(\mathbf{x}^{(i)}, y^{(i)})$ i = 1,…,*m* denote the independent and identically distributed (*i.i.d.*) observations (training data and outcomes) of the random variables **X**,Y with probability distribution $P(\mathbf{x}, y)$, where $\boldsymbol{X} = (X_1, ..., X_n) \in \mathbb{R}^n$ is a feature vector and Y is the target variable. Lower case symbol x_i denotes the realization of the random variable X_i and bold symbols denote vectors. Probability of event is denoted as Pr, e.g. Pr(Y=1) is the probability that Y = 1, the probability density is denoted by lower case *p* i.e. *p*(**x**,y). Generally, the outcomes or targets can be continuous and multi-valued discrete. In this report, we focus on supervised binary classification problem i.e. the outcomes of the training data is given (supervised) and the outcomes of the target are binary {-1,1}.

There are a wide variety of models (classifiers) that can be applied on the binary classification problem including decision trees (Quinlan, 1986), support vector machine (SVM) (Cortes and Vapnik, 1995) and logistic regression (Agresti, 2002) etc., which will be discussed in more details in Chapter 4. The classifiers can be viewed as families of parameterized functions $f : \Lambda^d \times \mathbb{R}^n \to \mathbb{R}, (\boldsymbol{\alpha}, \boldsymbol{x}) \to f(\boldsymbol{\alpha}, \boldsymbol{x})$, where $\boldsymbol{\alpha}$ is the associated parameterization.

To measure the prediction accuracy of a model, a loss function *L* should be defined. Specifically, a loss function is used to assess the difference between the prediction and the true outcomes of the target variable. Given a classifier *f*, the general choices of L are shown as follows:

i) The L_1 hinge loss

$$L_{hinge}(f(\boldsymbol{\alpha},\boldsymbol{x}),y) \coloneqq |1-yf(\boldsymbol{\alpha},\boldsymbol{x})|_+$$

where $|z|_+$ =z if z>0 and 0 otherwise

ii) The L_2 loss

$$L_2(f(\boldsymbol{\alpha},\boldsymbol{x}),y) \coloneqq |f(\boldsymbol{\alpha},\boldsymbol{x})-y|^2$$

iii) The logistic loss

$$L_{\log istic}(f(\boldsymbol{\alpha},\boldsymbol{x}),y) \coloneqq \log(1+e^{-yf(\boldsymbol{\alpha},\boldsymbol{x})})$$

iv) The 0/1 loss

If the output of *f* is either -1 or 1, i.e. the membership of a class

$$L_{0/1}(f(\boldsymbol{\alpha},\boldsymbol{x}),y) \coloneqq |-yf(\boldsymbol{\alpha},\boldsymbol{x})|_+$$

L_2 is frequently used in linear regression, while the L_1 hinge loss is used in SVM. Logistic regression takes the similar loss (negative likelihood) as the logistic loss. The 0/1 loss simply adds 1 unit of loss whenever the prediction is different from the truth.

Given the classifier *f* and the loss function *L*, an expected risk function R can be defined regarding the predictive power of a model by incorporating the stochastic nature of the problem. The expected risk function is defined with respect to the joint probability distribution of $\boldsymbol{X}$ and Y, $P(\boldsymbol{x},y)$ i.e.

$$R(\boldsymbol{\alpha}) = \iint_{\boldsymbol{x},y} L[f(\boldsymbol{\alpha},\boldsymbol{x}),y]P(\boldsymbol{x},y)d\boldsymbol{x}dy$$

Generally, predictive modeling amounts to finding the optimal parameterization to minimize the expected risk function R with the training samples. The step of finding optimal alpha is usually regarded learning in the domain of machine learning and as model estimation in statistics.

Besides choosing a classifier *f* for a specific *L* with small expected risk, feature selection is another key step in predictive modeling. One goal of feature selection is to search for a subset of features that will improve the prediction accuracy (smaller R empirically). To use a subset of features instead of the full set inevitably brings the question of why extra information will not facilitate the prediction, even though some features are known to be irrelevant to the target. For instance, in the previous clinical example, other than the physical conditions and genomic expressions of the patients, room temperatures, diet information, even seasons can also be taken into consideration. However, adding such features in fact increases the dimensionality of the model and in turn tends to bury the relevant features into noise and irrelevant features (relevance is discussed in Chapter 3). Specifically, given finite number of samples, in practice, models learned with large number of features tend to be very sensitive to noise and fail to provide reliable prediction when present with new data (overfitting). Therefore, proper feature selection has the potential in improving prediction by reducing the dimensionality of the feature space.

Feature selection can be generally defined in the following way regarding prediction (based on Lal et al., 2003). Let $\sigma \in \{0,1\}^n$ be an *n*-dimensional indicator vector, where $\sigma_i = 1$ indicating that the feature *i* is present in the selected subset, and $\sigma_i = 0$ indicating the feature is absent.

Definition 2.1.1 (Feature Selection) Let f be a parameterized family of functions $f : \Lambda^d \times \mathbb{R}^n \to \mathbb{R}, (\boldsymbol{\alpha}, \boldsymbol{x}) \to f(\boldsymbol{\alpha}, \boldsymbol{x})$. Feature selection then amounts to finding an indicator vector $\boldsymbol{\sigma}^*$ and a parameterization of f $\boldsymbol{\alpha}^* \in \Lambda^d$ that minimize the expected risk

$$R(\boldsymbol{\alpha}, \boldsymbol{\sigma}) = \iint_{\boldsymbol{x}, y} L[f(\boldsymbol{\alpha}, \boldsymbol{\sigma} \odot \boldsymbol{x}), y] P(\boldsymbol{x}, y) d\boldsymbol{x} dy$$

where $\odot$ denotes element-wise product. L is a loss function and $P(\boldsymbol{x},y)$ is the probability distribution of ($\boldsymbol{X}$,Y). The feature subsets corresponding to $\boldsymbol{\sigma}^*$ is the optimal subset with respect to L and f.

In practice, however, the definition above is not sufficient to capture the characteristics of the problem. We can understand the practical issues by looking at the terms in the definition more closely.

2.1.1 Estimate $R(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ and generalization errors

In most cases, especially for high dimensional genomic data, the distribution $P(\boldsymbol{x},y)$ does not have reliable empirically estimate from the data. Actually, estimating $P(\boldsymbol{x},y)$ is a far more difficult problem than predictive modeling. To overcome this, by the law of large number, the integral in Definition 2.1.1 can be replaced by the arithmetic mean of the losses over the observations (training data) called the empirical risk.

$$\hat{R}(\boldsymbol{\alpha}, \boldsymbol{\sigma}) = \frac{1}{m} \sum_{i=1}^{m} L[f(\boldsymbol{\alpha}, \boldsymbol{\sigma} \odot \boldsymbol{x}^{(i)}), y^{(i)}]$$

Empirical risk is a straightforward approximation for expected risk, but in practice provide little information about the generalization ability of a model (feature subsets), especially when the training data is of small size or dimension of $\boldsymbol{X}$ is large. This is due to the fact that the empirical risk is estimated from only the training data, and relies heavily on the convergence speed by the law of large number. Therefore, another measure for generalization ability is needed.

There are various schemes to assess the generalization ability of models i.e. generalization errors. Intuitive, a separated test data set can be used to evaluate the predictive power of the model estimated from training data, given that the test data are from the same distribution. For a small data set, the generalization error estimated from such a single split is very inaccurate. To overcome this, the general k-fold cross-validation (CV) split (usually randomly) the training data into k subsets with (approximately) equal sizes. Iteratively, the model is learned with one subset left out and that subset is then use to compute an error measure for the learned model by averaging the errors over the leave-out subset. Then the k-fold CV generalization error estimate is the average of the k averaged error terms.

2.1.2 The indicator vector $\boldsymbol{\sigma}$

To search for the optimal $\boldsymbol{\sigma}$ for a fixed parameterization $\boldsymbol{\alpha}^*$ and f is a combinatorial problem, which is NP-hard with the search space being $2^n - 1$. Therefore, feature selection algorithms either use greedy heuristics to avoid searching over full space or relax the 0/1 integer constraints on indicator vector to real domain [0,1] as weights $\boldsymbol{w}$ (as we shall see in embedded methods). Additionally, further constraints can be imposed on $\boldsymbol{\sigma}$ to input prior via the term $\Omega(\boldsymbol{\sigma})$ which is some norm measure on the (relaxed) indicator vector :

i) ℓ_0 norm: $\Omega(\boldsymbol{\sigma}) = \ell_0(\boldsymbol{\sigma})$ the number of non-zero elements in $\boldsymbol{\sigma}$

ii) ℓ_1 norm: $\Omega(\boldsymbol{\sigma}) = \sum_{i=1}^{n} |\sigma_i|$

iii) ℓ_2 norm: $\Omega(\boldsymbol{\sigma}) = \sum_{i=1}^{n} \|\sigma_i\|^2$.

$\Omega_c(\boldsymbol{\sigma})$ is then either imposed as constraint ($\Omega_c(\boldsymbol{\sigma}) \le 1$)

$$\min_{\sigma,\alpha} R(\boldsymbol{\alpha}, \boldsymbol{\sigma}) \quad \text{subject to} \quad \Omega_c(\boldsymbol{\sigma}) \le 1$$

or is added to the empirical risk term to form a new optimization problem,

$$\min_{\sigma,\alpha} R(\boldsymbol{\alpha}, \boldsymbol{\sigma}) + \Omega_c(\boldsymbol{\sigma})$$

where C is a constant to control the magnitude of penalization by $\Omega(\boldsymbol{\sigma})$ i.e. $\Omega_c(\boldsymbol{\sigma}) \equiv C.\Omega(\boldsymbol{\sigma})$

2.1.3 The loss function *L* and the classifier *f*

Feature selection is sensitive to the choices of L and f. One can proved that optimal features selected for a pair (L,f) is not necessary optimal for another pair (L^*,f^*) (Tsamardinos et al., 2003a). Therefore, feature selection should be conducted with different combinations of (L,f) to compare the results, while the performance of a feature subset should always be compared with the same combination of L and f.

In statistical learning theory, assuming the exact probability distribution $P(\boldsymbol{x},y)$ is known and both $\mathbf{X}$ and Y are discrete, *Bayes classifier* is defined as the function,

$$f_{Bayes}(\boldsymbol{x}) = \begin{cases} +1, & \Pr(Y=1|\boldsymbol{X}=\boldsymbol{x}) \ge 1/2 \\ -1, & \text{otherwise} \end{cases}$$

Bayes classifier is proved to be the best classifier with minimal 0/1 prediction errors (Bayes errors) and is seen as the bound of the prediction performance of a practical classifier. The Bayes classifier is of only theoretical importance by providing designing principles, since it is assumed that the Bayes classifier knows exactly $P(\mathbf{x},y)$ which is rarely possible for high dimensional data..

2.1.4 Variations of feature selection

The previous definition (Definition 2.1.1) focuses on finding a set of features that

minimize some loss function (L) for the underlying classifier (f) i.e. improving prediction performances. However, a feature that is informative in predicting the target Y is not necessary relevant. It is observed for SVM, adding noisy features with the same zero mean random noise would reduce the overall influence of a single noisy feature to the SVM (Lal et al., 2003). Therefore, when the relevance of a feature to the target is of more concern, the design of feature algorithms should have schemes to cope with such inconsistency between prediction and relevance (Nilsson et al., 2007, Yu and Liu, 2004). We shall look into the problem in more details in Chapter 3.

2.2 Principles of Model selection

Feature selection can be regarded as part of model selection, where feature selection is performed as integrated part of the model building. On the other hand, feature selection can itself be seen as model selection i.e. to choose a model that can select the best subset of features for prediction or causal inference. Therefore, it is useful to review some general aspects in model selection before looking into specific feature selection methods.

Model selection is the most fundamental step in data analysis. Given a dataset D and a set of models $\{M_i\}$, model selection amounts to choosing a model that best explains the data in a specific perspective (prediction, feature selection or Bayesian network learning etc.). A model is a function or functional with a set of parameterization. Here, the dataset D is assumed to be stochastic. One of the most important statistical principles of model selection is to balance bias and variance of a model. In the terms of statistical learning, bias refers to the errors occurring when the model is not complex enough to describe the true generating process while variance refers to the errors when the model has more degrees of freedom than the true generating process (overfitting). Such decomposition is very informative for linear regression problem with mean-square loss, where the mean square errors can be decomposed into bias term (errors related mismatches between the model and the true generating process) and variance term (errors related to learning from finite samples).

To see the importance of balancing bias and variance, the simplest example is the two-dimensional curve fitting problem: given a set of points $\{u^{(i)},v^{(i)}\}$,fit a curve to the data. Bias can be introduced when a linear model is fit to the data when $\{u^{(i)},v^{(i)}\}$ are points in a third degree polynomial curve. On the other hand, variance is large when fitting a higher degree polynomial to $\{u^{(i)},v^{(i)}\}$ with noisy linear relationship. Bias and variance tradeoff is the nature of model selection due to its attempt to reverse engineering the probabilistic and complex generating process from data with finite parameterization, which is known to relate to inverse problem.

The tradeoff of bias and variance is generally resolved with proper penalization in the models. Several penalization criterions have been proposed and widely used. First of all, it is useful to introduce the concept of complexity of a model. Models may differ in i) sets of features ii) model complexity: e.g. number of parameters in linear models, number of support vectors for SVM iii) initial parameter values or stopping criteria. Complex models with complicated structures, large amount of parameters or

unbounded stopping criterion in search tend to fit the training data well. However, estimating such models with small dataset will introduce the problem of overfitting where the model fits the training data extremely well but fails to predict new data. To address this issue, Akaike (1974) proposed Akaike's information criterion (AIC) to penalize the number of free parameters n in model selection. Schwarz (1978) introduced the Bayesian information criterion (BIC) with heavier penalization on the n. Neither methods address the intrinsic complexity of the function of the model. Alternatively, Vapnik and Chervonenkis (1971) defined a measure of the capacity of a classifier (model) as the cardinality of the largest set of points that the classifier can shatter, VC-dimension. Structure risk minimization with VC-dimension (SRMVC) takes into account of the complexity of the modeling function regarding its power that has connection to the number of free parameters. Empirically, cross-validation (CV) scheme is used to access the estimation of bias and variance of a model. The errors calculated from cross-validation can be proved to be consistent estimate of the bounds for errors in a new dataset from the same distribution.

In practice, Moore (2001) compared the AIC, BIC, SRMVC and CV-errors on simulated data and pointed out that as the sample size goes to infinity, AIC is equivalent to leave-one-out CV and BIC is asymptotically equivalent to a carefully chosen k-fold CV and tends to perform better for Bayesian network structure learning and clustering. As for SRMVC, it is observed to be very conservative regarding bounds. CV is computational intensive and has larger variance, since CV itself involves multiple training on the dataset.

2.3 Feature selection general guidelines

It is very useful to understand the general steps of solving a feature selection problem. Guyon and Elisseeff (2003) suggested the following steps:

1. Feature constructions
2. Feature ranking with filter methods
3. Excluding outliers
4. Comparisons between different feature selection methods
5. Assessing stable solution

The first step preprocesses the data by transforming them with some prior knowledge. For instance, when the measurements involve counts or very small real values (e.g. on the order of 10^{-5}), it is reasonable to take the logarithms of data to recover the nature of distribution. On the other hand, the measurements of a set of features do not usually commensurate. Therefore, normalization over samples or features is a basic step to avoid the effects of large differences on levels. Such differences might be sensitive to some classifiers such as SVM and Fisher's linear discriminator. Furthermore, conjunctive features or linear combinations of features (principle component analysis) are also of interest if features do not need to be analyzed separately. The second step is to use filters as baseline results before applying more complicated methods. Thirdly, since outliers affect the performance of most classifiers, discarding outliers is an essential step before evaluating the

performance of feature selection methods. One may detect outlier examples using the top ranking variables from filter methods or smooth the effects of outliers with a sigmoid function. With outliners eliminated, the fourth step starts with linear classifier, combining with forward selection with "probe" as stopping criterion If increasing subsets of features can improve or match the performance with a small subset using a sequence of predictors of same nature, evaluate the subset with a non-linear classifier. Other methods can also be applied and should be compared. Finally, one might resample the data (if the data set is not large enough) and evaluate the results with bootstrapping.

2.4 Approaches of feature selection

Generally, the approaches of feature selection can be divided into three types: filters, wrappers and embedded methods. These approaches differ in three ways i.e. (according to Guyon and Elisseeff, 2003)

- search strategies
- evaluation criterion definition (e.g. relevance index or prediction of classifiers)
- evaluation criterion estimation (statistical test or cross-validation/performance bounds)

2.4.1 Filters

Filters estimate a relevance index for each feature to measure how relevant a feature is to the target. Then filters rank features by their relevance indices and perform search according to the ranks or based on some statistical criterion e.g. significance level. The most distinguishing characteristic of filters is that the relevance index is calculated based solely on a single feature without considering the values of other features. Such implementation implies that filters assume orthogonally between features which usually is not true in practice. Therefore, filters omit any conditional dependence (or independence) that might exist, which is known to be one of the weaknesses of filters, since they might miss optimal subset of features. However, filters are efficient and proved to be more robust to overfitting theoretically (Ng, 1998).

There are various heuristics to design relevance indices for filters, including univariate prediction error rate (i.e. evaluate the relevance of a feature as how accurate the prediction is using only itself), correlation-based (e.g. Pearson coefficient, signal to noise ratio), distances between distributions (K-L divergence, Jeffreys-Matusita distance), information theory (mutual information, Minimum Description Length (MDL)), decision trees (C45, CART), Relief (a class of filters incorporating sample relations into feature selection). Most of heuristics are derived from their relations to the bounds of Bayes errors of single feature. On the other hand, they differ in how to use data to evaluate the usefulness of a single feature. Heuristics other than decision trees and Relief are global, i.e. they do not account for distances between samples. Relief makes use of the local information of an area of the feature

space to calculate the average usefulness of a feature, since features can be relevant to the target in some area. Decision trees divide the feature space hierarchically to investigate the relevance of features at different stages. Such local information will be helpful when the data domain is complicated, especially is image analysis.

There is no general guideline for choosing the most appropriate relevance index for a problem. However, the performance of relevance indices are related to the type of data (binary, integer or continuous) and prior information about the data distribution, according to their definitions and properties. Other than those, the following matters are worth noting. First of all, different relevance indices may rank features in distinct orders, since the abilities of relevance indices to capture dependence might vary. Secondly, there is always bias and variance when the relevance indices are estimated from data. Generally, complex relevance indices that are less biased could be very unstable (large variance) e.g. MDL is more robust to bias regarding discovering the non-linear dependences. However, MDL tends to have large variance when estimated from data with low data to features ratio and causes over-fitting. Therefore, one should be careful when evaluating the ranking of features using complicated relevance indices.

Contrast to the traditional univariate filters, recently, several authors succeeded in applying conditional independence tests to filter out feature that are not in the Markov blanket of the target (Aliferis et al., 2003b, Nilsson et al., 2007) (Markov blanket can be seen as the subset of features conditional on which the target is independent of any other features). Such methods no longer assume orthogonality of features and search the feature space in a recursive way to efficiently test conditional independence. They have similarities to univariate filters in that they do not rely on any classifiers but depend only on the power of the independence tests. The designs of those methods are discussed in more details after the formal definition of Markov blanket in Chapter 3.

2.4.2 Wrappers

Instead of ranking every single feature, wrappers rank feature subsets by the prediction performance of a classifier on the given subset, which were first proposed by Kohavi and John (1997). Unlike filters, wrappers can be used to search through all possible subsets of features and explore the mutual information between features. After choosing a classifier (preferably consistent), wrappers evaluate the prediction performance either by cross-validation or theoretical performance bounds. Other than the choices of classifiers, wrappers differ in the underlying search strategies. Exhaustively searching combinatorial subsets is NP-hard and is prone to overfitting. Therefore, greedy search strategies are generally preferred, such as sequential forward selection or backward elimination. Since search strategy is a topic important for both wrappers and embedded methods, the details of search strategies are discussed later (Section 2.5).

The idea of wrapper has been used before Kohavi and John's proposal. For instance, the use of AIC in model selection for linear regressions, where the feature subsets are compared based on AIC, i.e. the subset with smaller AIC is prefer. To search for optimal subset, one need to apply search strategy, either evaluate AIC over

all possible subsets, or start with whole set and eliminate one at a time the feature without which the AIC is the smallest. Kohavi and John (1997) presented a more formal discussion of this kind of methodology by introducing variability in choices of classifiers and search strategies.

2.4.3 Embedded methods

Embedded methods select features based on criterions that are generated during the learning process of a specific classifier. In contrast to wrappers, they do not separate the learning from the feature selection part, i.e. the selected features are sensitive to the structures of the underlying classifiers. For this reason, in most cases, the feature selected by one embedded methods might not be suitable for others. Formally, embedded methods are designed explicitly or implicitly to approximate solutions of the minimization problem with respect to weights for features and the parameterization of a classifier. The methods to pursue such approximate solutions can be 1) Greedy search based on the gradient between the empirical risk and the weight indicators. Methods of this kind include Least Angle Regression (LARS) (Efron et al., 2004), Recursive feature elimination (Guyon et al., 2002) and decision trees, 2) Relaxation of the integrality restriction on weight indicators. The minimization problem is then solved either with gradient descent regarding the bounds for generalization errors (Weston et al 2000) or incorporating proper priors for the weights (Joint Classifier and feature Optimization (JCFO) Krishnapuram et al., 2004) and 3) Inclusion of a sparsity term in the minimization problem (for linear models). Methods of this category combine the loss function in the original problem with a regularization term, which is usually the l_0 norm or l_1 norm of the weights. There are some potential advantages of using l_0 *or* l_1 norm regularization to the widely used l_2 norm due to the convexity of l_2 norm, which will be discussed in more details in the next chapter. Least Absolute Shrinkage and Selection Operator (LASSO, Tibshirani, 1996), Generalized LASSO (Roth, 2003) and 1-norm SVM (Zhu et al., 2003) use l_1 norm whereas Feature Selection Concave (Bradley and Mangasarian, 1998) and Multiplicative Update (Weston et al., 2003) approximate the l_0 norm by a smooth function whose gradient direction is known to avoid NP-hardness and overfitting.

2.4.4 Comparisons and Ensemble methods

The three approaches of feature selection methods are different in their design methodology and each has its own strongness and weakness. As far as computation efficiency is concerned, filters do not require any model learning and thus are time efficient. Wrappers are generally the slowest among the three, since the cross-validation procedures on every iteration are costly. Embedded methods may incorporate schemes to speed up the evaluation of quality of a feature subset to avoid the cross-validation. On the other hand, filters have the lowest model complexity. Embedded methods and wrappers tend to have higher complexity than filters for the

parameterization in classifiers. Nevertheless, filters, as mentioned above, usually miss capturing mutual information between features, and thus tend to be biased.

The choices of feature selection methods are problem-dependent. When the training sample size is small, filters are expected to perform better for their ability to provide more stable estimation, whereas more advanced wrappers and embedded methods will outperform filter methods when the training sample size increases, for their abilities to detect mutual information between features. Other than the size of data set, a closer look at the data via different data analysis tools will be helpful to identifying the characteristics of the data. For classification, visualization of the target variable among two or three features could assist in selection of parameterization of methods (say. whether the target is linearly separable, if not, choose a non-linear kernel for SVM).

Supervised ensemble methods represent a methodology of constructing a set of base algorithms and using their weighted outcome for prediction (feature selection). Theoretically, ensemble methods are confirmed to be more stable for generalization (Poggio et al., 2002, Poggio et al., 2004). They also tend to perform better when there exist redundant features (redundancy is related to relevance, Chapter 3). For feature selection, a simple example of ensemble method can be: choose a set of feature selection algorithm e.g. a set of different relevance indices for filters, and then take the intersections or union of the top ranking features or choose features based on their frequency in the top ranking subsets. Another example is that random forest for feature selection, where ensemble of decision trees is used to implicitly select features. The condition for ensemble methods to reduce variance of the optimal subset is that the set of base algorithms should not be correlated (or only weakly correlated) and not be of high complexities (capacities).

2.5 Search strategies

Search strategies are crucial in designing feature selection algorithms, especially for wrappers, where optimal subset is to be identified by searching through the subset space. Exhaustive search is both not practical and prone to overfitting, which should be overcome by proper search strategies.

2.5.1 Deterministic Search

Deterministic search represents a large class of search strategies, which does not involve randomness in the search procedures, i.e. search proceeds according to pre-specified schemes.

Sequential backward selection (SBS) and sequential forward selection (SFS) are two most basic search strategies that many advanced search strategies based on. SBS (Maril and Green, 1963) starts with the full set of features, and eliminates a feature whose elimination yields the best result according to an evaluation function (e.g. empirical risk function) at each step until a specified number of features are left or other stopping criterions are met. On the contrary, SFS (Whitney, 1971) starts with an

empty set and adds a feature whose inclusion yields the best result until the method reaches a specified number of features or obtains no gain on the evaluation function. Concerning the performance for the two, SFS is generally faster than SBS, since it involves fewer features in the beginning, while SBS progresses with larger set of features backwardly. However, SFS is known to fail in XOR-like problem (Guyon and Elisseeff, 2003), where a feature is not informative without the other feature. SFS has difficulty in including such a feature when the other feature is not chosen in an earlier step. Nevertheless, no empirical evidences suggest that one is superior to the other for these two methods (Kudo and Sklansky, 2000), which might imply that such XOR effects are mitigated by noise or the present of other features.

Instead of examining one feature at each step, both SBS and SFS can be generalized to evaluating a fixed size k subset every step, which increases the computation to polynomial up to the degree of k. Once again, even though there exist synthetic examples where one-at-time algorithms fail, no theoretical evidence supports that the generalization versions are better.

Further generalizations of the forward and backward strategies include

i) Backtracking During Search, in which both inclusion of l features and elimination of r features occurs in each step
ii) Beam Search (Siedlecki and Sklansky, 1988, Aha and Bankert, 1996) which generates candidate search branches that can be visited again to extend the search space
iii) Floating Search (Pudil et al., 1994), a combination of a one step SBS (SFS) and a SFS (SBS) until the best subset is found.

There are other modifications varying on how subsets of features are incorporated into evaluation or how to avoid redundant processing etc. However, when the search strategies become complicated, the problem of overfitting arises. For example, the backtracking in floating search might stop after a long time (or may not stop at all). Therefore, the optimal subset might be lack of statistical significance for new data. A proper stopping criterion is needed in this case to avoid overfitting.

All the methods presented above maintained the candidate subset(s) which will be evaluated at each step. Nilsson et al. (2007) recently suggest a backward-like wrapper algorithm that is consistent for *strictly positive distribution* (in Section 3.2) to identify Markov blanket if the black-box classifier is consistent to Bayes classifier. Unlike the previously mentioned methods, the algorithm does not evaluate the candidate subset directly and adds the feature without which the prediction error increases (by an amount set as a parameter) compared to the full set of features. The prediction performance of the optimal subset is never evaluated during the search. It is linear to the number of features, but its time efficiency depends mainly on the learning of the classifier at each step. We will study this method in more details in the experiment section.

2.5.2 Stochastic Search

For a given data set and a particular initialization, a deterministic search strategy always returns the same subset, which makes it extremely sensitive to the change of

the data set. Randomizing the search schemes (stochastic search) is a reasonable way to introduce randomness to account for the stochastic nature of the data (as in boosting). Moreover, stochastic search strategies usually converge fast and have sound theoretical consistency in finding sub-optimal results, which is preferable in avoiding overfitting.

2.5.2.1 Simulated Annealing

Simulated annealing starts with an initial subset that is chosen either randomly or from the outputs of some feature selection methods. At each step, the current subset is subject to some small random change. If the change produces a subset better prediction performance, then it is accepted as a new candidate subset. While the change results in worse subset, it is accepted with a probability that is dependent on the 'temperature'. The temperature is high at the initial state and decreases in the course in a preset rate. The low temperature at the end ensures that the algorithms will produce a local optimal. Simulated Annealing was independently proposed by Kirkpatrick, Gelatt and Vecchi in 1983 and by Černý in 1985, and was proved to be very useful in optimization problems. It can be easily implemented but rely greatly on the choices change schemes, temperature decreasing rate etc. which might be varied for different data set.

2.5.2.2 Genetic Algorithms

Genetic algorithms utilize a set of candidate subsets (population) instead of one subset in simulated annealing. At each step, the set of subsets undergo i) mutations: minor random changes in the subset ii) crossover: the subset is changed based on other subset in the population by including features that belongs various parts of the other subsets. When changes are made accordingly, subsets in the population with better performance is chosen with high probability into the next generation. The algorithm stops until the pre-specified number of generation is reached or other criterion is met. Genetic search algorithms have great power in combinatorial problems such as Traveling Salesman Problem (TSP), it is also reviewed by (Kudo and Sklansky, 2000) for feature selection.

2.5.2.3 Randomized backward selection

For deterministic sequential backward selection, each feature is eliminated only if its elimination gives better prediction result. Stracuzzi and Utgoff (2004) proposed a randomized version of backward elimination. In their framework, the probability that a relevant feature is included in a randomly selected subset of size k is computed. Such subset of size k is considered for elimination from the candidate subset and the value of k is chosen to ensure that the removal of k variables has high probability of being truly irrelevant. If the removal causes the error estimate to increase, one or more of the pruned features are considered to be indeed relevant. Therefore, the removal is cancelled and another new random subset is chosen. The algorithm stops until a preset number of consecutive cancellations are encountered, which can be seen as a sign that all the features in the current candidate subset are relevant. Stracuzzi and Utgoff pointed out that this scheme works well when the proportion of relevant features is

small.

Most of search strategies discussed in this part can be incorporated with various classifiers to form wrappers. Similar to the choices of feature selection methods, there are generally no empirical evidence supports that any search strategy is universally better than others. In practice, such choices are dependent on the sample size and the nature of the data distribution.

2.6 Statistical aspects of feature selection

The feature subset output by a feature selection algorithm should be evaluated statistically. General properties of the selected subset, such as significance level, false positive rate, false discovery rate etc. should be assessed after feature selection is performed. Additionally, issues related to the performance of feature selection algorithms are shortly discussed with respect to statistical learning theory.

2.6.1 Subset Quality Evaluation

The false positive rate (FPR) is the probability that a feature is not relevant while being selected by the underlying feature selection algorithm. The falsely selected features are called false positives. For filters, the false positive rate can generally be controlled by setting the significance level of relevance indices, which can be computed analytically using parametric or nonparametric hypothesis testing (based on large sample properties), e.g. T-test criterion, ANOVA, Wilcoxon test and AUC criterion Chi-square statistics etc. However, in reality the assumption for hypothesis testing is sometimes not satisfied e.g. the distribution of the features is not Gaussian or the number of samples is usually too small to have asymptotic properties. More importantly, the distributions of most relevance indices cannot be calculated analytically. According to statistical learning theory, when distribution functions are not the goal, one should never estimate them due to the complexity of the problem. Thus, one should always circumvent such complexity by other techniques. Alternatively, Oukhellou et al. (1998) suggested that generating "random probes" i.e. features that are not related to Y and comparing their relevance with the relevance of candidate features is a good way to estimate false positive rate. In order to mimic the 'behavior' of irrelevant features, the random probes should have connections to the data at hand. In Dreyfus and Guyon (2003), two ways to generate random probes were given

i) generating random features with a distribution which is similar to that of the irrelevant candidate features (e.g. a normal distribution)

ii) permuting the values of the features across observations in the training data (similar to the methodology of permutation test)

By computing the relevance indices of candidate features and generating random probes, one can choose a threshold r_0 on the relevance index that guarantees an upper bound on FPR. The FPR is defined as the ratio of the number of false positives to the

total number of irrelevant features. Under the assumption that the distribution of probe variables is similar to the distribution of irrelevant variables and that the number of generated probes is large, the FPR for this threshold can be estimated as the ratio of the number of selected probes to the total number of probes. For instance, when univariate forward selection is performed, assessing FPR can be summarized in the following steps:

1. Generate random probes using the methods mentioned above
2. Compute the relevance index for both candidate features and probes
3. Rank the features including probes in decreasing order
4. Stop the selection progress when the ratio of the selected probes to total number of probes is larger than or equal to the chosen FPR

Variants of this can be used in the framework of wrappers or embedded methods.

In feature subset selection, FPR is misleading by underestimate the p-value since multiple testing is performed. Bonferroni correction, as an attempt to overcome this problem is based on the first order approximation of the independent testing assumption. However, it increases the specificity by overestimate the p-value (Perneger, 1998). Benjamin and Hochberg (1995) proposed the false discovery rate (FDR) as a further correction which has been used intensively in medical testing. It is defined as the ratio of false positives to the total number of selected features. Genovese and Wasserman (2002) showed that FDR is a measure that is intermediate between Bonferroni correction and no correction. More importantly, FDR is robust in the sense that it provides a better estimation of the number of false positives. Two ways of estimating FDR approximately using random probes is given in (Dreyfu and Guyon, 2003), which should be used based on the fraction of relevant features.

2.6.2 Performance evaluation

The prediction performance of feature subsets should be assessed statistically especially the size of test samples is small. For a classification problem, the misclassification rate (number of misclassifications divided by the number of test samples) is usually used to measure the prediction performance of a classifier or feature subset. We denote the misclassification rate as E. It has standard error $stderr(E) = \sqrt{\frac{E(1-E)}{m_t}}$, where m_t is the number of test samples (Guyon et al., 1998). Therefore, the comparisons of the prediction performance between two feature subsets should be evaluated with cautions. Guyon et al. (1998) suggested the McNemar's test when comparing the prediction performance with misclassification rate for different feature subsets. McNemar's test a non-parametric method used originally to determine the potential treatment effects (where the performance of the two subsets can be seen as different effects before and after treatments):

Assuming i.i.d. errors, one-sided test and approximating binomial with normal law, E_i the average number of misclassifications where the misclassifications are *only* made by feature subset *i* but not by the other. Then the McNemar's statistic in the following is chi-square distributed with 1 degree of freedom. If the statistic is

significant (e.g.5% significance level), then the feature subset with small generalization errors is significantly better than the other in prediction.

$$\chi^2 = m_t \cdot \frac{(E_1 - E_2)^2}{E_1 + E_2}$$

McNemar's test will lose power when $m_t(E_1 + E_2)$ is small (<20), in which case, the exact Binomial test should be used instead. For binomial test, the null hypothesis is that both models are equivalent. Thus, the rate of errors that only one model makes should be equal, i.e. for an observed error, the probability that it is only made by either model should be 0.5. Therefore, the differences in number of errors u only made by oneself (i.e. $m_t \mid E_1 - E_2 \mid$) should be distributed with binomial distribution $Bin(m_t(E_1 + E_2), 0.5)$. The confidence interval can then be estimated by assessing the inverse of the binomial distribution.

2.6.3 Consistency

The size of data is an important factor affecting the performance of model selection. Consistency evaluates the convergence to a property of a model when the size of the data goes to infinity. The convergence and the corresponding property provide assessment of the behavior of a model. For example, k-nearest neighbor (k-nn) is consistent to Bayes classifier (consistent classifier means the prediction performance of such classifiers converge to Bayes classifier when the sample is infinite). In practice, the performance of consistent model relies on closely the size of the data. When the size of train data is small, consistent methods with slow convergence rate might not perform well in practice.

2.6.4 Overfitting

In model selection, overfitting is a crucial issue that should be carefully coped with in practice. Theoretically, overfitting is related to the ill-posedness of inverse problem when trying to model the continuous and infinite world with finite and(or) discrete parameters. Overfitting occurs when the model is selected for best describing the training data, but fails to fit new data. For feature selection, one should always choose methods with caution of overfitting, especially when the size of training data is small. As earlier explained, filters might be preferred to more complicated methods like wrappers.

Another practical issue of overfitting is in the process of learning a model. As showed before, when estimating a model, besides the parameters, there are other factors that should be also decided upon the data. Those factors can be called meta-parameters including e.g. the number of features, the number of trees in random forest or stopping criterion. In model selection framework, data generally are split into three parts: training set that is used for parameter estimation, a validation set that is used for model selection and a test set that is used for evaluate the generalization ability of the selected model. The comparisons for the performance of feature selection algorithms with their errors on validation set should always be avoided. Such comparisons are problematic, since the meta-parameters (model selection) are

trained on the validation set, which might reduce the generalization ability of the model and in turn widen the confidence interval on the generalization error. On the other hand, regardless of its high computation requirement, searching for optimal parameter (or models, feature subsets) can be problematic since it decreases the statistical significance of the solution with finite data size in practice. Therefore, some form regularization (e.g. penalization on the some norm of the parameters) and greedy criterion (e.g. early stopping in search) should be included in model learning to cope with overfitting.

Chapter 3

Relevance and Causality

In this chapter, we review some concepts on relevance and causality. The relevance between the features and the target variable has crucial implication on feature selection. Intuitively, features should be selected based on the degree of relevance. Therefore, it is necessary to clarify the different definitions of relevance. On the other hand, for causal feature selection, the causal structure is the central t that should be carefully evaluated.

3.1 Relevance

Kohavi and John (1997) defined relevance in strong and weak sense regarding the variation of conditional distribution of target Y on different subset of features. $\boldsymbol{R}_i \in \boldsymbol{X} \setminus \{X_i\}$ represents the set of all features in $\boldsymbol{X}$ except X_i. The definitions in this chapter assume all the features are discrete if not otherwise stated. The following definitions can be extended when any one of the features is continuous, by replacing the probability (Pr) with cumulative distribution.

Definition 3.1.1 (Strong relevance). A feature X_i is strongly relevant if and only if there exists some x_i, y, and r_i for which $Pr(\boldsymbol{X}_i=\boldsymbol{x_i}, \boldsymbol{R}_i=\boldsymbol{r}_i) >0$ such that

$$\Pr(Y = y \mid X_i = x_i, R_i = \boldsymbol{r}_i) \neq \Pr(Y = y \mid R_i = \boldsymbol{r}_i)$$

Definition 3.1.2 (Weak relevance). A feature X_i is weakly relevant if and only if it is not strongly relevant, and there exists a subset of features $\boldsymbol{R}'_i$ of $\boldsymbol{R}_i$ for which there exists some x_i, y and r'_i with $Pr(\boldsymbol{X}_i=\boldsymbol{x_i}, \boldsymbol{R}'_i=\boldsymbol{r}'_i) >0$ such that

$$\Pr(Y = y \mid X_i = \boldsymbol{x}_i, R'_i = \boldsymbol{r}'_i) \neq \Pr(Y = y \mid R'_i = \boldsymbol{r}'_i)$$

Relevant features are either weakly relevant or strongly relevant. A strongly relevant feature contains information of the distribution of Y which no other features can replace, whereas a weakly feature can be useful only when some features are not present. As for prediction, Kohavi and John (1997) argued that in practice relevance does not imply optimality for classifiers induced from data and vise versa in sense of prediction. As a matter of fact, irrelevant features can be informative in prediction when the data is noisy, since such features as whole could be included to extract information on the noise. To circumvent the ambiguity in KJ's definition for relevance, Yu and Liu (2004) introduced the concept of redundancy, and subcategorized weakly relevant features into redundant and non-redundant subsets with their algorithms.

Guyon and Elisseeff (2003) defined feature irrelevance and subset relevance probabilistically in a similar form as KJ's (based on conditioning), but focused on the specification of redundancy.

When the performance of prediction is the primary goal for feature selection, it is of interest to investigate correspondence between optimal subset for prediction and relevant features. As previous discussed, KJ's definition relevance fails to completely capture such a correspondence. Tsamardinos et al. (2003a) suggested the definition of relevance for a target Y should be functions of each feature subsets instead of individual features. Additionally, they argued that relevance should be defined with respect to the underlying classifiers and loss functions by proving that there exists feature relevance given the probability distribution of data and the classifier (or loss function) that is dependent on the loss function (or the classifier). They pointed out that according to the No free lunch theorems for search and optimization (Wolper and Macready, 1997), the average performance of different optimization or search algorithms (wrappers) over all possible problems is equivalent, which is another evidence that feature relevance should be related the classifier and loss function. In all, for the existences of such extreme example, Tsamardinos et al. (2003a) concluded that in principle, feature selection methods should be evaluated based on specific classes of loss functions and classifier. Similar argument is also given by Antos et al. (1999) who had shown that no Bayes errors estimate can be trusted for all data distribution, not even if the data size goes to infinity. For errors estimated by consistent classifiers such as k-nn, the speed of convergence to Bayes errors is arbitrarily slow. Therefore, it is not possible to make conclusion of universally superior feature selection methods without constraints.

Regarding feature relevance, Nilsson et al. (2007) defined feature selection in new perspective: given the data, to find all the relevant features (both strongly and weakly) to the target. They proved that finding all relevant features is much harder than finding optimal subset for prediction. It is NP-hard in strictly positive distribution for which the latter has polynomial algorithms (O(n), n is the number of features). A polynomial algorithm of identifying all relevant features was proposed for a more constraint class of distribution which is strictly positive and satisfies composition and weak transitivity. The most important implication from their discussion is that when evaluating a feature selection algorithm, besides the classifier f and the loss function L, the distribution of the given data should also be taken into consideration.

3.2 Markov Blanket

Markov blanket of a (target) variable is defined as the subset of features conditional on which the distribution of the (target) variable is independent of other features. Markov blanket in many cases turns out to be a competitive features selection guideline to relevance (Koller and Sahami, 1997, Tsamardinos et al., 2003a, Yu and Liu, 2004, Nilsson et al., 2007).

Definition 3.2.1 (Markov Blanket)
Let $\Phi=\{\mathbf{X},Y\}$ be a set of variables, the Markov Blanket of a variable Y, denoted as MB(Y) is a minimal set of variables, such that $\forall V \in \Phi \setminus \{Y\}$, $Pr(Y|MB(Y),V) = Pr(Y|MB(Y))$

By the form $Pr(Y|MB(Y),V) = Pr(Y|MB(Y))$, it means that the equality holds for all realizations of the variables. We can interpret Markov blanket of Y as the feature subset conditional on which Y is independent of any other features in the given set Φ. Therefore, theoretically, MB(Y) is the minimal subset sufficient to provide the best prediction for Y. We will see later that such assertion is not necessarily true in practice.

Markov blanket has close connections to the concept of Bayesian network which gives graphical interpretation of the dependence structures between features. Bayesian network is directed acyclic graph (DAG) with its nodes representing random variables (features), and edges representing the dependences between features. The edges are directed to specify to describe the conditional independence between features. Cycles are avoided in the model for cycles introduce ambiguity to the conditional independency between features.

Definition 3.2.2 (Bayesian Network)
Let $\Phi = \{\boldsymbol{X}, Y\}$ be a set of variables and J be a joint probability distribution over Φ and G be a directed acyclic graph (DAG) over a subset of variables $S \subset \Phi$. Let all nodes in G have one-to-one correspondence to the variables in Φ subject to the Markov condition i.e. for every node $V \in \Phi$, V is independent of all of its non-descendents in J, given its parents. Then $\langle \Phi, G, J \rangle$ is called a Bayesian network.

Bayesian network is proved to be able to represent any joint probability distribution. The following definition connects the joint distribution and the graphical structure of Bayesian network.

Definition 3.2.3 (d-separation) A path is said to be d-separated (dependence-separated) by a set of nodes Z if and only if it fulfill one of the following (Pearl.2000)

1. contains a chain $i \to m \to j$ or a fork $i \leftarrow m \to j$ such that the middle node m is in Z,
2. contains inverted fork (collider) $i \to m \leftarrow j$ such that the middle node m is not in Z and such that no descendant of m is in Z.

A path in graph is a set of consecutive edges in which the ending nodes each preceding edge is the starting node of next edge. d-separation is defined in terms of Bayesian theorem in the joint distribution and the Markov Condition. Combining both, we can prove the following theorem

Theorem 3.2.1 (Probabilistic implications of d-separation) If two variables U and W are d-separated by Z in a directed acyclic G, then U is independent of W

conditional on Z in every distribution compatible with G. Conversely, if U and W are not d-separated by Z in a DAG G, then U and W are dependent conditional on Z in at least one distribution compatible with G. (Pearl, 2000)

Markov Condition ensures that every conditional independence entailed by G is also in J. However, it is not sufficient to admit the correspondence of conditional dependence between J and G. Hence the following definition is needed.

Definition 3.2.3 (Faithfulness) The graph G of a Bayesian network is faithful to a joint probability distribution J over feature set $\Phi = \{\boldsymbol{X}, Y\}$ if and only if every dependence entailed by G is also present in J. A Bayesian network is faithful if there is a probability distribution J to which it is faithful.

The terminology of faithfulness by Spirtes et al. (2003) is also coined as stability (2000) by Pearl. Pearl (2000) defined stability as a property of a distribution such that the independences entailed by the distribution are not affected by the parameterization of the causal model. Specifically, stability (faithfulness) states that exact cancellation of dependences by parameterization can not occur in a faithful Bayesian network, e.g. deterministic relations. Stability is a concept introduced on causal model preference along with minimality, which is more instructive in the sense of model selection.

According to the definition of d-separation and Theorem 3.2.1, the Markov blanket of a variable Y is the set of variables that d-separate Y, which is the set of parents, children and co-parents (parents of the common children) of Y (Neapolitan,2000). Under faithfulness assumption, one can prove that Markov blanket of a variable is unique. In faithful Bayesian network, one can establish the correspondence between KJ's strongly relevance and Markov blanket. For completeness, the proofs for the two following theorems are shown based on Tsamardinos et al. (2003a).

Theorem 3.2.2 In a faithful Bayesian network, a variable $X_i \in MB(Y)$ if and only if X_i is strongly relevant to Y.

Proof: If $X_i \in MB(Y)$, from the definition of d-separation, we can see X_i cannot be d-separated from Y by $\boldsymbol{R}_i$ ($\mathbf{X}/\{X_i\}$) which implies that Y and X_i are conditionally dependent given R_i i.e. $\Pr(Y = y \mid X_i = x_i, R_i = \boldsymbol{r}_i) \neq \Pr(Y = y \mid R_i = \boldsymbol{r}_i)$. Conversely, suppose that X_i is KJ strongly relevant but not in MB(Y). First of all, we can see that $MB(Y) \subseteq \boldsymbol{R}_i$ if $X_i \notin MB(Y)$. Therefore, conditional on MB(Y) makes X_i and Y independent according to the definition of Markov blanket. Hence, X_i and Y cannot be strongly relevant which is contrary to the assumption.

The KJ weakly relevance also has relation to the MB in a faithful Bayesian network.

Theorem 3.2.3 In faithful Bayesian network, a variable X_i is weakly relevant if and only if it is not strong relevant and it is an undirected path from X_i to Y.,

Proof: If there is an undirected path from X_i to Y., we should consider two situations: 1) there are a set of colliders Z in p. Then X_i and Y are not d-separated given Z, i.e. $\Pr(Y = y \mid X_i = x_i, \boldsymbol{Z} = \boldsymbol{z}) \neq \Pr(Y = y \mid \boldsymbol{Z} = \boldsymbol{z})$ 2) there are no colliders in p, denote R_i^p as the set of variables exclude X_i and any variables in the path. We can see that $\Pr(Y = y \mid X_i = x_i, R_i^p = \boldsymbol{r}_i^p) \neq \Pr(Y = y \mid \boldsymbol{R}_i^p = \boldsymbol{r}_i^p)$. For both scenarios, if X_i is not KJ-strongly relevant, it must be weakly relevant. Conversely, if X_i is KJ weakly relevant, then there exists a set $\mathbf{Z}$ such that $\Pr(Y = y \mid X_i = x_i, \boldsymbol{Z} = \boldsymbol{z}) \neq \Pr(Y = y \mid \boldsymbol{Z} = \boldsymbol{z})$. If there is no path between X_i and Y, then they are d-separated by any set i.e. conditional independent given any set, contrary to the above.

The connection between relevance and Markov blanket (Theorem 3.2.2 and 3.2.3) was generalized to the class of strictly positive distributions which is a wider class of distribution than DAG-faithful ones by Nilsson et al. (2007). Their work also provides solid theoretical background for algorithms aiming at identifying MB for feature selection. The equivalence of MB and strongly relevance in strictly positive distribution allows MB learning to be meaningful when the underlying distribution is not faithful (various real work network has been proved to be non-faithful, e.g. networks with deterministic relationships between variables). Nevertheless, there are data that violate the assumptions of strictly positive distribution e.g. noise-free data (where $f(x)>0$ is violated), such as inference of logic propositions.

For arbitrary distributions, Yu and Liu (2004) suggested a framework using a concept called approximate Markov blanket, in which the correlations between features and target are assessed for relevance, and correlations between features are evaluated in the meantime to eliminate KJ weakly relevant but redundant features. Redundancy exists when a feature X_i is filtering out by another feature X_j (approximate Markov blanket) i.e. if and only if X_j is more correlated to the target than X_i, and the correlations between the two features is larger than that between X_i and the target. However, there is no established theoretical argument to verify the consistency of such framework yet.

For the task of prediction, Markov Blanket is the minimal set of features that are needed for Bayes classifier to produce optimal decisions if data is drawn from strictly positive distribution (Nilsson et al., 2007). Similarly, Markov Blanket (or the smallest among all MB when the data is not faithful) is solution to feature selection problem when a classifier that can approximate any probability distribution with mean-square loss metric is used (Tsamardinos et al., 2003a). MB captures all the information of the posterior distribution of the target such that the target is conditionally independent of other variables in various classes of distribution (e.g. strictly positive distribution). All features in MB are needed to generate such posterior probability of the target at either state given the features. The advantage of producing such conditional probability is that one can use it in decision theory to calculate expected utility and in machine learning to evaluate the power the classifier with AUC. However, when only the class label corresponding to (0/1-Loss) is of interest (whether the posterior probability to

one class is larger 1/2), only some of the features in MB are required or even features that do not belong to MB should be needed (Tsamardinos et al, 2003a).

3.3 Causality

Recent works have been focus on exploring the possibility of feature selection in causal discovery (Tillman and Spirtes, 2008). A feature that is relevant in prediction is not necessarily important and stable under manipulation/intervention (policy change in economics) of which the joint distribution of the data along is not sufficient to capture all the information. Causal Bayesian network is a useful concept for causal feature selection.

Definition 3.3.1 (Causal Bayesian network)
Let P be a joint probability distribution on a set of variables $\Phi = \{\mathbf{X}, Y\}$, and let P_v denote the distribution resulting from the intervention do(**V**=**v**) that sets a subset of variables **V** to constants **v**. Denote by P_* the set of all interventional distribution P_v where $V \subseteq \Phi$ including P, which represent no intervention. A DAG G is said to be a causal Bayesian network compatible with P_* if and only if the following three conditions hold for every $P_v \in P_*$

(i) P_v is Markov relative to G (as the Markov condition in Bayesian network)

(ii) $\Pr(U_i = u_i \mid do(\boldsymbol{V} = \boldsymbol{v})) = 1$ for all $U_i \in V$ whenever u_i is consistent with V=v

(iii) $\Pr(U_i = u_i \mid PA_i = pa_i, do(\boldsymbol{V} = \boldsymbol{v})) = \Pr(U_i = u_i \mid PA_i = pa_i)$ for all $U_i \notin V$ whenever pa_i is consistent with **V**=**v**, where PA_i denotes the parent of u_i. (Based on Pearl, 2000)

The notion that do($\boldsymbol{V}$=$\boldsymbol{v}$) is regarded as intervention or an action, which is intrinsically different from observation of $\boldsymbol{V}$=$\boldsymbol{v}$. The effect of the former is via ordinary Bayesian conditioning on the posterior distribution, while the latter alter the distribution factorization by truncating any conditional probability that **V** is not conditional on, which is related to the following properties,

i) For all i's, $\Pr(U_i = u_i \mid do(\boldsymbol{PA_i} = \boldsymbol{pa_i})) = \Pr(U_i = u_i \mid PA_i = pa_i)$

ii) For all i and for every subset S of variables disjoint of Tsamardinos $\{U_i, PA_i\}$, $\Pr(U_i = u_i \mid do(\boldsymbol{PA_i} = \boldsymbol{pa_i}, S = s)) = \Pr(U_i = u_i \mid do(\boldsymbol{PA_i} = \boldsymbol{pa_i}))$.

The first property states that intervention of a parent is equivalent to conditioning on. The second property states that once the direct causes are controlled, no other interventions on other variables will affect the probability of U_i.

Though the class of distribution that causal Bayesian network can describe could be narrower incorporating stronger assumptions, the usefulness and implication of causal Bayesian network is essential in designing feature selection algorithm in practice. Under intervention, features that are relevant can become non-informative, even for features in Markov blanket if the feature selection is conducted before intervention. For instance, when the children of the target variable are manipulated to set to a fix value, no features other than the parents are relevant to target. Parents are the only features that are invariantly relevant under intervention. Therefore, for

feature selection on data distribution where interventions exist (policies changes, controlled experiments), causal structure learning (differentiating parents and children etc.) is important to obtain stable results, where identifying MB along is not sufficient.

Chapter 4

Feature Selection Methods

In this chapter, several feature selection methods are discussed in more details. A feature selection method is an integrated entity of optimization, searching, and assessment. Before comparing their performances, it is crucial for one to understand the design of those algorithms.

4.1 Filters

There are a wide range of filters can be applied for feature selection as described in Chapter 2. Here we choose two filter criterions for comparisons: t-test statistic and mutual information.

4.1.1 T-test statistic

T-test statistic (Snedecorand and Cochran, 1989) is defined as follows to measure the weighted mean differences for feature X_i between the two classes of target Y.

$$t(X_i, Y) = \frac{\bar{X}_{i+} - \bar{X}_{i-}}{\sqrt{s_{i+}/m_+ + s_{i-}/m_-}} \quad (4.1)$$

where $m_{\pm}$ is the number of the samples in class ± 1 respectively, $\bar{X}_{i\pm}$ and $s_{i\pm}$ denote the sample mean and sample standard deviation of X_i for each class of Y . The statistic above has t-distribution.

With t-test statistic, we take the mean differences between classes as a measure of relevance between X_i and Y. We prefer t-test statistic over similar correlation based filters in that it provides a way for significance test (unlike Fisher's criterion) and it is more suitable for classification problems (unlike Pearson's coefficient which is better for regression).

4.1.2 Mutual Information

Mutual information (Shannon et al, 1949) criterion is derived based on information theory to measure the amount of information gained for Y given X_i. It is equivalent to KL-divergence and Information Gain (IG), which defined as

$$MI(X_i, Y) = -\sum_{j=1}^{h}\sum_{k=1}^{l} \Pr(X_i = x_{i_k}, Y = y_j) \log_2 \frac{\Pr(X = x_{i_k}, Y = y_j)}{\Pr(Y = y_j)\Pr(X = x_{i_k})} \quad (4.2)$$

where h and l are the number of classes (states) of Y and X_i

It is straight forward to estimate mutual information for discrete features and target. For continuous features and target, they are usually discretized into several intervals based on some criterions or use kernel probability estimation method to approximate the probabilities in the definition. Generally, to make use of the power of state-of-art classifiers to reduce bias and variance, the probabilities are estimated from the output of a classifier (e.g. SVM) over the training data. For multi-valued features (or multi-level discretizations), several information criterions based on MI exist, for instance, Information gain ratio (IGR, Quinlan, 1993) which reduces biases.

4.2 Naïve Bayesian learning

Naïve Bayesian (NB) classifier is based on the Bayesian rules and aims to maximize the likelihood function. The classifier is naïve in the sense that it treats every feature as independent components when constructing the likelihood function and the conditional distributions are chosen as priors from simple distributions, e.g. Gaussian Mixtures. When there are dependencies between features, naïve Bayesian classifier tends to perform worse than more sophisticated models which take into account of the dependencies of the features, such as SVM. However, it is suggested by many authors that naïve Bayesian classifier will perform equal well in some situations (Bishop, 1995). The reason is that the simplified assumption of naïve Bayesian classifier actually mitigates the effect of overfitting, i.e. there is a tradeoff between bias and variance, where naïve Bayesian sacrifices a somewhat larger bias for small variance. In many of the application, naïve Bayesian classifier is used as the baseline algorithm for evaluation of other classifiers.

There are various ways to applied NB classifier in feature selection. First of all, one can use NB as a filter criterion to rank the features based on each feature's prediction performance with NB. Secondly, it is easy to use NB as a black box wrapper to conduct feature subset selection. It is worth noticing that while in the model of NB, independences between features are assumed, the procedure of subset selection takes into account of the mutual information between features. Finally, since NB is a weak learner, it is suitable to be use as ensemble learners, and rank the features on the bootstrapped data set, and evaluate the importance according to the frequencies.

4.3 Decision Trees

Decision tree is a kind of predictive model with tree structure predictors. There exist tree learning algorithms with many variants regarding prediction accuracy, time efficiency and generalization ability.

4.3.1 ID3, CART and MARS

Quinlan developed ID3 (Iterative Dichotomiser 3 s, 1986) and its extension C4.5 and C5, which is based on Occam's razor with heuristic to produce the smallest trees. Quinlan's algorithms perform feature selection implicitly during learning based on

information gain, which is actually a filter that suits well to the underlying algorithms. While Quinlan's algorithms only deal with categorical target, CART (classification and regression tree) algorithm is suitable for both classification and regression with similar implicit feature selection based on a filter score Gini impurity (related to squared probabilities of membership for each target category in the node). MARS (multivariate adaptive regression splines) uses the similar selection criterion as CART.

4.3.2 Random Forest

Random Forest (RF) is another variant of decision trees. The basic idea of RF is to grow multiple no-pruning classification trees on bootstrapped data, with a fixed number of randomly selected variables at each node splitting (the variable with best split is used). The decision is then based on the votes of trees grown. The advantage of random forest is that it utilizes the bootstrap technique to maximize the usage of the data, and gives superior performance in terms of balance bias and variance in many real world data. The extra randomness introduced by bootstrapping and random selected feature subsets effectively reduces the variance of RF in practical problems.

According to Breiman (2001), there are two ways to perform feature selection with random forest. The first is based on a 'variable importance' score calculated as follows: for each variable, permute its values in out of bag (oob) data. Then subtract the number of correct votes for the permuted oob data from the number corrected votes in the original oob data. The average of this number over all the trees is the raw importance score for the variable. Oob data is usually one third of the bootstrapped data used in every tree construction to compute an error estimate. A z-score can compute as normalized raw score across trees, assuming there are no correlations between trees. By normality assumption, significance levels for z-scores follow. One may also calculate a local importance score for a variable to a specific observation in a similar fashion. This procedure formulates in the form of leave-one-out backward selection, which is shown to be consistent for strictly positive distribution (Nilsson et al.). The second way to compute feature importance is via a Gini importance score that is calculated as the sum of decreasing gini impurity of the two descendents nodes to the parent node. This importance score can be computed faster and provide corresponding results with the permutation measure. Diaz-Uriarte and Alvarez de Andrez (2006) conducted a set of experiments in comparing the performance of RF-based feature selection with other methods, and showed that RF-based feature selection is comparable to other state-of-art algorithms.

4.4 Recursive Feature Elimination and l_2 norm penalization

Recursive feature elimination (RFE) is a technique that can be incorporated with a class of algorithms where the weights of the features can be viewed as importance of the features to prediction. At every iteration, a specified fraction of features are eliminated based on the ranking of their weight, until the required number of features are left or the prediction errors do not decrease. RFE combined with support vector

machine (SVM) is first proposed by Guyon and Weston et al. (2003) to study gene microarray where the number of features is up-to thousands. The result was compared with Golub's (1999) univariate ranking methods, and showed that features selected by RFE-SVM is superior to Golub's regardless of the classifiers used. In the following, we introduce the version of SVM that will be use in our experiment and other RFE algorithms.

4.4.1 RFE-SVM

The original SVM algorithm is proposed as a large margin classifier by Vapnik in 1963. SVM is designed to reduce the expected generalization risk (similar to prediction error) as it is defined in learning theory. Since its introduction, SVM has shown to be of great power in the area of machine learning and feature selection (Guyon et al., 2003). In this report, we use the soft-margin linear SVM (Cortes and Vapnik, 1995). It adds flexibility to the original SVM by incorporate slack variables aiming to learn a separation hyper-plane with better generalization, which is the following optimization problem

$$\min_{w} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\xi^{(i)} \quad \text{subject to } y^{(i)}(\boldsymbol{w}.\boldsymbol{x}^{(i)} - b) \geq 1 - \xi^{(i)},\ \forall \xi^{(i)} \geq 0, 1 \leq i \leq m \quad (4.3)$$

where $\boldsymbol{w}$ is a weighting vector, and $\xi^{(i)}$'s are slack variables to allow misclassification for datum $\boldsymbol{x}^{(i)}$, C is constant to control the tradeoff between margin and penalization on misclassification. The decision rule for a new datum x' is that if $D(\boldsymbol{x}') = \boldsymbol{w}.\boldsymbol{x}' - b \geq 1$, then y' = 1; otherwise if $D(\boldsymbol{x}') = \boldsymbol{w}.\boldsymbol{x}' - b \leq -1$, then y'=-1. Data that do not satisfy the rules above, will be either regarded as rejection (can not be classified by the SVM) or classified by replacing 1 and -1 in the D($\boldsymbol{x}$) with 0, respectively.

Geometrically, the inverse of $\|\boldsymbol{w}\|^2$ is equal to the distance between the two hyper planes that separate the two classes. In the dual form of the minimization problem, we can see that *w* is a linear combination of feature vector weighted by the product of class label (y) and the vector weight calculated from the dual. For feature selection, the importance of $\boldsymbol{w}_i^2$'s is that they are proportional to the gradient effect of a feature to the bounds of prediction errors (Guyon et al., 2003), which in turn justified the RFE scheme by selecting features that have large effects on the generalization error bound. On the other hand, the parameter C is useful in controlling the sparsity of the SVM model, where small C tends to shrink the model into fewer support vectors. Boser et al. (1992) suggested a non-linear version of SVM with kernel tricks by replacing the inner product by a non-linear kernel function in the dual form, which can be used to classify data that is not linear separable.

It is shown in some study that, when applying RFE-SVM, selecting all available features actually provides the best prediction results even though some features are known to not relevant to the target (Guyon et al., 2003). This can be understood in the sense that, for data with noise, SVM is able to extract the information of the noise from irrelevant features (e.g. background noise caused by measurement methods) and incorporate such information to improve the prediction.

If the causality between features and target is of primary concern, the RFE-SVM selection algorithm is sometimes not informative. Hardin et al. (2004) studied SVM-weighted-based methods in learning causally relevant features theoretically and showed that: i) the irrelevant variables will be zero-weighted by a linear SVM in sample limit ii) zeros weights may be assigned to causally relevant features and non-zero weights may be assigned to non-causally relevant features. On the other hand, Statnikov et al. (2006) experimented with simulated data and pointed out that in practice SVM-weight-based methods might perform even worse in differentiating causally-relevant, non-causally relevant and irrelevant features. One of intrinsic reason for this is that in practice, features that are needed to construct large margin (in SVM) do not necessarily have correspondence to causality. In all, they suggested that one should always be careful when trying to interpret causally the features selected by SVM-weighted based algorithms and algorithms that taking into account of causal structures are indeed needed. Nevertheless, revised SVM-based feature selection algorithms have gained success in the causal feature selection competition, which focused on discovering the causal structures from data that incorporate manipulation. However, recent study showed that such success is related to the data generation methods and most importantly the generalization ability of SVM overcomes the small bias when omitting the causal structure (Tillman and Spirtes, 2008).

4.4.2 RFE-Penalized logistic regression

Logistic regression is a generalized linear model that predicts the probability of the class label by fitting the data to a logistic curve. The decision for the class label is then made according to the probability. l_2-penalized logistic regression (l_2-PLR) adds a l_2-norm penalization on the weights of the features in the original logistic negative regression likelihood, which can be defined as the following optimization problem:

$$\min \sum_{i=1}^{m} -y^{(i)} g(\boldsymbol{x}^{(i)}) + \ln(1 + e^{g(\boldsymbol{x}^{(i)})}) + \lambda \left\| \boldsymbol{w} \right\|^2 \quad \text{where} \quad g(\boldsymbol{x}) = b_0 + \boldsymbol{w}.\boldsymbol{x} \quad (4.4)$$

Note that to apply logistic regression as above, the target should be coded as {0,1} instead of {-1,1}, which is trivial in implementation.

l_2-PLR combined with RFE is also applied in gene selection for gene microarray classification problems (Zhu and Hastie, 2002). PLR is a generalized linear model with logistic function and an l_2 norm penalization on the weights. Zhu and Hastie (2002) investigated the performance on several sets of gene microarray data, and showed that RFE-PLR algorithm tend to select fewer features and comparable prediction accuracy to RFE-SVM in those datasets. Moreover, it also provides the probability of certain gene patterns belonging to particular class (probability close to 1 does not necessarily indicate strong evidence of inclusion into the particular class). The reason for the differences of SVM and PLR, as explained by Zhu and Hastie is that the dataset under study is usually linear separable by small set of genes. In this situation, the solution of RFE-PLR is more sensitive to the regularization parameter than SVM-PLR, thus easier to detect such relevant genes. There are also non-linear versions of PLR called kernel-PLR (Keerthi et al., 2005).

4.5 ℓ_1-norm penalization

Both PLR and SVM applied l_2-norm on the regularization term instead of l_1-norm. Feature selection via l_1-norm penalization possesses different statistical properties. (Note that l_1-norm penalization can also be incorporated in RFE)

4.5.1 Lasso

Lasso（least absolute shrinkage and selection operator）was first proposed by Tibshirani (1996) to incorporate l_1-norm penalization in ordinary least square regression, and has been generalized to general convex loss function such as exponential loss in logistic regression combined with the ideas of boosting (Zhao and Yu, 2007). 1-norm SVM was also proposed by (Zhu et al., 2004) in the light of lasso. The intrinsic differences between l_1-norm and l_2-norm regularization in the paradigm of Bayesian inference are the distinct prior on the parameters of the models. Specifically, l_1-norm penalty corresponds to double-exponential prior, whereas l_2-norm relates to Gaussian prior. One advantage of l_1 norm penalty is that it tends to regulate the algorithms to produce feature weights that equal to 0 due to the heavier tails of double-exponential prior. Therefore, feature selection via l_1-norm is performed automatically during the optimization without the need to combine with RFE (Zhu and Hastie, 2002).

4.5.2 l_1-penalized logistic regression

l_1-penalized logistic regression uses l_1-norm to penalized the weights $\boldsymbol{w}$ in (4.4). As lasso, l_1-PLR tends to produce sparser model than l_2-norm PLR. It is applied both in feature selection (Genkin et al., 2004) and Bayesian network structure learning or Markov blanket (Schmidt et al., 2007, Lee et al., 2007, Wainwright et al., 2006). The advantage of l_1-PLR over lasso is that it has the ability to capture more complex relations between features with the logistic function, while retains the sparse property of lasso. It is also shown by (Ng. 2004) that l_1-PLR is more sample efficient in the sense that the number of samples for l_1-PLR to learn a 'good' model grows only logarithmically in the number of irrelevant features while for l_2-PLR, SVM etc. based on l_2-norm penalization, the number of samples required grows at least linearly to the number of irrelevant features. This could be a support for the use of l_1-PLR or other l_1-norm penalization methods when the number of irrelevant features is large.

As far as prediction accuracy is concerned, the performance of l_1-norm and l_2 norm regularization relate to the sparsity of "true" mechanism generating the data. The sparsity of the true generating function is related to the signal to noise ratio (SNR) and the size of basis function dictionary to capture the true function. With data generated from generalized linear model, Tibshirani (1996) pointed out that lasso produces the best result when the models have small to moderate number of moderate-sized effects, following by l_2-norm penalized regression. While there is small number of large effects univariate subset selection outperform lasso, and l_2

norm does poorly. l_2-norm penalty performs the best when there are large number of small effects. l_1-norm is preferred by several authors (Zhu et al, 2004, Friedman et al 2004) in the sense that it produces better prediction when the underlying data are generated by a sparse model (e.g. when there are only a small number of coefficients are not zero. In this case, l_1-norm penalty will perform better even when the actual distribution of the coefficients is Gaussian in the simulation). For dense model, neither l_1- nor l_2-norm regularization will perform well according to experiments where the data to feature ratio is too small, which can be due to the curse of dimensionality. Nevertheless, l_1-norm fails when there are a large number of small effects in which case it is suggested that l_1-norm sacrifices prediction errors for sparsity. Therefore, one could see that l_1 penalty and l_2 penalty actually to some extent detect the characteristics of the true function generating the data. When conducting a feature selection task, it is of interest to apply both l_1 and l_2 to investigate the underlying properties of the generating function.

4.6 Markov blanket discovery

As previously discussed, Markov blanket of the target is a very informative subset both theoretically and in the sense of prediction. Due to the importance of MB of the target, denoting as MB(Y), many algorithms have been designed to locate this subset. Even though global Bayesian network learning algorithm can be an intermediate step to determine the MB(Y), it generally lacks computational efficiency and statistical significance when the number of features is large. In the following, we focus mainly on local structure learning algorithms.

4.6.1 Koller and Sahami's algorithm

The first algorithm designed to learn Markov blanket of a variable is proposed by Koller and Sahami (1997). Koller and Sahami aimed to approximate the Markov blanket with iterative elimination. Their algorithm starts with the full subset. At each iteration, a feature is eliminated when its approximate MB can be found in the current subset. Such procedure ensures that the elimination of a feature will not affect the elimination of another feature afterwards. Koller and Sahami pointed out that this algorithm can only approximate MB due to the coarse measure of conditional independence via cross-entropy, and the large conditioning set which require exponentially increase number of sample to obtain reliable estimate. This algorithm is highly time efficient but is not scalable to data with large number of features due to the lack of power with large conditional sets.

4.6.2 IAMB, MMMB and HITON-MB

A set of local learning algorithms for MB were proposed mainly by a group in Vanderbilt University, including incremental association Markov blanket algorithm (IAMB) , the max-min Markov boundary algorithm (MMMB) and HITON-MB. The

design and properties of the three algorithms are discussed shortly in the following:

IAMB (Aliferis et al., 2002a) differs from KS's algorithm in that instead of pruning features that have approximate MB, IAMB starts at first greedily include features that are in MB(Y) through reliable independence test, and then tries to remove false positive with an additional step. In IAMB, the independence tests are run conditional on the current candidate MB, which suggest that the size of samples required is exponential to the size of the true MB.

MMMB (Tsamardinos et al., 2003c) takes into account of the data inefficiency in IAMB by divide the algorithms into two parts. The first part is to identify the parents and the children of the target Y with max-min parents and children (MMPC) algorithm. The second part is to search for the parent of common child by investigating the conditional independence between the target Y and the parents and children of the features output by MMPC. In doing so, MMMB circumvents the data inefficiency problem in IAMB, and depends only on the topological connectivity of the underlying network, since the conditional set is always bounded by the number of features connected to the target i.e. parents and children

HITON-MB (Aliferis et al., 2003b) is similar to MMMB with a modification in the conditional independence in HITON-PC (MMPC) that the conditional feature set (current PC in MMPC) is replace by empty set to further improve the data efficiency.

All these three algorithms are proved to be completely identified the MB with possible false positives under the following assumption:

i) the underlying distribution is faithful to a DAG.
ii) samples are i.i.d.
iii) the independence test is reliable given the sample size.

The second assumption is generally satisfied and assumed by other methods. As for the faithfulness assumption, the authors pointed out that the faithfulness assumption is the key to locate the MB efficiently and the success of all three algorithms rely on the fact that either biomedical data do not exhibit severe violations of the faithfulness assumption or such violation is mitigated by variable connectivity (missed MB variables are taken place by their proxies) or other factors. Finally, the third assumption concerned about reliability of the independence test related to both the sample size and the power of the tests. To reduce the sample requirement, the conditional set should be as small as possible in the test (MMMB and HITON-MB). On the other hand, various independence tests can be used. Chi-square independence test is used when the features are discrete, and Fisher's z test is run for the case where all but the target Y are continuous. Kernel-based independence test are consistent for any probability functions, whose estimation could introduce additional complexity into the algorithm and thus is prone to overfitting when sample size is small.

The advantage of the three local structure learning algorithms is that they overcomes the over-fitting problem and time-consuming process by discovering Markov blanket which ensures the optimal classification performance under quadratic loss (which implies optimal performance under 0/1 loss or AUC). On the other hand, due to the possible introduction of false positive, Aliferis et al. (2002a) suggested that the produced MB can be post-processed with PC or FCI algorithm (Sprites et al, 2000)

or other customized methods (e.g. the criterion of symmetry (Aliferis et al. 2002b), if a feature V is in the MB(Y) then Y should also be in MB(V).

4.6.3 PCMB

Parents and children based Markov boundary algorithm (PCMB) (Peña et al. 2007) is guaranteed to identify MB(Y) with no false positives under the same assumptions of the three algorithms above. Unlike the three above-mentioned algorithms where false positives are eliminated after MB(Y) is produced, PCMB copes with false positives in the recursive conditional independence test by incorporating more tests. The searching procedures are conducted similar to that of MMMB and HITON-MB by dividing the problems into two sub-problems but the first sub-problem is solved by GetPCD(Y) and GetPC(Y). The correctness of identifying MB(Y) is proved theoretically when the sample size goes to infinity.

4.6.4 Backward Search MB

Nilsson et al. (2007) proposed a simple backward search method based on a real-value criterion function that accounts for strong relevance, which is consistent if the underlying distribution is strictly positive. They proved the optimality of strongly relevant features for prediction as well as the correspondences between the Markov blanket of Y and strong relevant features in strictly positive distributions. The class of strictly positive distribution is larger than the class of faithful distribution. Then, they proposed a polynomial algorithm to identify strongly relevant features which is equivalent to MB(Y). The algorithm can be described in the following pseudo-code based on Nilsson et al. (2007).

1. Start with an empty set MB(Y), Pick a real-valued criterion function (pair of L and f) F.
2. for each feature X_i in X
   if $F(D_{R_i}) > F(D) + \varepsilon$
   where $\varepsilon \in (0, \eta)$ with $\eta = \min_{i \in MB(Y)}(F(R_i) - F(X))$
   then $MB(Y) = MB(Y) \cup X_i$
   endif
   endfor

The algorithm is guaranteed to output MB(Y) under the assumptions i) the underlying data is strictly positively distributed ii) samples are i.i.d. iii) the classifier f is consistent to Bayes classifier and provides consistent estimate on the ranking of subsets regarding generalization errors with L iv) the sample size goes to infinity. There are many ways to choose f that satisfied the assumption in iii) e.g. k-nearest-neighbor, SVM etc. Therefore, the algorithm is flexible in providing a framework to derive consistent wrappers and filters. Nevertheless, Nilsson et al. noted that such methods might suffer from the similar problem of large conditional sets as IAMB where an implicit left-one-out conditional dependence tests are performed.

Thus, it might not be practical to apply this simple method for small data set with high dimension. On the other hand, the theoretical correctness of this algorithm gives a solid mathematical support for the success of some feature selection algorithms such as RFE (Guyon et al. 2002). Nilsson et al. pointed out the consistency of REF can also be proved with slight modification in the proof of the backward search algorithm.

Though it is possible to choose ε for controlling false positive rate and false negative rate, there is no theoretical result on the defined upper bound for ε. The heuristic to control the parameter was not extensive discussed and simple chosen as 0 in Nilsson et al. (2007). In this report, we studied several criterions that might be informative to incorporate regarding the undefined magnitude of difference i.e. ε, which will be discussed in more details in next section.

Chapter 5

Comparative study

In this section, we would like to investigate the performance of those algorithms on synthetic data regarding prediction and causality. Synthetic data are simulated to ensure some characteristics of distributions and dependence are presented which will be used to study the possible distinct behaviors of different algorithm. In addition, we also compare those algorithms on real data to investigate the correspondence between features selected, and the usefulness of those algorithms. [1]

5.1 Data Description

5.1.1 Synthetic data

To study the potential of SVM in causal discovery, Statnikov et al. (2006) simulated two types of networks that mimic the real-world gene regulatory networks. Both networks are simple in the sense that there are generally no dependences between features (except the U in Network 2, which is correlated to all V_i's). This might not be the case for data from another domain, or even in gene regulatory networks. Nevertheless, Statnikov et al.'s idea (2006) is to prove that even for such simple networks, the weighted-based SVM fails to capture causality in a stable way. We modify their simulation slightly to gain balance data set (Network 2), and use different notations as well as incorporate different dependence structure (Network 1b).

Network 1a

The first type of network is generated as follows

1. U is a binary variable with P(U=-1) = 1/2 and P(U=1) = 1/2. U is hidden, and used to generate V only.
2. $\{V_i\}_{i=1,\ldots,N}$ are binary variables with $P(V_i= -1|U= -1) = q$ and $P(V_i=1|U=1) = q$, where q is fixed and chosen to be 0.95 for our experiment.
3. $\{W_i\}_{i=1,\ldots,M}$ are independent binary variables with $P(W_i=-1)=1/2$ and $P(W_i=1)=1/2$.
4. Y is the target variable with $P(Y=-1|V_1=-1) = 0.95$ and $P(Y=1|V_1=1) = 0.95$

In Network 1, feature V_1 is the only causally relevant to Y, while are not independent of V_1, since U is hidden from the observation, it renders all V_i's to be dependent (hidden common parent). Therefore, $\{V_i\}_{i=2,\ldots,N}$,they are non-causally relevant to Y

[1] All the experiments are implemented in MATLAB. The source codes are available upon request yubin@maths.lth.se. Several packages for algorithms were used in the experiments. See the specific section for details.

and $\{W_i\}_{i=1,\ldots,M}$ are irrelevant to Y presenting as noise. Thus, the MB of Y is only V_1. It represent a kind of gene regulatory network where a gene (U) regulating many proteins that may be strongly correlated with each other (V_i's), but only one of which (X_1) is responsible for the present of a specified disease (Y).

To incorporate higher order of Markov dependence into the network, we modify network 1a as follows

Network 1b

1. U and W_i's are generated as in Network 1a
2. V_N is a binary variable with $P(V_N=-1|U=-1) = 0.95$ and $P(V_N=1|U=1) = 0.95$, and for $\{V_i\}_{i=1,\ldots,N-1}$, $P(V_i=-1|\ V_{i+1}=-1) = q$ and $P(V_i=1|\ V_{i+1}=1) = 0.95$
3. Y is the target variable with $P(Y=-1|V_1=-1) = 0.95$ and $P(Y=1|V_1=1) = 0.95$

Therefore, in Network 1b, the MB of Y is the same as it is in Network 1b. However, all V_i with i>1 now become the ancestors of Y, while only V_1 is the parent of Y.

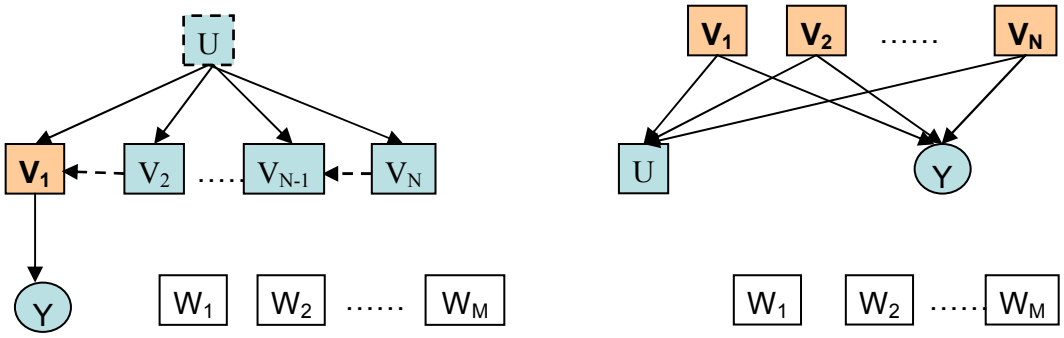

Figure 5.1 Network structures for Network 1a (left, Network 1b with also the dash arrows) and Network 2 (right)

To increase the number of strongly relevant features,

Network 2

1. $\{V_i\}_{i=1,\ldots,N}$ are independent binary variables with $P(V_i=-1) = 1/2$, and $P(V_i=1)=1/2$
2. $\{W_i\}_{i=1,\ldots,M}$ are independent binary variables with $P(W_i=-1) = 1/2$, and $P(W_i=1)=1/2$
3. U is a variable calculated as $U = \sum_{i=1}^{N} V_i / N$,i.e. the average of V_i's
4. Y is a binary variable defined as $sign(\sum_{i=1}^{n} t_i V_i - N/k)$,where t_i is uniformly distribution random variable from (0,1) and fix for all experiments. k is chosen to be 16 instead of 4 in the Statnikov et al. to ensure that the data is not extremely unbalanced when N becomes large.

In Network 2, all V_i's are causally relevant to Y (MB of Y), and U is non-causally relevant to Y. W_i are simply binary noise which are irrelevant. Network 2 is a simplified representation of gene regulatory network where a set of regulatory genes (V_i's) are mutually coordinating regulation of many genes (here U and Y).

For both networks, noise can be added by replacing k% of the data with corresponding values randomly sampled from the distribution of the generated data. In experiments of Statnikov et al. (2006), the levels of noise do not affect the results on differentiating causality but only decrease the prediction accuracy with more noise.

Since these two synthetic data sets have resemblance to the reality and appropriate simplicity, it might be of interest to test different feature selection algorithms on them.

5.1.2 Real data

Generally, synthetic data do not possess enough information to capture the realistic characteristics of data distributions from real world. Therefore, it is important to investigate the performance of feature selection methods on real data.

AML-ALL data

The AML-ALL data consists of a matrix of gene expression values obtained from micro-arrays for a number of Leukemia patients. The intention of study this data is to differentiate AML and ALL (two types of Leukemia) with the gene expression. There are in total 7129 genes measured for 72 patients which are labeled as AML or ALL. The data is split into training set with 38 samples (27 ALL and 11 AML) and test set with 34 samples (20 ALL and 14 AML). This data set is studies extensively by various authors (Golub et al., 1999, Guyon et al., 2000, Li and Yang, 2005) and is known to be linearly separable with few features.

5.2 Experiment designs

For synthetic data, we can control various properties of the generated data, such as the sample to feature ratio, the levels of noise etc. By monitoring such factors and the different networks, we can conduct sensitivity analysis of different feature selection methods. We are particular interested in studying the ability of different methods in differentiating causally relevant/ non-causally relevant and irrelevant features. The experiments for synthetic data is conducted as follows

1. For the triplet {M,N,*m*}, where *m* is the number of training samples, we simulate data with {50,50,500},{100,100,200},{100,100,50} for all the networks (approximately with sample to feature ratio of 5, 1 and 0.25) for two levels of noise 0% and 5%; All the results in the following are averaging over 30 trials.
2. Assessing the average rankings of features. For filters and feature selection methods providing weights for features, such measures can be used to evaluate whether causally relevant features are ranked higher than others. The rankings are in ascending order of the weights and are normalized to the number of features.
3. Assessing the average selection frequencies of features. For wrappers, and MB discovery methods etc. where a subset of features are output, the differences in selection frequencies provide information of the powers of differentiating different kinds of features.

Another set of experiments, are conducted to investigate the possible modification on Nilsson et al.'s backward search algorithm with synthetic data. We would like to show the intuition of our designed schemes and whether different schemes will improve the performance of the original backward search algorithm.

For real data, we follow similar evaluation procedures except that they are only

performed on the training set and additional prediction performance is assessed on test set specified.

For there are large amount plots with different combinations of sample to size ratios and noise levels, we only present figures where such factors affect the overall performance of the methods. In experiments with Network 1a and 1b, V_1 is plotted as the first variable, while $V_{2...N}$ as 2 to N_{th} variables, W's as $N+1_{th}$ to $M+N_{th}$ variables. With Network 2, V_i's are plotted as the first N variables, and U is the last variable. W's are those variables between V_i's and U.

5.3 Results

5.3.1 Synthetic data

5.3.1.1 Filters

We estimate the t-test statistic and mutual information of each feature to the target for different networks. We can see that (Figure 5.2) that both filters are insensitive to the sample to feature ratio for network 1a. (actually for all networks). It is reasonable for this observation since it is generally understood that the estimation of such univariate statistics are stable with small sample size (Guyon et al., 2003). On the other hand, noise does not affect the results on ranking at all (Figure 5.3). Interestingly, it is observed that, as far as the differentiation of causally-relevance is concerned, all the methods are not sensitive to noise.

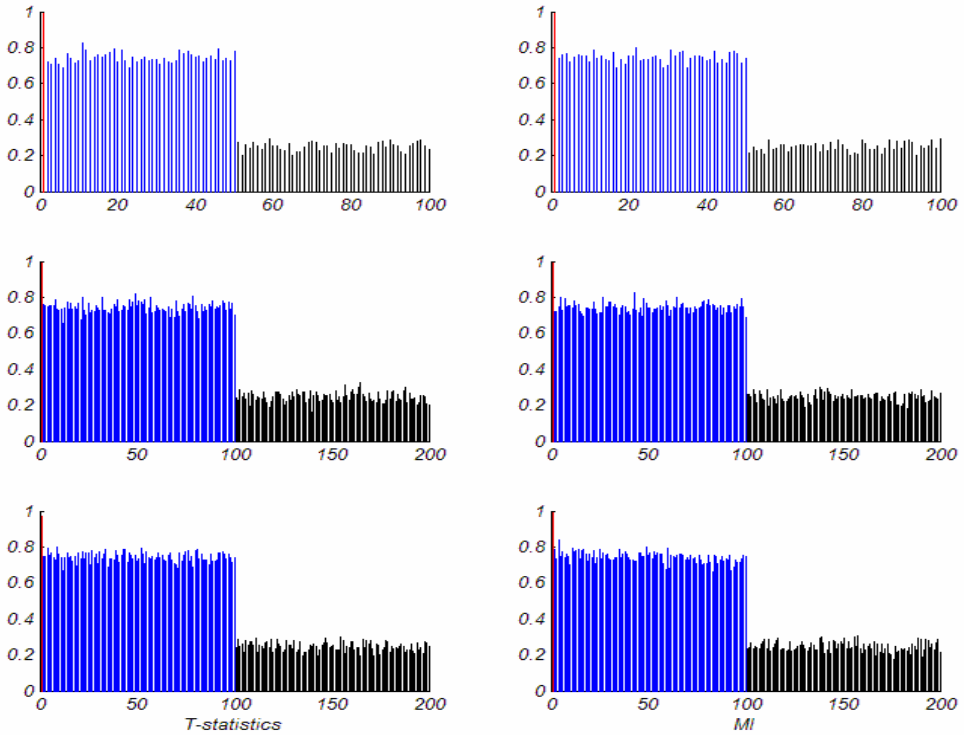

Figure 5.2. Average ranking based on t-test statistic and mutual information(with random forest) for Network 1a with 5% noise. The three rows are calculated with different number of features and sample sizes i.e. {50,50,500},{100,100,200},{100,100,50}. V's are showed as the first N(50,100) variables, while W's are the last M (50,100) variables.

When we exam the two filters across different networks, we obtain the following observations:

1. The rankings based on the two filters have strong correspondence. They behave in similar way for different kinds of features for different networks.
2. Both filters rank the features in Network 1a properly regarding their causal relevance (strong relevance), i.e. causally relevant feature V_1 always ranks the first, followed by non-causally relevant ones with clear difference in averaging ranking 60% to irrelevant ones.
3. For network 1b, both filters rank the features regarding their Markov order. As *i* becomes larger, the rankings of non-causally relevant features V_i decrease accordingly. When i > 30, the rankings of V_i have no differences than the irrelevant ones, 0.95^{30} ~= 0.20. Both filters lack power in differentiating such non-causally relevant features to irrelevant ones.
4. For Network 2, both filters fail to correctly differentiating features based on their causal relevance (Figure 5.4). Specifically, both filters rank the non-causally relevant feature i.e. U as the top feature, while the causally relevant features V_i's can have lower ranks than irrelevant features. The strong preference to U for both filters is that U have the same function form as Y (linear combination of V_i's). Closer look at the V_i's that have lower rankings, we see that they actually correspond to features that have small random weights t_i's (around 0.2). Therefore, both filters in some sense capture the linear correlations of features to the target. When the linear correlation is smaller than 0.2, both filters fail to differentiate those features from random noise.

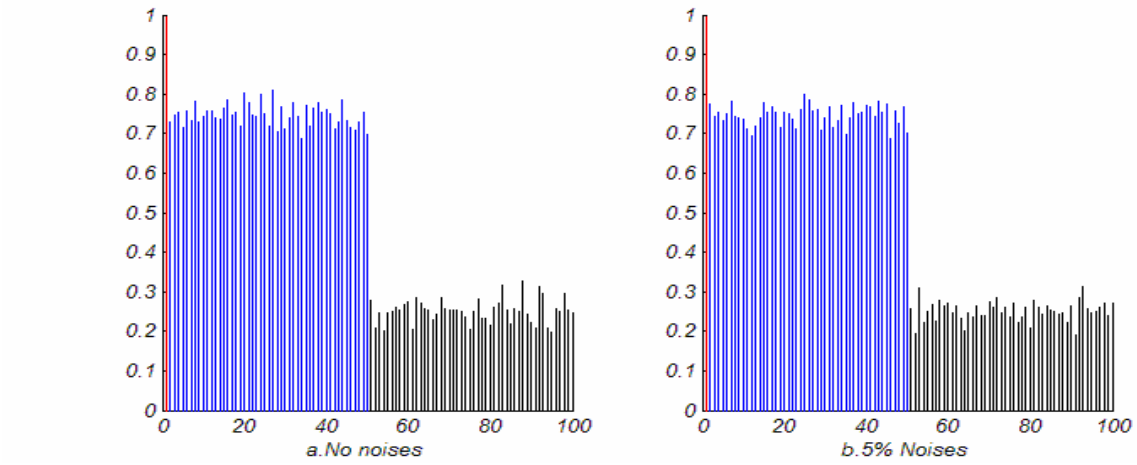

Figure 5.3 Average ranking based on mutual information (random forest) for Network 1a two levels of noise a) no noise b) 5% noise with {50,50,500}. V's are shown as the first 50 variables, while W_i's are last 50 variables.

From the experiments, we can see that the two filters have limited success in differentiating the three kinds of features (Network 1a) and they are invariant to small sample to feature ratio. However, such univariate filters can not detect the dependence between features, thus have no power in discovering conditional independence e.g. in Network 2, U is independent of Y given V_i's. More sophisticated methods are needed.

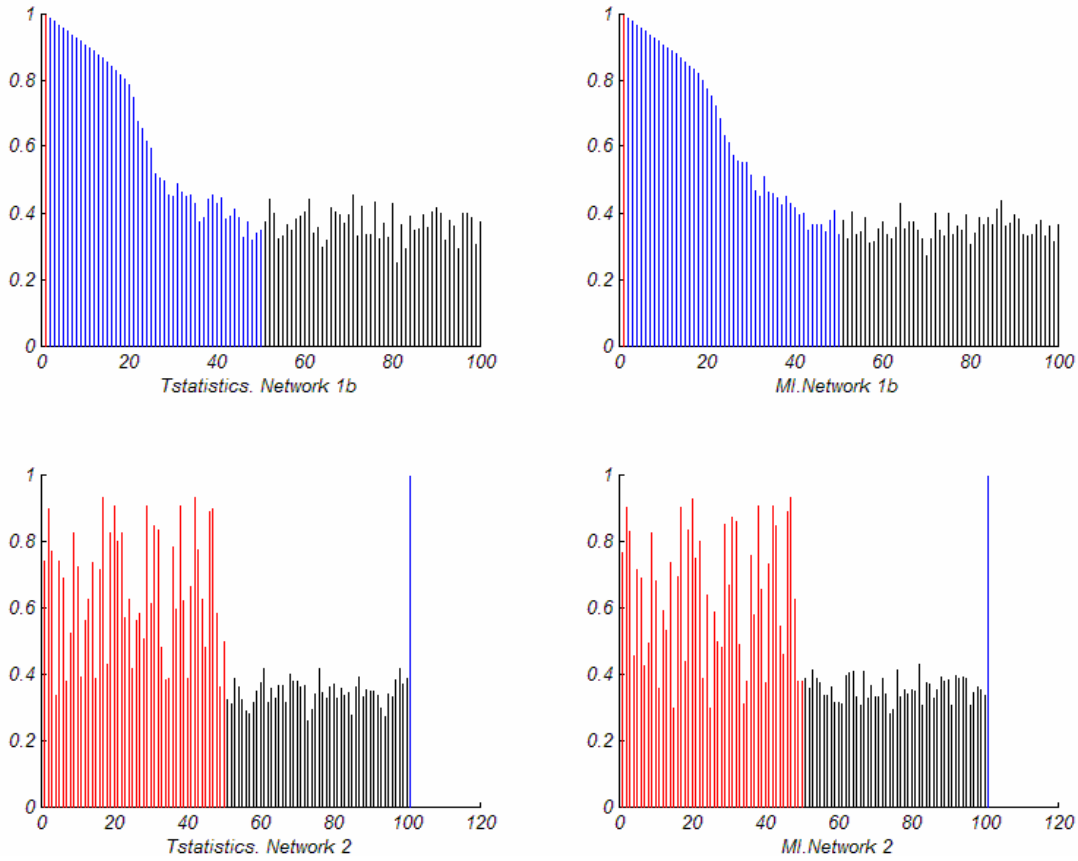

Figure 5.4 Average ranking based on t-test statistic and mutual information for network 1b and Network 2 with {50,50,500}, no noise. V's are showed as the first 50 variables, while W's are the next 50 variables (51-100). For Network 2, U is the last variable.

5.3.1.2 Ranking based on weights of SVM and l_2-PLR

Instead of running RFE, we investigate the rankings of features based on the weights of linear SVM and l_2-PLR. We can assess the weights by learning the SVM l_2-PLR on the full set of features. We follow the Statnikov et al.'s study to investigate how different levels of penalization on the norm will affect the results. In the experiments, we vary the C and lambda in {0.001, 1, 1000}

- **C and Lambda control the sparsity of the model, but has limited ability in differentiating the strongness of relevance of different features.** Generally, SVM with larger C tends to produce denser models, while in l_2-PLR, larger lambda actually favors sparser models (since in l_2-PLR lambda is assigned to the norm, while in SVM C is assigned to the inseparability term). This is observed for all networks. In Network 1b (Figure 5.5, as C increases (lambda decreases), fewer and fewer non-causally relevant features ($V_{2...N}$) have higher ranks than irrelevant ones, which is similar to Network 1a. On the other hand, both SVM and l_2-PLR can assign higher weights to non-causally relevant features to causally relevant ones in Network 2. Such observation is invariant to the change of C and lambda. Statnikov et al. pointed out theoretically that for such generation mechanism, SVM will always assign higher weight to U than others in Network 2 in order to produce a large margin.

- **The rankings based on SVM and l_2-PLR weights are sensitive to sample to feature ratios.** As we exam the results on different feature to sample ratio, we can see that the relative rankings between different kind of features are extremely volatile. i) For the data are sufficient {50,50,500}, the rankings of features with both methods are very similar to the rankings of the two filters tested regardless of C and lambda. ii) When data are insufficient, ({100,100,100}, both causally relevant features (Network 2) and non-causally relevant features (Network 1a/1b) can have lower (or the same) averaging rankings than irrelevant features (Figure 5.6).

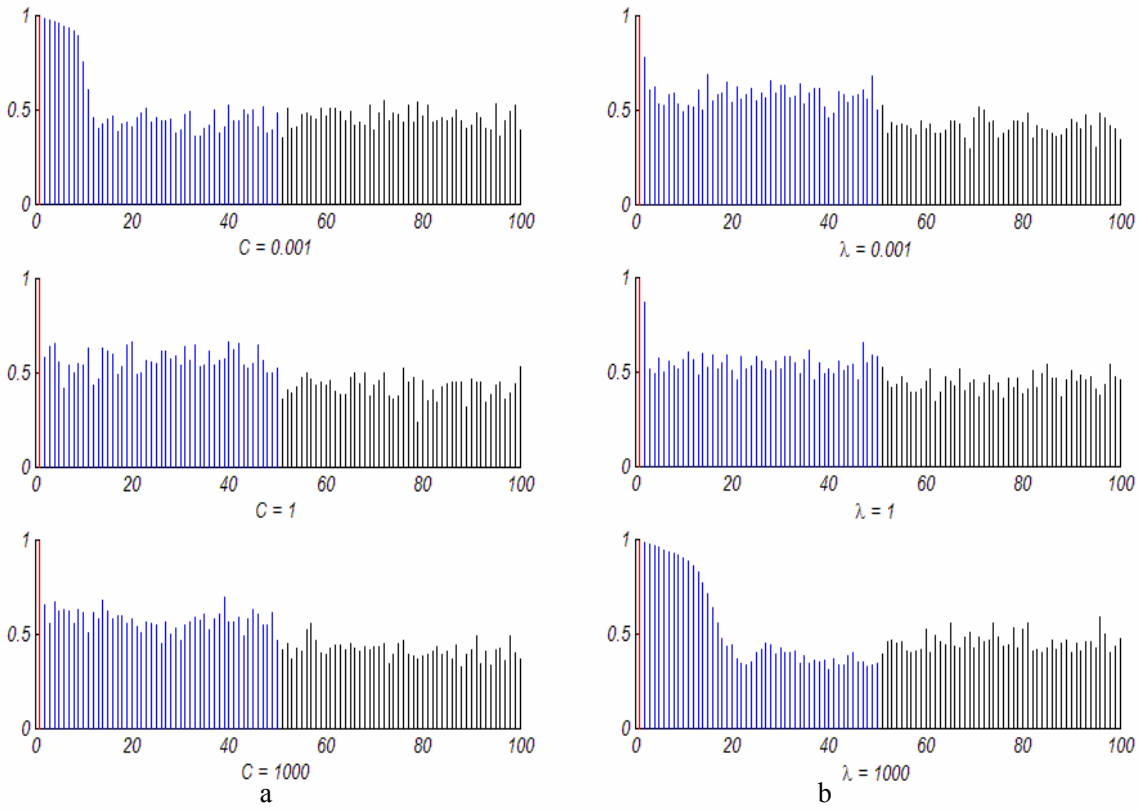

Figure 5.5: Average ranking based on SVM (a) and l_2-PLR (b) weights for Network 1b with {50,50,500} and no noise with different levels of penalization. Causally relevant features is V_1, and non-causally relevant feature(s) are shown with V_2-V_{50}, while non-relevant features are W's (51-100).

The first observation is the same as Statnikov et al.'s work and in turn suggests that one should be cautious whenever feature selection are based on SVM and l_2-PLR weights (e.g. RFE). One way to regulate the selection of C and lambda is to choose model parameters for both models via cross-validation on the training set and select the ones that yield the best prediction performance. With 3-fold cross-validation for smallest CV-errors, the C parameter for SVM is tuned to 0.001~0.1, while for l_2-PLR, the penalization choose 1000 or larger (for network 2) for different networks. For Network 1a and Network 1b, lambda equals to 1000, gives the best prediction performance and the best lambda for Network 2 seems to be unbounded. With best C

and lambda and sufficient data, SVM and l_2-PLR produce the rankings similar to filters, except that the rankings of non-causally relevant features in Network 1b decrease faster regarding their Markov orders.

Aside from the sensitivity to C and lambda, the instability to sample to feature ratio seems to more problematic since low sample to feature ratio is the typical situation in gene micro-array analysis. For such simple networks, both methods fail to differentiate irrelevant feature from non-causally relevant ones (Network 1a), or even causally relevant ones (Network 2), when sample size is small. This suggests that why filters are of interest to try when the data at hand is small in size and large in dimensions. The success of RFE-SVM on micro-array data with very high prediction accuracy (Guyon et al., 2001) where the sample to feature ratio is as small as approximately 1/200 (38/7129), does not necessarily imply the power of such method using in discovering MB of the target.

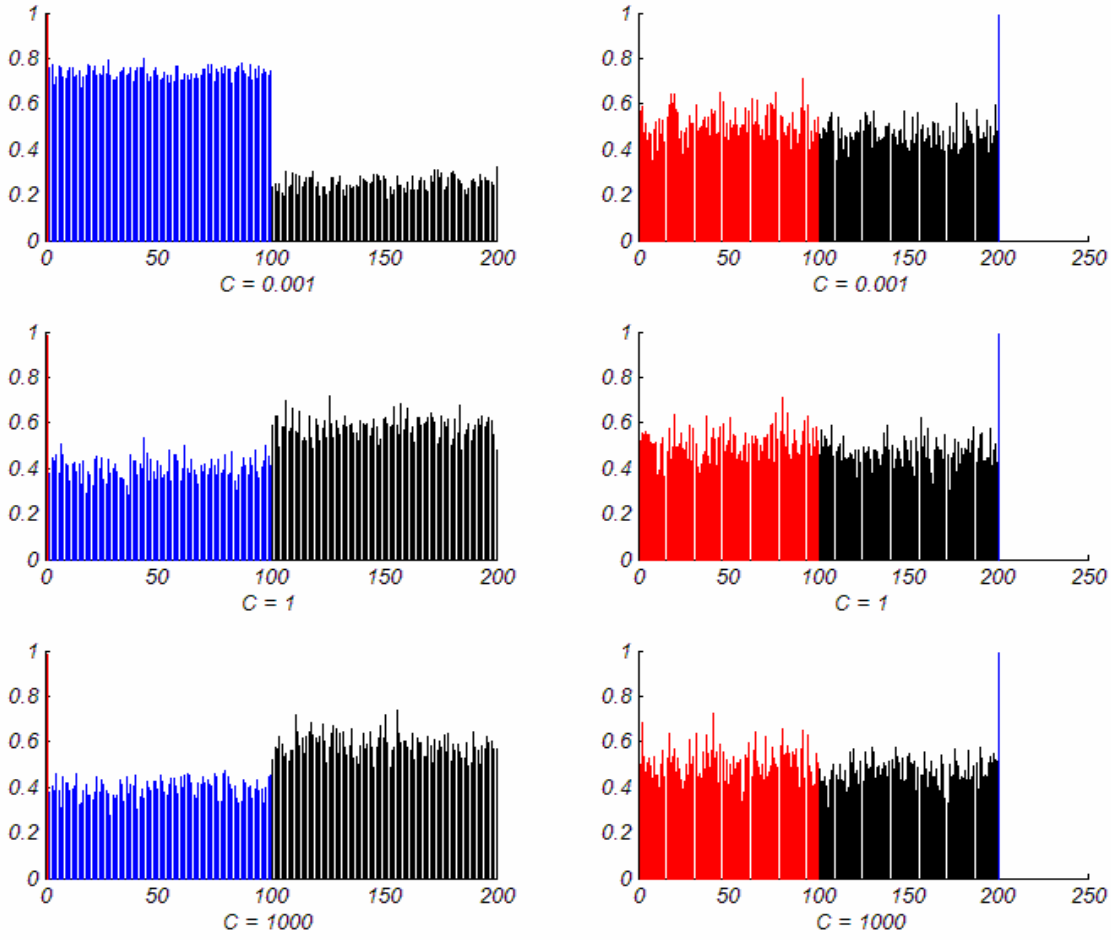

Figure 5.5 Average ranking based on SVM weights for Network 1a (left) and Network 2 (right) with small sample size ({100,100,100}) and no noise. The rows corresponds to different levels of penalization of C. V's are showed as the first 100 variables, while W's are the next 100 variables (101-200). For Network 2, U is the last variable.

5.3.1.3 l_1 penalized logistic regression

Several authors have studied using l_1 penalized logistic regression (l_1-PLR) to locate the Markov Blanket of a variable which are generally used in local Bayesian structure

learning (Lee et al., 2006, Schmidt and Murphy, 2006). On the other hand, Ng (2004) proved that l_1-PLR is more data efficient to l_2-PLR, in the sense that for the equivalent prediction performance, the former requires sample size that grows logarithmically to the number of irrelevant features compared to the latter which requires linearly growing number of samples to the irrelevant features. Here, we run l_1-PLR on the data, and treat those features that have non-zeros weights (l_1-PLR has strong penalty on small effects, which are shrink to zeros) as the MB of Y and plot the frequency of each feature being selected. We use the Schmidt's implementation of GeneralL1[1] (Schmidt et al., 2007). The default level of penalization (lambda) is 50, where larger lambda amounts to preference to sparsity. For the given networks, when lambda exceeds 20, the weights for all features become zeros. Therefore, we investigate the effects of lambda's by learning the models with lambda = {1,5,10}

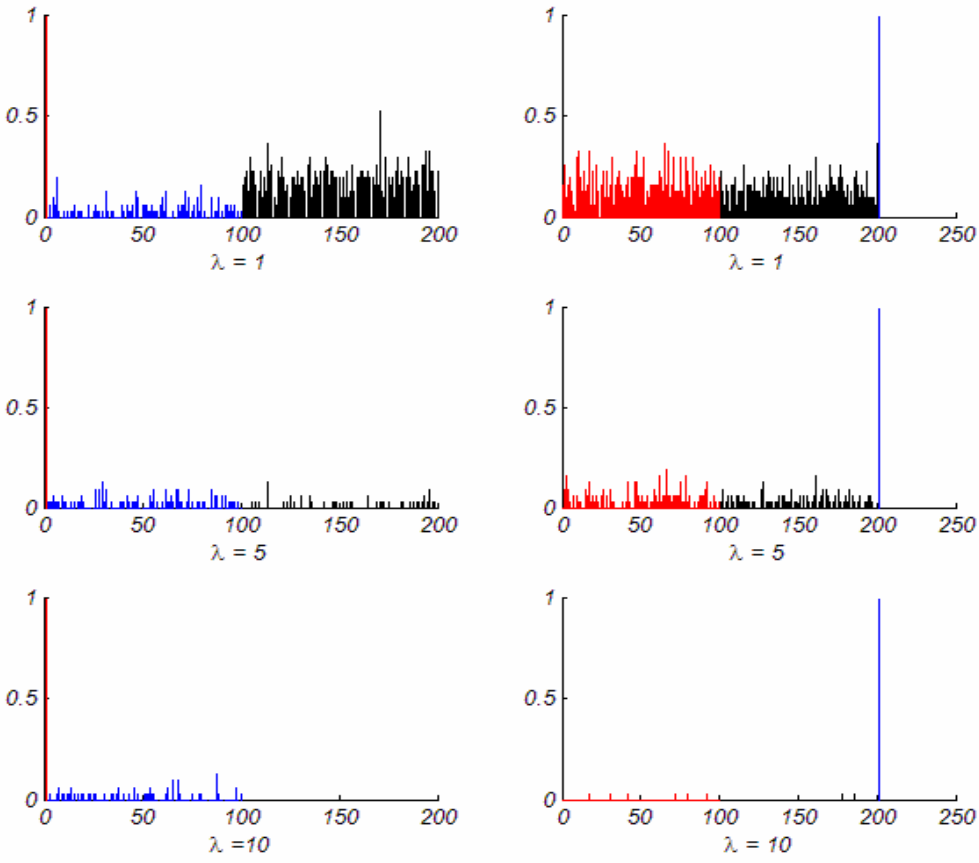

Figure 5.6 Selection frequencies of features by 1-norm Logistic regression for Network 1a (left) and Network 2(right) with {100,100,100} and no noise. The rows correspond to different lambda {1,5,10} respectively. V_i's are showed as the first 100 variables, while W_i's are the next 100 variables (101-200). For Network 2, U is the last variable.

- l_1-PLR **can select irrelevant features more frequently than non-causally relevant ones.** Network 1a (Figure 5.6), when lambda = 1, the averaging frequency of selecting irrelevant features can be approximately 20% more than non-causally relevant ones (not dependent on the sample size, see also Figure 5.7).

[1] Matlab implementation available on http://pages.cs.wisc.edu/~gfung/GeneralL1/ by Schmidt

As lambda grows, non-causally relevant features seems to be selected more frequent than non-relevant ones (see for lambda = 5 and 10).

- l_1-PLR **can select non-relevant features more frequently than causally relevant ones.** For network 2, even though some causally relevant features have higher selection frequency than non-relevant ones, some non-relevant features are selected in higher probability than many relevant ones.
- l_1-PLR **can not detect the causally relevant features over non-causally relevant ones in Network 2.** U is almost always selected, while V_i are only selected in a very low frequency (<0.25). This corresponds to the observation of lasso (Tibshirani, 1996), where there are large number of small effects (causally relevant features).
- **Sample size matters.** The performance of l_1-PLR is similar to its l_2 counterpart (Figure 5.5) with small feature to sample ratio. When sample size increases, the difference in probability of selecting causally relevant feature to that of selecting non-relevant ones grows in network 2 The probabilities of choosing some causally relevant features grow up to 1 (Figure 5.7). Interestingly, with even large sample size, for Network 1a, the probabilities selecting non-causally relevant features can always be smaller than non-relevant ones. We can see how sample size will affect the causal discovery process of such method.

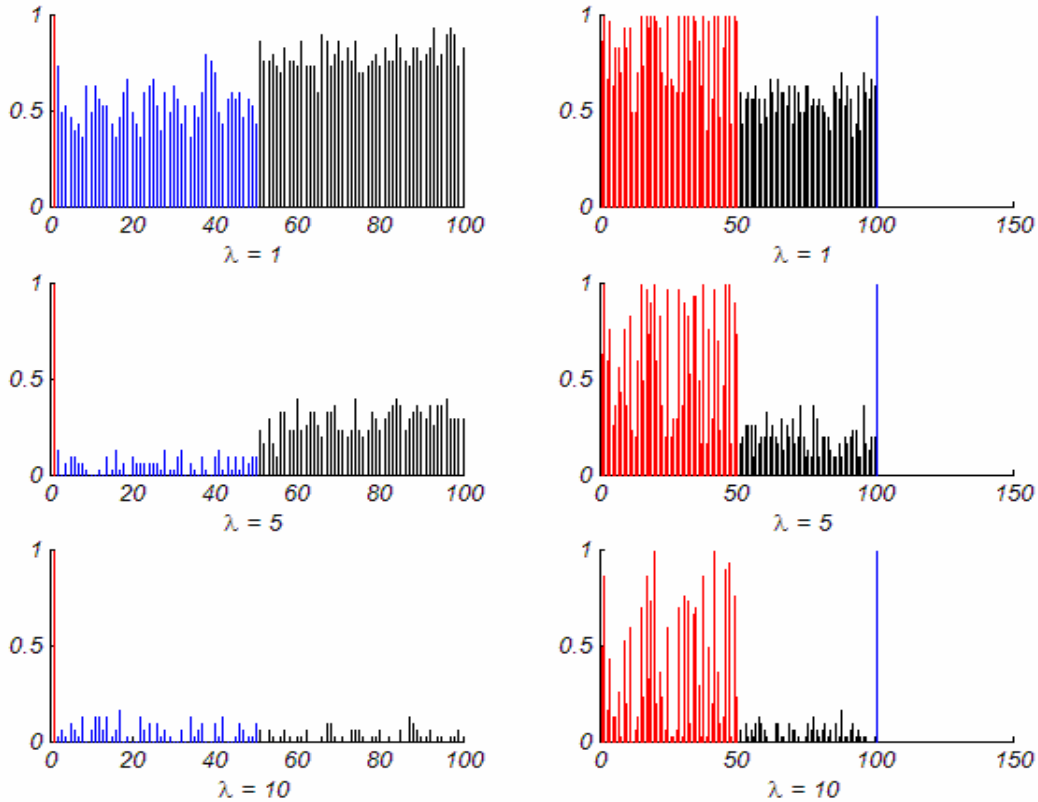

Figure 5.7 Selection frequencies of features by 1-norm Logistic regression for Network 1a (left) and Network 2(right) with {50,50,500} and no noise. The rows correspond to different lambda {1,5,10} respectively. V_i's are showed as the first 50 variables, while W_i's are the next 50 variables (51-100). For Network 2, U is the last variable.

5.3.1.4 l_0-norm SVM (linear separable)

By replacing the penalization of SVM in (4.3) with l_0-norm, we have the l_0-SVM. As discussed before, learning with zero-norm penalty is NP-hard. Here, we study the zero-norm SVM proposed by Weston et al. (2003). The algorithm approximates the zero-norm with multiplicative update and an additional constraint on the number of features and is shown to have competitive performance over other zero-norm approximations. Specifically, the linear separable case of Weston et al. scheme is the following optimization problem:

$$\min \|\boldsymbol{w}\|_l \quad \text{subject to} \quad y^{(i)}(\boldsymbol{w}.(\boldsymbol{x}^{(i)} \odot \boldsymbol{z}) - b) \geq 1 \quad \text{and} \quad \|\boldsymbol{w}\|_0 < r$$

The multiplicative update is used to search for the optimal for the problem until converge, where the weights are updated recursively, that is $\boldsymbol{z} = \boldsymbol{z} \odot \boldsymbol{w}'$ and $\mathbf{w}'$ is the previous solution to the problem. l can be 1 or 2 and r is pre-specified or chosen with the smallest prediction errors for the training set.

Since zero-norm SVM tends to output zeros weights for features, we can assess both rankings and the selection frequencies for each feature. Here, we use the implementation of zero-norm SVM in Spider[1] (Weston et al., 2005) with default settings. It is worth noting that the implementation copes with only linear separable cases, which is equivalent to C = Inf. i) When the number of features is decided by minimizing generalization errors, the algorithm will almost select only one feature (Figure 5.9) V_1 (>90%) for Network 1a/1b and U for Network 2). No relevant features are never selected ii) On the other hand, the rankings of features are also very similar to the previous methods, with no superiority in differentiating causal relevance. (Figure 5.10a) iii) the rankings based on zero-norm SVM are also affected by the sample size (Figure 5.10b)

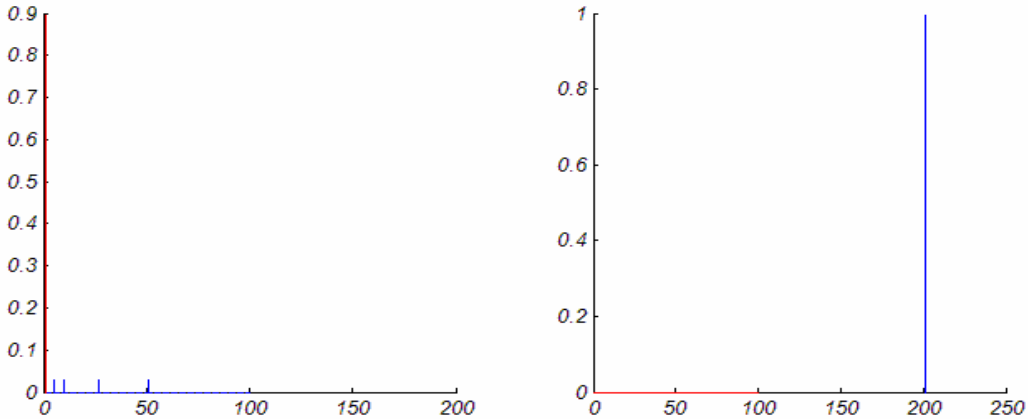

Figure 5.9 Selection frequencies of features by Zero-norm SVM for Network 1a (left) and Network 2(right) with {100,100,100} and no noise. V's are showed as the first 100 variables, while W's are the next 100 variables (101-200). For Network 2, U is the last variable.

[1] Spider is a MATLAB package for machine learning

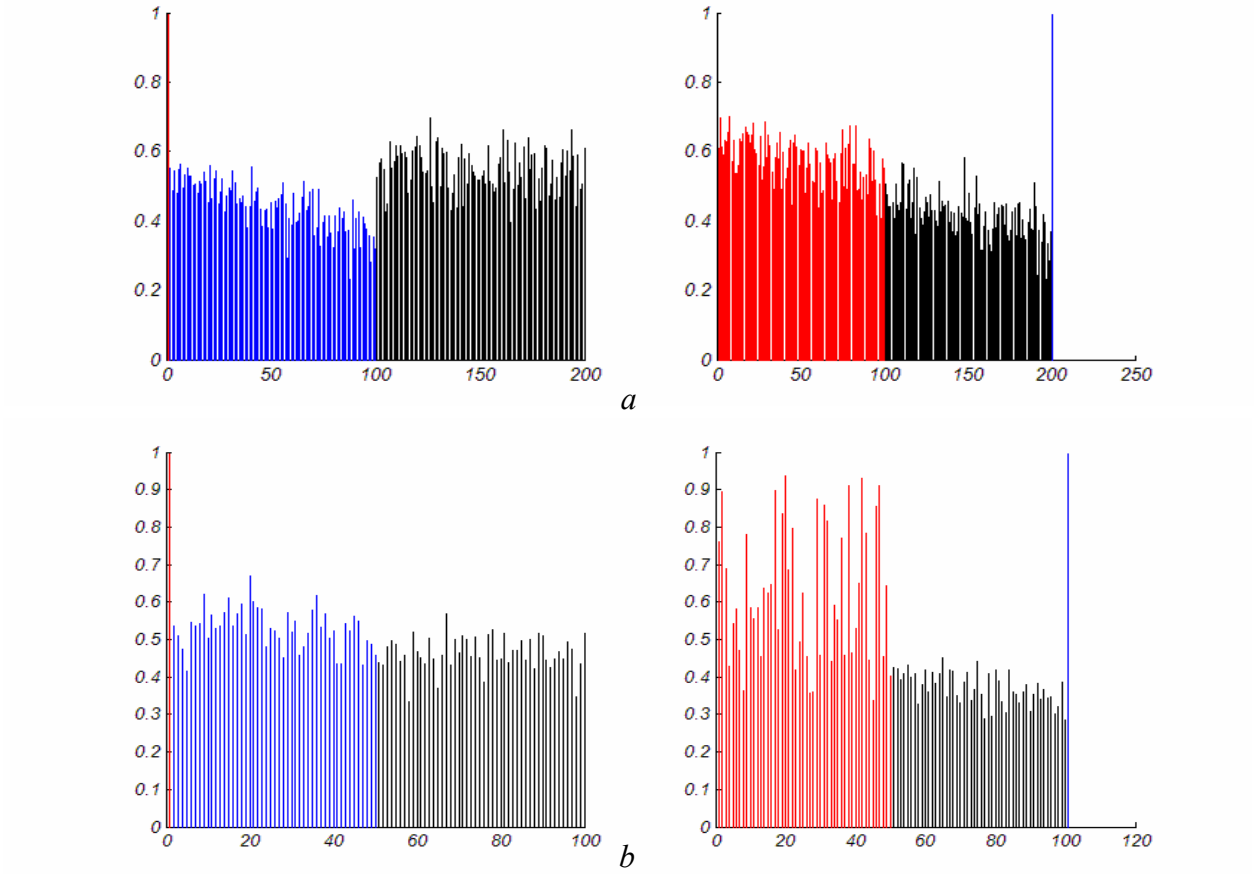

Figure 5.10 Average rankings based on Zero-norm SVM for Network 1a (left) and Network 2(right) with {100,100,100} and no noise with different feature to sample ratio a) {100,100,100}, b) {50,50,500}. V's are showed as the first 50(100) variables, while W's are the next 50 (100) variables (51-100,101-200). For Network 2, U is the last variable.

5.3.1.5 HITON-MB

Though HITON-MB has difficulties in eliminate false positives on the fly, it is worth trying to compare its performance to other methods that are designed specifically for MB discovery. It is implemented in CausalExplorer[1] (Aliferis et al. 2003b) and a set of features are output as the MB candidate from the training data. Again, we can evaluate the quality of the learned MB visually based on the frequency plot. For conditional independence tests, we can either choose G^2 test (for discrete data) or Fisher's z test (for continuous data). To apply those tests, the variables should either all be continuous or discrete. It is not the case for Network 2. In order to run the algorithm, we can transform U to discrete variable following the discretization scheme in Mitchell's book (1997). The discretization routine first tests whether the continuous variable is significantly correlated with the target using (Wilcoxon rank sum test or Kruskal-Wallis ANOVA) with 0.05 significance. The features should be discretized differently regarding whether it is significantly correlated with the target. For the setting of HITON-MB, we set the significance level of G^2 to be 0.05, and the maximal number of conditional variables to be 3. From Figure 5.11, we can see that

[1] MATLAB package available on *discover1.mc.vanderbilt.edu/discover/public/**causal_explorer**/*

- HITON-MB almost always selects only one feature (V_1 for Network 1a/1b, U for Network 2) for all networks. Non-relevant feature are almost never selected (similar to zero-norm SVM).
- The performance of HITON-MB is robust to the decreasing feature to sample ratio. For small feature to sample ratio {100,100,50}, HITON-MB produces the similar patterns in selection frequency to {50,50,500}.

The failure of HITON-MB in discovery the true MB of Y in Network 2 can be due to the fact that the network is unfaithful. There is deterministic relations between V_i's, U and Y, since the weights on v_i's in generating Y are fixed, implying that deterministic function between V_i's and U. Any V_i's can be substituted by U according to the generation. In fact, as the number of sample grows, U will contain sufficient information to predict Y regardless of V_i's, since knowing all the values of V_i's could be redundant where Y is a signed function of linear combination of V_i's.

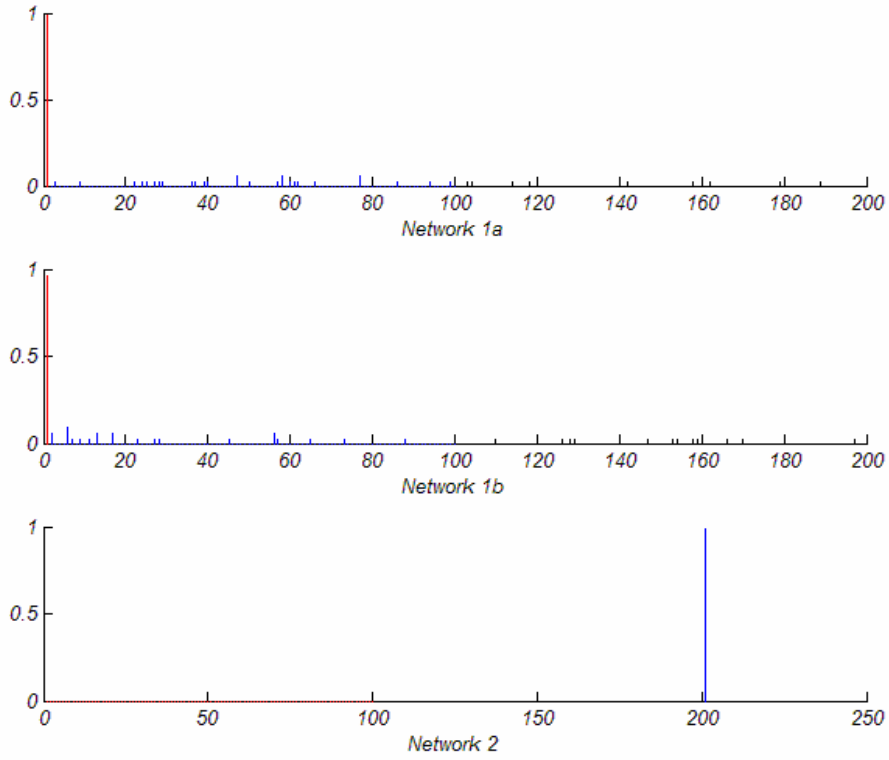

Figure 5.11 Selection frequencies by HITON-MB with {100,100,100} and no noise for the three networks. V_i's are showed as the first 100 variables, while W_i's are the next 100 variables (101-200). For Network 2, U is the last variable.

5.3.1.6 Backward Search MB

We implemented the framework of backward search algorithm with different classifiers and loss functions with the help of Spider. Generally, the losses with different classifiers are estimated via k-fold cross-validations and the model parameters are tuned with a pre-run cross-validation. For example, for linear SVM, we find the best C with 5-fold cross-validation on the full training data. Then we train the linear SVM with the given C through each subset of features with one left out. And we choose features whose left-one-out subsets have greater cross-validation errors than the full set. We notice that by simply setting ε to 0, the algorithm has

very little power in identifying MB of Y regardless of the classifiers and loss functions used when the sample to feature ratio is large. Therefore, we study several heuristics to be incorporated into the original backward search algorithm.

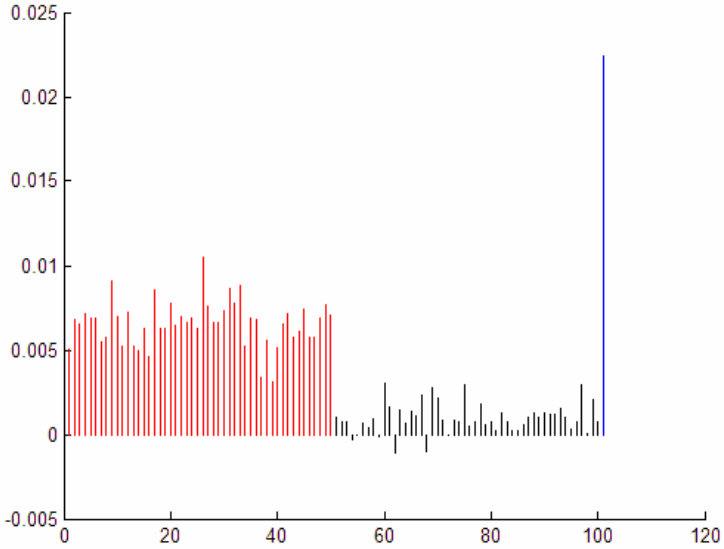

Figure 5.12 The average leave-one-out error of features by backward search (SVM) with {50,50,500} and no noise for the Network 2. Causally relevant features are shown in V_1-V_{50}, and non-causally relevant feature(s) are shown with U (the last one), while non-relevant features are shown with W_1-W_{50}(51-100).

- For a specified classifier and the CV errors (0/1 loss for all folds) on the training subsets, we use the McNemar's test to assess the significance level of $F(D_{R_i}) > F(D)$, then we select features with significance level 0.05. The exact McNemar's test lacks power due to the small size of samples. We use a modification by substituting $m_t(E_1 + E_2)$ as the total errors made by the two classifiers. In some sense, such tests provide a way of controlling the false positives implicitly.

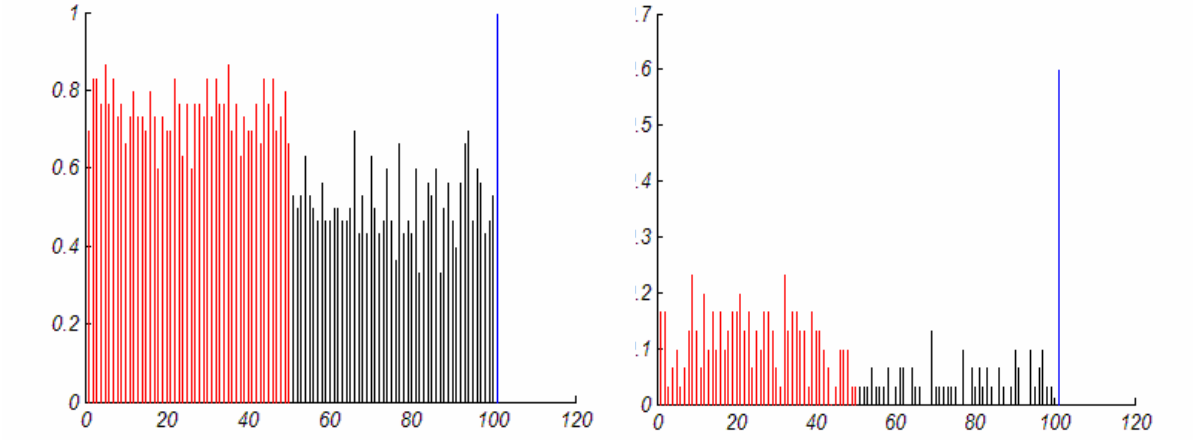

Figure 5.13 Selection frequencies by backward search (SVM) with {50,50,500} and no noise for the Network 2 with epsilon equal to 0 (left) and McNemar's test (right). Causally relevant features are shown in V_1-V_{50}, and non-causally relevant feature(s) are shown with U (the last one), while non-relevant features are shown with W_1-W_{50}(51-100).

The idea of using statistical test comes from the observations that even though some

irrelevant features (without which) do increase the CV errors, the amounts of increase are a lot lower than relevant ones on average (for 30 runs, Figure 5.12). Therefore, a statistical test based on levels of errors increase might be helpful. We can see that (Figure 5.13) that the original method with enough samples behaves similarly to filters for Network 2. On the other hand, McNemar's scheme helps to eliminate non-relevant features, but also decrease the probability of selecting relevant ones, which suggests that the McNemar's test is not very sensitive to relevance per se. Further study is needed for the potential of incorporating statistical tests.

Other simple heuristics to be incorporated include i) replace McNemar's test with Wilcoxon test or ii) assuming the distribution of $F(D_{R_i}) - F(D)$ is normally distributed with the same mean and standard variation for and i's, for feature in MB and not in MB respectively. Therefore, we can gather features with $F(D_{R_i}) > F(D)$, and estimate the mean and standard deviation of $F(D_{R_i}) > F(D)$. For all features with $F(D_{R_i}) > F(D)$, we then discard features with $F(D_{R_i}) > F(D)$ that are one standard deviation less than the estimated mean. It is of interest to see the whether such approximation is valid for such simple networks, where we use histogram to study the empirical distribution of $F(D_{R_i}) > F(D)$ for different kind of features. In the experiments not showed here, we notice that the exact Wilcoxon test fails drastically, and the second scheme is not as good as the McNemar's test.

From the experiments, we can see that even though backward search algorithm is theoretical consistent (guaranteed to find MB when sample size goes to infinity), it performs poorly in practice without proper control on epsilon. It is due to the slow speed of convergence and also relating to the large conditional set (equivalent to conditional independence tests), which can not be mitigated by controlling epsilon. Nevertheless, it is of interest to see the limited success of such a simple algorithm.

5.3.2 Real data

Even though some of above mentioned algorithms fail to capture the causal structure of the simple networks, they have been proved to be able to perform feature selection and improve the prediction for real data. Comparisons between those algorithms on the same data set might shed light on how feature selection should be conduct in practice. Specifically, we apply t-test statistic, RFE-SVM, l_1-logistic.regression, Zero-norm SVM and HITON-MB to the AML-ALL data and try to interpret the correspondence between the features selected by different algorithms. On the other hand, we evaluate the quality of the subsets selected by those algorithms by their improvement in prediction with two classifier 1) linear SVM and 2) lambda method (Golub et al., 1999, Guyon et al., 2003). The lambda method simply weights each feature with their importance (Golub, 1999), and adds an intercept regarding their class means. Y is then estimated as the sign of their linear combination. McNemar's test is used to assess the significance of the method of one method is different from the other due to the small test sample size. For RFE-SVM (Guyon et al., 2001) and RFE-PLR (Zhu et al., 2002) have been studied previously. In addition, Li and Yang (2005) applied RFE-ridge-regression to the data set and showed that 3 features are

sufficient to have no errors on the test set and suggested that feature selection should related to both classifiers and loss functions.

5.3.2.1 Performance evaluation

All the feature selection methods are used mostly with their default setting with exception on RFE-SVM and l_1-logistic.regression. Specifically, RFE-SVM is run with first an elimination step of factor 2 (half of the features are eliminated) when more than 64 features are left followed by the one by one elimination (this might be the difference in prediction for test data from Guyon et al., 2002). On the other hand, since L1General is unable to optimize such large dimension data, the feature selection with l_1-logistic.regression is done in three steps i) Eliminate features that are not significant with t statistic (0.001), ii) Perform l_1-logistic.regression on the features left with lambda = 12 iii) We take the top k features on weights with the best prediction on the training set. Note that for t statistic, we select the feature with significance level of 0.01, though the significance level can also be verified with cross-validation. The prediction errors (0/1 loss) on both the training set and test set with different feature selection method and the two classifiers are shown below.

Table 5.1 Prediction performance for different feature selection algorithms on AML-ALL

	No FS	T statistic	RFE-SVM	l_1-PLR	l_0 SVM	HITON-MB
Nr. of Features	7129	799	16	13	3	5
SVM	3(0)	2(0)	1(0)	3(0)	2(1)	8(1)
λ-method	3(0)	1(0)	1(0)	1(0)	4(1)	4(0)
Random Forest	3(0)	3(1)	3(0)	3(1)	6(0)	3(0)

The two numbers are number of test errors out of 34 samples (number of training errors out of 38 samples) respectively. The prediction performance that are significantly (0.05) different from the one with no feature selection are underlined.

From Table 5.1, for the same classifier, we can see that

i) RFE-SVM seems to be the best candidate for this data set for the overall performance for the three classifiers.

ii) As a matter of fact, l_1-PLR is also competitive since it is not fair to compare RFE-SVM and l_1-PLR on SVM classifier, which favors SVM-based selection.

iii) l_0 SVM shrinks the number of feature down to 3, with slight increase in test error (2 compared to only 1 in RFE-SVM). But the feature selected by l_0 SVM is not informative to both lambda method and random forest in the sense that it boosts the test errors into 4 and 6, respectively.

iv) T-statistic turns out to be useful in the sense that it reduces number of features to 799 and obtain relatively low test errors. It again suggests that univariate filter can be informative when sample size is small.

v) HITON-MB seems to be worst for all classifiers (for several choices of

significance levels and number of conditional set) except random forest (l_0 SVM is the worst in that case). Since the HITON-MB is of size 5, we try all the possible subsets of the features in HITON-MB, it turns out that there is one feature (#4847 (Zyxin- X95735_at)) very informative, with only which the test errors and training errors change into 3(1), 2(3), 3(1).

The performance of feature subsets is sensitive to the classifier used. Given 0/1 loss, since the data is known to be linear separable, both linear SVM and lambda methods performs equally well for all feature selection methods except HITONMB. Random forest has relatively larger prediction errors to the other two. On the other hand, random forest gives very similar prediction performance for all feature selection methods, but it seems to fail in extract information from the information from the feature subset output by l_0 SVM.

Table 5.1 indicates that feature selection methods should always be compared with the same classifiers; otherwise, the conclusion can be misleading when we assess the prediction quality of a feature subset. Different feature selection methods might favor specific classifiers.

5.3.2.2 Feature correspondence and MB discovery

By checking the selected feature subsets more closely, we can study the correspondences between those feature subsets. The best reported subset (Li and Yang, 2005) includes #4211, #6201 and #1882(0 test errors) with RFE-ridge regression, which are not in the intersection or a real subset of the union of the six feature selection methods. With the three features, the prediction performance improve greatly except for lambda classifier (0(0), 4(1), 2(0)). This again suggests the sensitivity of feature subset to the classifier used. To look at each method separately regarding the correspondence to the Li and Yang's subset, we can see that RFE-SVM contains #6201 and #1882 is in l_0-SVM, while HITON-MB recovers none of them. On the other hand, it is not surprised that T-statistic contains all three features since its size is still very large (when setting significance level to 0.001, #4211 is filtered out). l_1-PLR fail to recover any of them (it is not related to the preprocessing of T statistic filter since with 0.001 significance level, both #6201 and #1822 features are still in the feature subset). For this particular data, Yi and Yang pointed out that the ridge regression classifier successfully boosts relevant feature and penalizes irrelevant ones during the RFE process, which in turn identify this subset. Meanwhile, Ridge regression has very low capacity (simply linear regression with 2-norm penalization on the weights), and is very good candidate for data with small sample size.

To investigate the possibility of those algorithms in discovering MB(Y) (unknown), we try looking into the intersections of those subsets. There are two features selected by all methods, which are feature #3320 (U50136_rna1_at) and #5039 (Y12670_at). However, using this smaller subset does not improve the performance at all (for SVM the test error increases to 10). This indicates that these two features are not informative without the presence of other features, suggesting strong feature interactions. In all, we can see that the intersections of subsets given by different feature selection methods can be very uninformative by themselves.

Therefore, it might not be a good way to try to identify MB as the intersections of different methods. Nevertheless, we can see that the methods used here are not 'weak' in the sense of boosting, and they are partly correlated (e.g. RFE-SVM and Zero-norm SVM are all based on SVM, they can be correlated). It is of interest to see how weaker feature selection methods with small correlations can be combined to identify MB.

Chapter 6

Discussion and Conclusions

6.1 Discussion

The simulations and experiments conducted in this thesis aim to shed light on the following questions for two major subjects in feature selection: a. the causal structure (dependence structure) b. the prediction accuracy

1. Will feature selection algorithms without accounting for causal structures give better prediction to those designed based on dependence structures?
2. How well will non-structural based feature selection algorithms discover the causal structures of the generating network?

For the first question, we can see that most non-structure based feature selection algorithms have the power to greatly reduce size of features needed for the classifiers whereas ensuring the prediction accuracy in our study on AML-ALL data and simulated data. NIPS 2008's debate gave some more insights on this question. Robert Tillman and Spirtes (2008) conducted experiments on synthetic Bayesian network and their intention for this is that in the causal feature selection contest, algorithms like RFE-SVM, RFE-PLR, LASSO disregard the existence of causal structure of the data while producing equivalent or better prediction than algorithms incorporating the information of MB in causal feature selection competition in 2008. They did a set of simulations with manipulation on some variables and tried to compare the performance of well-designed SVM-variants and Markov-blanket-based algorithms. They pointed out that non-causal methods will actually perform worse when they are not invariant under manipulation. Furthermore, they suggested that in practice, one should handle the trade-off between the errors related to causality and errors due to over-fitting and non-true parametric assumptions for a causal model. Methods such as SVM which treat over-fitting well will sometimes cancel the errors caused by not taking causality into account. Methods based on learning causal structures should always be used carefully to avoid over-fitting. It is also suggested that learning complete causal structure is more difficult than prediction problem due to the existence of equivalent classes.

Other than the trade-off between bias and variance, feature selection should be discussed with respect to the classifiers and loss functions used, as pointed out by Tsamardinos et al. (2003a). In our study on the AML-ALL data, we can see that using the features from HITON-MB (based on MB), both SVM and lambda methods give significantly worse performance to other feature sets. However, the prediction

performance of HITON-MB improves greatly when the underlying classifier is random forest. For this matter, HITON method actually includes a post-processing wrapper to eliminate features from the approximated MB to improve the performance of prediction. This is another aspect of the problem one should be cautious about when conducting feature selection.

As for the second question, we can see that non-causal-structure based methods can actually extract some limited information on the causal structure of the network but some of them fail to distinguishing relevant feature to noise (SVM with large C). Nevertheless, we can see that for such feature selection methods, the ability in approximating MB should depend on the choice of model parameters. There exist parameters that can be controlled regarding the sparsity of the models (e.g. C in SVM), which mimic the behavior of controlling false positive in conditional independence test with significance level in discovery MB. Unlike the task of prediction where parameters can be selected with cross-validation errors (related to generalization errors), there is no theoretical argument on how to choose those parameter for MB discovery. Under the assumption that MB facilitates prediction accuracy, a straightforward heuristic is to simply use the same parameters for best prediction (Frey and Fisher et al., 2003). On the other hand, for Bayesian network learning, the parameter can be chosen based on the minimization of MDL (Schmidt et al, 2006). Some promising results were shown with simulation on simple networks with such heuristics for different methods such as decision trees (Frey and Fisher et al., 2003) and L1-penalizaed structure learning (Schmidt et al., 2006), which suggest that if carefully tuned, non-causality-based methods have potentials in discovering MB.

6.2 Conclusions and Future work

In this report, we present our study on several feature methods on both synthetic data and real data. This first goal of this study is to investigate the ability of different methods in discovering MB (or even causal structure) of the target in classification problems. Secondly, we studied and compared the prediction performance of the feature selection methods to understand the importance of identifying MB for prediction in real data.

Through the simulations, we are able to get insights on how various algorithms will be able to detect different relationships between variables, as well as the factors that will affect the performance of the algorithms. Specifically, we noticed that filters can be superior to other methods of high capacity when the sample sizes are small and that the effects of penalization on the sparsity of the models should be carefully taken care of.

For the real data set, we found that the performance of the feature subset can be very sensitive to the classifier used (for the same loss function). Therefore, when evaluating the quality of a feature set in prediction, one should always try several classifiers to ensure the stability of the feature subset.

Several aspects of this work should be extended for further study. Since in this study, the simulation is focused on simple networks, it is of interest to generate

network with other structure (e.g. XOR) to extend the study and see how the relaxation of faithfulness (Network 2) will affect the performance of those algorithms in MB discovery. Meanwhile, the analysis of feature selection in this report deals only with the observable features. The effects of hidden features on prediction are not assessable for absence of those features. However, whenever feature selection is used as an intermediate step in causal inference, the effects of hidden variables are not negligible. For example, the hidden common parent in a Bayesian network will alter two features that are non-relevant into relevant. Such effects are not discussed in this report. More importantly, it is of interest to study theoretically how various penalization function, classifiers and loss function will contribute to discovering MB. After that, more sophisticated heuristics on choosing model parameters for MB discovery might be derived.

Bibliography

Aha D. W. and R. L. Bankert. A comparative evaluation of sequential feature selection algorithms. In D. Fisher and J.-H. Lenz, editors, *Artificial Intelligence and Statistics* V, pages 199–206. Springer-Verlag, 1996.

Agresti, A. (2002). *Categorical Data Analysis*. New York: Wiley-Interscience

Akaike H., A new look at the statistical model identification". *IEEE Transactions on Automatic Control* 19 (6): 716–723. 1974

Antos A., L. Devroye, and L. Gyorfi. Lower bounds for Bayes error estimation. IEEE Transactions on PAMI, 21(7):643–645, July 1999.

Aliferis C.F., and I. Tsamardinos. Methods for Principled Feature Selection, for Classification, Causal Discovery, and Causal Manipulation. *Technical Report DSL-02-01*, 2002a.

Aliferis C.F., I. Tsamardinos. Algorithms for Large-Scale Local Causal discovery and Feature Selection In the Presence of Limited Sample or Large Causal Neighborhoods. *Technical Report DSL-02-08* Department of Biomedical Informatics. Discovery Systems laboratory. Vanderbilt Univerisity 2002b

Aliferis C.F., I. Tsamardinos, and A. Statnikov. HITON, a novel Markov blanket algorithm for optimal variable selection. In *Proceedings of the 2003 American Medical Informatics Association (AMIA) Annual Symposium*, pages 21–25, Washington, DC, USA, November 8-12 2003a.

Aliferis C.F., I. Tsamardinos, and A. Statnikov. Causal Explorer: A Probabilistic Network Learning Toolkit for Biomedical Discovery. The *2003 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS '03)*, June 23-26, 2003b.

Benjamini Y. and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Stat. Soc. B*, 85:289–300, 1995.

Breiman L., Random forests. *Machine Learning*, 45(1):5–32, 2001.

Bradley P. S. and O. L. Mangasarian. Feature Selection via Concave Minimization and Support Vector Machines. In *Proc. 15th International Conf. on Machine Learning*, pages 82–90. Morgan Kaufmann, San Francisco, CA, 1998.

Boser B. E., I.. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *5th Annual ACM Workshop on COLT*, pages 144-152, Pittsburgh, PA, 1992. ACM Press

Cerny V., A thermo-dynamical approach to the traveling salesman problem: an

efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41-51, 1985

Chickering D.M., D. Heckerman, C. Meek Large-Sample Learning of Bayesian Networks is NP-Hard. *Journal of Machine Learning Research* Volume 5 (Dec. 2004)
Cortes C. and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Díaz-Uriarte R. and S. Alvarez de Andrés. Gene Selection and Classification of Microarray Data Using Random Forest. *BMC Bioinformatics* 2006, 7:3

Dreyfus G. and I. Guyon Assessment Methods. In *Feature Extraction Foundations and Applications* Guyon I, Gunn S, Nikravesh M., Zadeh L.A. Springer; 1 edition 2006

Efron B., T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

Friedman J., T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, Discussion of "Consistency in boosting" by W. Jiang, G. Lugosi, N. Vayatis and T. Zhang. *Annals of Statistics*. 2004

Genkin A., D. D. Lewis, D. Madigan. Large-Scale Bayesian Logistic Regression for Text Categorization. *Technometrics.* August 1, 2007, 49(3): 291-304.

Genovese C.R., and L. Wasserman. Operating Characteristics and Extensions of the False Discovery Rate Procedure, *J. Royal Statist. Soc*. B, 64, 499--518.2002

Golub T.R. et al. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring *Science* Vol. 286 15 October 1999

Guyon I. and A. Elisseeff,.An Introduction to Variable and Feature Selection *Journal of Machine Learning Research* 3, 1157-1182 2003

Guyon I., J. Makhoul, R. Schwartz, and V. Vapnik. What size test set gives good error rate estimates? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:52–64, 1998.

Guyon I., J. Weston, and S. Barnhill, Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 46, 389–422, 2002

Hardin D., I. Tsamardinos, C.F. Aliferis: A theoretical characterization of linear SVM-based feature selection. *Proceedings of the Twentieth First International Conference on Machine Learning (ICML)* 2004.

Keerthi S.S., S.K. Shevade, A.N. Poo, A Fast Dual Algorithm for Kernel Logistic Regression. *Machine Learning*, 61, 151–165, 2005.

Kohavi R. and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence* 97, 273-324,1997

Kirkpatrick S., C.D. Gelatt, M.P. Vecchi (1983-05-13). Optimization by Simulated Annealing. *Science. New Series* 220 (4598): 671-680. ISSN 00368075.

Koller D. and M. Sahami. Toward optimal feature selection. In *13th International Conference on Machine Learning*, pages 284–292, July 1996.

Krishnapuram B., L. Carin, and A. Hartemink. Gene Expression Analysis: Joint Feature Selection and Classifier Design. In B. Schölkopf, K. Tsuda, and J.-P.Vert, editors, *Kernel Methods in Computational Biology*. MIT Press, Cambridge,MA, 2004.

Kudo M. and J. Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33(1):25–41, 2000.

Lal T.N., O. Chapelle, J. Weston, and A. Elisseeff, Embedded methods, In *Feature Extraction Foundations and Applications Guyon* I, Gunn S, Nikravesh M., Zadeh L.A. Springer; 1 edition 2006

Lee S.-I., V. Ganapathi, and D. Koller (2007). Efficient Structure Learning of Markov Networks using L1-Regularization. *Advances in Neural Information Processing Systems* (NIPS 2006).

Marill T. and D. M. Green. On the effectiveness of receptors in recognition systems. *IEEE Transactions on Information Theory*, 9(1):11–17, 1963.

Mitchell, *Machine Learning*, 1997, pages 72-73

Moore A.W., VC-dimension for characterizing classifiers. Lecture Notes. 2001 www.cs.cmu.edu/~swm

Ng A.Y., On feature selection: learning with exponentially many irrelevant features as training examples. In *Proceedings of the 15th International Conference on Machine Learning*, pages 404–412, San Francisco, CA, Morgan Kaufmann. 1998.

Ng A.Y., Feature selection, L_1 vs. L_2 regularization, and rotational invariance, Proceedings of the twenty-first international conference on Machine learning, p.78, July 04-08, 2004, Banff, Alberta, Canada

Nilsson R., J.M. Peña, J.Björkegren and J.Tegner, Consistence Feature Selection for Pattern Recognition in Polynomial Time. *Journal of Machine Learning Research* 8 (2007) 589-612

Oukhellou L., P. Aknin, H. Stoppiglia, and G. Dreyfus. A new decision criterion for feature selection: Application to the classification of non-destructive testing signatures. In *European Signal Processing Conference (EUSIPCO'98)*, Rhodes,1998.

Pearl J.. *Probabilistic Reasoning in Intelligent System*. Morgan Kaufmann, San Mateo, Ca, 1988

Pearl J., *Causality: Models, Reasoning, and Inference*, Cambridge University Press, 2000

Peña J.M., R. Nilsson, J. Björkegren, J. Tegnér. Towards scalable and data efficient learning of Markov boundaries. *International Journal of Approximate Reasoning* 45 (2007) 211–232

Perkins S., K. Lacker, and J. Theiler. Grafting: Fast, Incremental Feature Selection by Gradient Descent in Function Space. *Journal of Machine Learning Research*, 3:1333–1356, 2003.

Perneger T.V., What's wrong with Bonferroni adjustments, BMJ 1998;316:1236-1238

Poggio T., R. Rifkin, S. Mukherjee, and P. Niyogi. General conditions for predictivity in learning theory. *Nature*, 428:419–422, 2004.

Poggio T., R. Rifkin, S. Mukherjee, and A. Rakhlin. Bagging regularizes. *AI Memo* 2002-003, MIT, 2002

Quinlan J.R., 1986. Induction of Decision Trees. *Machine Learning*. 1, 1 (Mar. 1986), 81-106

Quinlan J.R.. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA, 1993.C. M. Bishop. *Neural networks for pattern recognition.* Clarendon, Oxford, 1997.

Roth V.. The Generalized LASSO. *IEEE Transactions on Neural Networks*, 2003.

Schmidt M. and A. Niculescu-Mizil and K. Murphy, Learning Graphical Model Structure using L1-Regularization Paths, Department of Computer Science University of British Columbia/+Cornell University

Schwarz G., Estimating the dimension of a model. *Annals of Statistics* 6(2):461-464. 1978.

Shannon, C.E. and Weaver, W. (1949). The mathematical theory of communication. University of Illinois Press, Urbana, Illinois.

Siedlecki W. and J. Sklansky. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(2):197–220, 1988.

Siedlecki W. and J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(5):335–347, 1989.

Snedecorand, G.W, and W.G. Cochran. *Statistical Methods*, 8th ed. Iowa State University Press, Berlin, Heidelberg, New York, 1989

Spirtes P., C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, Springer-Verlag, 1993.

Stracuzzi, D. J., & P. E. Utgoff. Randomized variable elimination. Journal of Machine Learning Research, 5, 1331-1364, 2004

Statnikov A, Hardin D, Aliferis CF, Using SVM Weight-Based Methods to Identify Causally Relevant and Non-Causally Relevant Variables. *Neural Information Processing Systems (NIPS) 2006 Workshop on Causality and Feature Selection*, 2006

Tibshirani R.. Regression Shrinkage and Selection via the Lasso. *Journal of royal Statistical Society*, Series B (Methodological), volume 58, Issue 1(1996),267-288

Tillman R.E. and P. Spirtes. When causality matters for prediction: investigating the practical tradeoffs JMLR: *Workshop and Conference Proceedings 1: 1-16 NIPS 2008 workshop on causality*. 2008

Tsamardinos I. and C.F. Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In *Proceedings of the Workshop on Artificial Intelligence and Statistics*, 2003a.

Tsamardinos I., C.F. Aliferis and A. Statnikov. Algorithms for large scale Markov blanket discovery. In *The 16th International FLAIRS Conference,* St. Augustine, Florida, USA, 2003b.

Tsamardinos I., C.F. Aliferis and A. Statnikov. Time and sample efficient discovery of Markov blankets and direct causal relations. *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining* (KDD), 673-678. 2003c

Vapnik V. and A. Chervonenkis On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264--280, 1971.

Wainwright M., P. Ravikumar, J. Lafferty, High dimensional graphical model selection using L1-regularized logistic regression.. In *NIPS* 2006

Weston J., A. Elisseeff, B. Schölkopf, and M. Tipping. Use of the Zero-Norm with Linear Models and Kernel Methods. *Journal of Machine Learning Research*, 3:1439–1461, March 2003.

Weston J., A. Elisseeff, G. BakIr, F. Sinz. Spider. Machine Learning Toolbox http://www.kyb.tuebingen.mpg.de/bs/people/spider/,2006.

Weston J., S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In S.A. Solla, T.K. Leen, and K.-R. Muller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 526–532,Cambridge, MA, USA, 2000. MIT Press.

Whitney A.W.. A direct method of nonparametric measurement selection. *IEEE Transactions on Computers*, 20(9):1100–1103, 1971.

Wolpert D.H. and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*,1(1):67-82,April 1997.

Yu L. and H. Liu 2004. Efficient Feature selection via Analysis of Relevance and

Redundancy. *Journal of Machine Learning Research* 5 (2004) 1205–1224

Zhao P. and B. Yu. Stagewise Lasso. *Journal of Machine Learning Research* 8 (2007) 2701-2726

Zhu J. and T. Hastie, Classification of Gene Microarrays by Penalized Logistic Regression. *Biostatistics*, 5, 3, pp. 427-443, 2004

Zhu J., S. Rosset, T. Hastie, R. Tibshirani. 1-Norm Support Vector Machines. In *Proc. NIPS*, 2003