Feature Selection in Computational Biology

Dimitrios Athanasakis

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Engineering
of the
University of London.

Department of Computer Science
University College London

June 10, 2014

I, Dimitrios Athanasakis, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

This thesis concerns feature selection, with a particular emphasis on the computational biology domain and the possibility of non-linear interaction between features. Towards this it establishes a two-step approach, where the first step is feature selection, followed by the learning of a kernel machine in this reduced representation.

Optimization of kernel target alignment is proposed as a model selection criterion and its properties are established for a number of feature selection algorithms, including some novel variants of stability selection. The thesis further studies greedy and stochastic approaches for optimizing alignment, proposing a fast stochastic method with substantial probabilistic guarantees. The proposed stochastic method compares favorably to its deterministic counterparts in terms of computational complexity and resulting accuracy.

The characteristics of this stochastic proposal in terms of computational complexity and applicability to multi-class problems make it invaluable to a deep learning architecture which we propose. Very encouraging results of this architecture in a recent challenge dataset further justify this approach, with good further results on a signal peptide cleavage prediction task.

These proposals are evaluated in terms of generalization accuracy, interpretability and numerical stability of the models, and speed on a number of real datasets arising from infectious disease bioinformatics, with encouraging results.

Acknowledgements

Some of the few things surpassing professor John Shawe-Taylor's depth and breadth of knowledge on the subject of machine learning is his patience and encouragement towards students. My time working with John has been a privilege. I also need to thank my second supervisor, dr Delmiro Fernandez-Reyes for providing his support whenever it was asked for, during numerous occasions throughout my degree.

The people who have spent time reviewing my work as part of the thesis comittee have been pivotal in shaping the thesis. These include Mark Herbster, Juho Rousu, and Arthur Gretton. Their collective input has had great impact on the thesis.

The division of parasitology at the National Institute for Medical Research and the computer science department at UCL have been great places to work. They have my thanks for being excellent sources of interesting new problems and applications, and for giving me some of the smartest friends I've met to this day.

Finally, I would like to thank my family for always being there. This thesis is the result of my parents' encouragement to work on whatever I find fulfilling. Thank you.

Contents

1 Introduction **12**

1.1 Background . . . 12

1.2 Approaches to feature selection . . . 14

1.2.1 Filters . . . 15

1.2.2 Wrapper Methods . . . 16

1.2.3 Embedded Methods . . . 16

1.2.4 Meta-Selection Approaches . . . 17

1.3 Application to biomarker discovery . . . 17

1.4 Structure of the thesis . . . 18

2 Background: Enabling Technologies **19**

2.1 Linear Learning . . . 19

2.2 Max margin classification and support vector machines . . . 20

2.3 Non-linear maps and kernels . . . 21

2.3.1 SVM Dual . . . 21

2.3.2 Kernel Target Alignment . . . 21

2.4 Feature Selection For SVMs . . . 22

2.4.1 Filtering . . . 22

2.4.2 Wrappers . . . 23

2.4.3 Embedded Feature Selection . . . 24

2.4.4 Meta-Selection . . . 25

2.5 Evaluation . . . 27

2.5.1 Experimental Set Up & Cross-Validation . . . 30

3 Stability Selection **31**

3.1 Model Selection with KTA . . . 32

3.2 Extending Stability Selection . . . 35

3.2.1 loss functions for classification . . . 35

3.3 Experiments & Results . . . 37

3.3.1 Synthetic Data . . . 37

3.3.2 Real Data . . . 44

3.4 Discussion . . . 48

4 Randomised Feature Selection **50**

4.1 Related Work . . . 51
4.2 A randomized algorithm for feature selection . . . 53
4.2.1 Development of key ideas . . . 53
4.2.2 A randomized algorithm for feature selection . . . 54
4.2.3 Properties of the algorithm . . . 60
4.2.4 Model Selection . . . 60
4.3 Experiments & Results . . . 61
4.3.1 Synthetic Data . . . 62
4.3.2 Real Data . . . 67
4.4 Discussion . . . 70
4.5 A weighting scheme for randomized feature selection . . . 71

5 Deep(er) Learning **74**

5.1 Introduction . . . 74
5.2 Feature Selection for learned representations . . . 75
5.3 Prediction . . . 78
5.3.1 Results on the ICML Black Box Learning Challenge . . . 79
5.4 Application to cleavage site prediction . . . 79
5.4.1 Experimental Pipeline . . . 79
5.5 Experiments . . . 80
5.5.1 Results . . . 81
5.6 Discussion . . . 83

6 Conclusions **84**

6.1 Future Work . . . 84

Appendices **86**

A Appendix A **87**

A.1 Concentration Inequalities . . . 87
A.1.1 Hoeffding's inequality . . . 87
A.1.2 Hoeffding's concentration bound for U-Statistics . . . 87
A.2 Sparse Filtering Implementation . . . 87
A.3 Datasets . . . 89
A.4 Real Data . . . 89
A.5 Software . . . 90

Bibliography **90**

List of Figures

2.1 regularization path for lasso on TB dataset. The TB dataset comprises 523 features. Here, each line represents the computed weight-coefficients for each variable at varying levels of regularization. It can be seen that for increasing amounts of regularization, controlled by the parameter λ, the 1-norm of the weight $\mathbf{w}$ decreases, resulting in models that rely on fewer variables. The λ parameter essentially controls the exchange between sparsity and goodness of fit for the Lasso problem. 25

2.2 Examples of selected features over 30 cross-validation folds for a toy 20-dimensional dataset. Black bars indicate that the the feature was selected for that fold(features:left-to-right, folds: top-to-bottom). We consider the first two features to be relevant, and selected with varying probability between 1 and 0.1. The remaining, features are selected uniformly at random with a probability of 0.05. 28

2.3 Example Mean Variance for different inclusion probabilities of the relevant features in the scenarios of figure 2.2. This figure illustrates that only estimating the variance of a feature can be misleading. In this example, selecting the two relevant features with probability 0.9 appears to have larger mean variance than selecting them with probability 0.1. 29

2.4 Plot for the log likelihood corresponding to the different selection probabilities of the relevant features. The correspondence between log likelihood and the high selection rate of the relevant components is substantially improved. For selection probabilities 0.2 and 0.1 the log likelihood is positive, as we are estimating an upper bound of the true probability. 29

3.1 Dependence of target alignment on the σ parameter of the gaussian kernel. For increasing number of irrelevant features, smaller values of σ tend to produce higher alignment. This appears to stem from increasing the effective dimensionality, and has been also observed in ([SFG+09], sec. 5). 34

3.2 Different loss functions for classification. The lasso minimises the square loss (blue). The 0-1 loss function(black), which is directly related to accuracy, is NP-hard to optimise. Two loss functions commonly applied to classification problems are the logistic loss (magenta), and the hinge loss(red), that act as proxies for the 0-1 loss. 35

3.3 Results for the fake class dataset. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case. 39

3.4 Results for the Linear Zhang with Feature Noise. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case. 40

3.5 Results for the linear Zhang dataset with sample noise. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case. 41

3.6 Results for the linear Weston dataset. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case. 42

3.7 Results for the non-linear Weston dataset. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case. 43

3.8 Results for the XOR dataset. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case. 44

3.9 Results for the first TB task. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case. 45

3.10 Results for the second TB task. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case. 46

3.11 Accuracy results for the third TB task. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case. 46

3.12 Results for the fourth TB task. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case. 47

3.13 Results for the TB micro-array task. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case. 48

4.1 Mean alignment as a function of sample size. For each sample size the mean alignment is computed on 500 bootstraps of the data. Black line corresponds to a a random 25-dimensional multivariate gaussian. Red line depicts the mean alignment of 2 relevant variables generated according to the [WMC$^+$00], and 23 gaussian probes. Green depicts the alignment for only the two relevant features. The influence of self-interaction terms (the diagonal terms $c_{x_{ii}} c_{y_{ii}}$) decreases for larger sample sizes, leading to the displayed drop here. 52

4.2 Scatter plot of the resulting alignment over random splits of variables for an XOR dataset with the additional inclussion of a varying number of irrelevant variables (blue n=5, green n=10, red n= 15, light blue n= 20, magenta n = 25, bootstrap size = 100 samples). Over the different number of dimensions, a common pattern emerges: Samples concentrated over the lower left corner correspond to estimates in which each random split of the variables contains a single relevant feature, resulting in lower alignment for both splits. The bottom-right and top-left corners contain cases where one split contains both of the relevant variables, resulting in a visible hike in alignment for that split. It is also particularly instructive to notice that for samples corresponding to higher signal-to-noise ratios, the variance of the resulting alignment seems to be much higher, further justifying subsampling. 54

4.3 200-dimensional XOR classification problem. The expected contribution of the two relevant features is in red. It can be seen that as more of the noise features are removed in latter iterations of the method, the expected contribution of the two relevant variables rises substantially, in contrast to the contribution of the other features. 57

4.4 Results for the fake class dataset. 62

4.5 Results for the linear Zhang with feature noise dataset. 63

4.6 Results for the linear Zhang with sample noise dataset 64

4.7 Results for the linear Weston dataset. 64

4.8 Results for the non linear Weston dataset. 65

4.9 Results for the XOR dataset. 66

4.10 Results for the first TB task. 67

4.11 Results for the second TB task. 68

4.12 Results for the third TB task. 68

4.13 Results for the fourth TB task. 69

4.14 Results for the TB Micro-Array task. 69

5.1 Overall architecture; randSel is applied on the features learned by sparse filtering, producing a number of nonlinear combinations of learned features of increasing granularity. A number of kernels is defined on these nonlinear combinations of features, and multiple kernel learning is used for the overall prediction. 76

5.2 How the learned representation is generated. The amino-acid sequence is broken into smaller windows. Each amino-acid in the window is represented by its 54 distinct physicochemical properties. Sparse filtering is used to learn a representation for this encoding. . 80

5.3 Accuracy of different feature selection and representation approaches on the signal peptide dataset. All feature selection approaches that operate on the learned representation clearly outperform the original features. Methods employing a single kernel for prediction result in similar accuracy. Combining randSel with MKL outperforms all other approaches. 82

5.4 Substantial differences in correlation between full dataset (top row) and the support vectors (bottom row). This is largely due to the support vectors comprising of more atypical examples, for which the constraint of the SVM optimization problem is active. It is not visually obvious, but the two rows are very correlated, however some ordering information, which is important for feature selection, is lost in the active set. This means that while the ranking the features by correlation to the target output is largely similar between the active set and the entire set of variables, the differences in rank are substantial enough to affect the behavior of RFE. 82

List of Tables

3.1 Class proportions for synthetic data. 38
3.2 Fold Decrease for the time requirements of different methods when using alignment instead of cross validation 43

Chapter 1

Introduction

We present a family of two-stage techniques for kernel machines. The first stage of these techniques is a feature selection method in order to reduce the number of variables while the second stage is fitting a kernel machine on this lower- dimensional dataset. Given the potential benefits of a reduced representation it is not surprising that a rich variety of methods have been proposed in order to achieve this. Such algorithms are routinely used in order to alleviate problems such as the so-called curse of dimensionality and reduce generalisation error. Another common use for these methods is in reducing the storage and processing requirements for large datasets. However, the most important property as far as the field computational biology is concerned may be that of parsimony. Models relying on fewer variables are easier to explain and have the potential to accelerate experimental validation by providing valuable insight into the importance and role of the variables. Considering their potential applications we enumerate four highly desirable properties for the ideal feature selection algorithm:

1. *Low generalization error*. Feature selection should improve generalization performance, or at the very least not deteriorate the error rate as compared to learning on the full set of features.

2. *Parsimony*. The resulting models should use the smallest possible number of features. Models relying on fewer variables are easier to study. This can be viewed as a form of Occam's razor whereby hypotheses that rely on fewer variables are preferred.

3. *Selection Consistency*. Here selection consistency means that the variables selected should not radically change for small perturbations in the data. This is a key property in computational biology where *dirty* data such as noisy outputs or labels are not uncommon.

4. *Scalablility*. The size of datasets is increasing at a much faster rate than speed increases in commodity hardware can cope with. Given this simple fact, computationally efficient algorithms and methods which can be parallelized are advantageous.

1.1 Background

It is not uncommon to encounter features that exhibit a high degree of redundancy in practice. In such cases the SVM classifier tends to assign similar weights to features that exhibit a high degree of similarity between them. It is also possible however to have a large number of variables that appear to be irrelevant

to the target concept lead to deteriorating generalization. In both of these cases it is possible for a benign collusion between feature selection mechanisms and SVMs to yield many of the benefits previously listed above. For this to occur, an effective feature selection method is expected to find a small and informative set of relevant features as well as identify features that are redundant or irrelevant to the target concept. An attempt to formalize the notions of feature relevance follows.

The following section summarizes work and definitions in [GGNZ06], where the authors attempt to produce formal mathematical definitions of the notions of feature relevance.

Making the assumption that the dependency between input patterns $\mathbf{X}$ and desired outputs $\mathbf{Y}$ is governed by the joint distribution $P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{Y}|\mathbf{X})P(\mathbf{X})$ we introduce the following auxiliary notations. Let $\mathbf{V}$ be some subset of $\mathbf{X}$, $\mathbf{X}^{-i}$ bet the subset of $\mathbf{X}$ excluding feature $\mathbf{x}_i$ and $\mathbf{V}^{-i}$ be a subset of $\mathbf{X}^{-i}$. Those are used to make the following definitions of variable relevance or irrelevance.

1.1.0.1 Surely Irrelevant Features

A feature $\mathbf{X}_i$ is surely irrelevant iff for all subset of features V^{-i} including $\mathbf{X}^{-i}$ we have:

$$P(\mathbf{X}_i, \mathbf{Y}|\mathbf{V}^{-i}) = P(\mathbf{X}_i|\mathbf{V}^{-i})P(\mathbf{Y}|\mathbf{V}^{-i})$$

This definition can be extended through the use of the Kullback-Leibler divergence between $P(\mathbf{X}_i, \mathbf{Y}|V^{-i})$ and $P(\mathbf{Y}|\mathbf{V}^{-i})$ giving rise to the following measure of conditional mutual information, $MI(\mathbf{X}_i, \mathbf{Y}|\mathbf{V}^{-i})$:

$$MI(\mathbf{X}_i, \mathbf{Y}|\mathbf{V}^{-i}) = \sum_{\{\mathbf{X}_i, \mathbf{Y}\}} P(\mathbf{X}_i, \mathbf{Y}|\mathbf{V}^{-i}) \log \frac{P(\mathbf{X}_i, \mathbf{Y}|\mathbf{V}^{-i})}{P(\mathbf{X}_i|\mathbf{V}^{-i})P(\mathbf{Y}|\mathbf{V}^{-i})}$$

The above sum runs over all possible values of the random variables $\mathbf{X}_i$ and $\mathbf{Y}$. The conditional mutual information can be used to derive a score that encapsulates the relevance of a feature $\mathbf{X}_i$ by summarizing over all the values of $\mathbf{V}^{-i}$. For the relevance of feature $\mathbf{X}_i$, given $\mathbf{Y}$, we obtain the expected mutual information as:

$$EMI(\mathbf{X}_i, \mathbf{Y}) = \sum_{\mathbf{V}^{-i}} P(\mathbf{V}^{-i}) MI(\mathbf{X}_i, \mathbf{Y}|\mathbf{V}^{-i})$$

Using this quantity we can proceed to the following definitions.

1.1.0.2 Approximately Irrelevant Features

A feature $\mathbf{X}_i$ is approximately irrelevant with level of approximation $\epsilon > 0$, iff for all subsets of features $\mathbf{V}^{-i}$ including $\mathbf{X}^{-i}$,

$$EMI(\mathbf{X}_i, \mathbf{Y}) \leq \epsilon$$

1.1.0.3 Surely Sufficient Feature Subsests

We have provided a definition of feature relevance. However even relevant features may be redundant. Hence, merely ranking features by their relevance is not sufficient in order to extract a minimum subset of features that produce optimal predictions. We introduce the additional notation $\bar{\mathbf{V}}$ for the subset that complements a set of features $\mathbf{V}$ in $\mathbf{X}$: $\mathbf{X} = [\mathbf{V}, \bar{\mathbf{V}}]$. Using this we obtain that a subset $\mathbf{V}$ of features is surely sufficient iff, for all assignments of values to its complementary subset $\bar{\mathbf{V}}$

$$P(\mathbf{Y}|\mathbf{V}) = P(\mathbf{Y}|\mathbf{X})$$

Similarly to the definition of feature relevance, we can extend this through the use of mutual information, defining a new quantity, $DMI(\mathbf{V})$:

$$DMI(\mathbf{V}) = \sum_{\{u,\bar{u},\mathbf{y}\}} P(\mathbf{X} = [\mathbf{v}, \bar{\mathbf{v}}], \mathbf{Y} = y) \log \frac{P(\mathbf{Y} = y|\mathbf{X} = [\mathbf{v}, \bar{\mathbf{v}}])}{P(\mathbf{Y} = y|\mathbf{V} = \mathbf{v})}$$

The above quantity, which was introduced in [KS96] is the expected value over $P(\mathbf{X})$ of the Kullback-Liebler divergenvence between $P(\mathbf{Y}|\mathbf{X})$ and $P(\mathbf{Y}|\mathbf{V})$. It can be verified that :

$$DMI(V) = MI(\mathbf{X}, \mathbf{Y}) - MI(\mathbf{V}, \mathbf{Y})$$

1.1.0.4 Approximately sufficient feature set

Again in similarity with the previous definitions of approximate feature relevance, we have that a subset $\mathbf{V}$ of features if approximately sufficient with level of approximation $\epsilon \geq 0$, iff

$$DMI(\mathbf{V}) \leq \epsilon$$

In the case that $\epsilon = 0$ the subset $\mathbf{V}$ is called *almost surely sufficient*.

1.1.0.5 Minimal Approximately Sufficient Feature Subset

Finally a subset $\mathbf{V}$ of features is minimal approximately sufficient, with level of approximation $\epsilon \geq 0$ iff it is ϵ-sufficient and other ϵ-sufficient subsets of smaller size do not exist. From this definition it follows that a minimal approximately sufficient subset is a (possibly non-unique) solution to the following optimization problem:

$$\min_{\mathbf{V}} ||\mathbf{V}||_0, \text{ s.t. } DMI(\mathbf{V}) \leq \epsilon$$

Where $||\mathbf{V}||_0$ is the zero norm, which equals the number of selected variables. These definitions underpin the fact that there is no one-size-fits all approach to feature selection, as objectives can vary substantially.

1.2 Approaches to feature selection

Finding the minimal approximately sufficient feature set is a NP-hard problem. Naive attempts to computationally identify an optimal combination of features by exhaustive search are infeasible, even for a relatively small number of variables. Practical feature selection algorithms address this limitation by using a number of proxy criteria related to the selection problem. Depending on the core criterion, three principal selection approaches can be identified:

- *Filter Methods* employ simple heuristics or statistical tests for evaluating the importance of a feature.

- *Wrapper Methods*, in contrast to filter methods tend to refine an initial solution by relying on performance bounds or empirical estimates that encapsulate the impact that removing a feature, or number of features would have to the solution of the problem.

- *Embedded Methods*, where feature selection is embedded as part of the learning algorithm, typically by using L_1-norm minimization or similar approaches.

1.2.1 Filters

Filter methods provide a simple and fast approach to filter selection. Typically, filters rely on statistical criteria which act as a proxy for the underlying modelling problem. A typical example of this approach would be a filter that employs correlation coefficients. Making the assumption that informative features are highly correlated with the classification target, it ranks features by the magnitude of their correlation. Variables that do not exhibit a user-specified degree of correlation are filtered out of the dataset, resulting in a simple and efficient filtering method .

This simple, and efficient approach has been successfully employed in a large variety of problems. However, there are drawbacks to using correlation coefficients. First, correlation coefficients make the assumption that the underlying relation is linear. While this assumption applies to a certain degree in numerous problem domains, it is a clear limit to the scope of its application. Furthermore, correlation coefficients fail to encapsulate interactions between variables. The relief algorithm [KR92] is a multivariate filter method which addresses some of the shortcomings of correlation coefficients. The relief method uses a k-nearest-neighbor derived criterion.At each iteration, the algorithm cycles through the samples, estimating the relevance as a function of the nearest within-class and out-of-class samples. This is achieved through comparing the distance of the ith sample to its nearest hits (closest, within class neighbors) and nearest misses (closest out-of-class neighbors), with and without the inclussion of a feature.

In recent years, information-theoretic approaches to filtering have seen substantial growth. Numerous methods using mutual information as an indicator of statistical dependence have been proposed, with encouraging results on real world data [ZH02]. [Bro09],[BPZL12] propose the conditional likelihood of the training labels as a unifying theme for information theoretic approaches, viewing such filters as belonging to a spectrum of approximate maximizers of the conditional likelihood and differentiated by their implicit statistical assumptions.

Another line of inquiry relies on the estimation of cross-covariance between kernels defined on the input and output data [GBSS05], which proposes the Hilbert-Schmidt Independence Criterion. This principled approach benefits from sharp concentration estimates. [SBB$^+$07] illustrates procedures relying on HSIC are fairly simple to implement, and do not require user intervention in terms of regularization. Further work utilising HSIC for feature selection is found in [SSG$^+$12] where a slew of connections with existing algorithms is presented, including its similarity to Kernel Target Alignment (KTA).

1.2.2 Wrapper Methods

Wrapper methods are strongly paired to the classification problem. One of the first wrappers, specific to support vector machines (SVMs)was introduced in [WMC$^+$00]. The approach used gradient descent optimization on a margin bound, removing the least-weighted features after each iteration. Recursive feature elimination (RFE[GWBV02]), is similar in function, removing the least weighted feature from the actual linear predictor at the end of each iteration. In the kernelized setting, RFE is significantly more demanding. Non-linear RFE quantifies the sensitivity of the learning rule to the removal of each individual feature. This means that for non-linear, high-dimensional datasets, RFE is tractable but computationally prohibitive as at each iteration it needs to train a support vector machine, and then estimate the sensitivity to the individual removal of the remaining features.

1.2.3 Embedded Methods

In recent years, there has been a flurry of embedded feature selection methods. Embedded methods achieve parsimony by enforcing additional constraints to the recovered solution. Lasso,[Tib96] is one of the earliest proposals in this class of algorithms. The lasso algorithm attempts to minimise the sum of the square loss of the derived predictor and a multiple of the ℓ_1-norm of the weight vectors. Setting the regularization of the weights complicates model selection. To some degree, this complication is addressed by regularization paths. A regularization path, is a piecewise linear function that associates the regularization parameter with a corresponding recovered solution to the optimization problem. The homotopy algorithm described in [OPT00], leverages regularization paths to describe an efficient algorithm that computes the entire space of recovered solutions with small additional computation overhead. This is further elaborated in [EHJ$^+$04] in the context of the LARS algorithm, where an algorithm that computes the regularization path in time similar to ordinary least squares is introduced.

The Lasso objective optimizes the square loss. The square loss is applicable to a wide variety of settings, including classification and regression. In classification problems the same approach, using loss functions specific to the classification setting such as the logistic loss [LLAN06],is applied. Warm-start techniques [KKB07], use a previously computed solution as the starting point for computing the solution for an updated regularization term. For the hinge loss, commonly employed by SVMs, a regularization path approach has been proposed in the case of ℓ_2-regularized SVMs [HRTZ04]. To the best of my knowledge, there is no published work detailing a regularization path approach for ℓ_1-regularized SVMs.

Boosting [FS97] can also be thought of as an embedded feature selection algorithm. Boosting attempts to produce a strong learning rule by combining predictions produced by weak base learners. A weak learner can be thought of as a simple prediction algorithm relying on a subset of features. By limiting the number of weak learners boosting can be effectively used as a feature selection algorithm, a procedure which is illustrated in [BY06]. The most common boosting variant, adaboost [FSA99], optimizes the exponential loss. LPBoost [DBST02] is another boosting algorithm, geared towards optimizing the hinge loss, and in the setting where a base learner corresponds to a single feature can be thought of as an ℓ_1-regularized approach to the support vector machine.

Finally, there is a wealth of approaches for feature selection in the framework of multiple kernel

learning. Once more, by defining low-rank kernels on small groups of features, this approach effectively performs feature selection. Some early propositions include [LCB+04] where semi-definite programming was used to learn the kernel matrix. Multiple refinements to this approach have been presented, including [BLJ04], where sequential minimal optimization (SMO) techniques were utilised for greater efficiency, [SRSS06] which relies on a semi-infinite linear programming formulation and [RBCG08] which presents a simple approach which is essentially based on gradient descent. More recently an approach relying on the optimization of KTA was proposed in [CMR12] which utilises existing SVM solvers.

1.2.4 Meta-Selection Approaches

The previous sections presented the three major feature selection approaches. By meta-selection strategies, we refer to feature selection regimes which intelligently exploit properties of these three basic feature selection approaches. Two methods that exemplify this approach are BoLasso [Bac08] and Stability Selection [MB10]. Both methods rely on bootstrapped Lasso estimates, exploiting the fact that relevant variables will enter the model with substantially higher probability than those that are irrelevant. This property suggests that with a large number of bootstrapped lasso estimates of a given sample, intersecting their supports leads to improvements in selection consistency. This simple idea is further elaborated in [MB10], where the approach is extended to a more general framework, that combines subsampling with sparse selection algorithms. A significant contribution of stability selection is its ability to provide probabilistic guarantees for the false discovery rate of the selected features. Examples illustrating the need for meta-selection algorithms are given in [DET06] and [CT07], where the irrepresentable condition for the Lasso and related algorithms is introduced. The irrepresentable condition is violated in scenarios where the number of variables exceeds the number of samples, and a substantial degree of between-variable correlation is present leading to inconsistent selection.

1.3 Application to biomarker discovery

Biomarkers are small, statistically validated sets of variables that strongly characterize an underlying biological process. High throughput biological screening methods such as micro-arrays and time-of-flight mass spectrometry capture large numbers of variables related to biological samples. Through systematic comparison to appropriately chosen reference samples it is possible to discover, and statistically validate disease specific proteome patterns. This is a process where feature selection plays a major role. Feature selection is pivotal in shifting the focus from a very large number of variables to a substantially smaller number of of features, which at this point are considered as biomarker candidates.

Feature selection is a generic methodology that needs to effectively address the already large number of extant technologies in biology as well as anticipate the advent of new technologies. In so doing, these algorithms need to retain their connection to the biological context of the problem. An example of this is [AFRP+06] where feature selection on mass-spectrometry data from a Tuberculosis(TB) study, resulted in the identification of novel biomarkers. The impact of accurate algorithms is not limited to just novel diagnostics. It is often the case that biomarkers can reveal a vast amount of information on the

underlying biological process, as well as providing a target for therapeutic approaches to focus on.

In terms of practical considerations, feature selection algorithms need to address a series of issues. They need to be computationally efficient in order to address the ever-increasing size of the datasets. All the other desiderata also apply, in terms of consistency, parsimony and generalization. Additionally it needs to address nuances specific to biology. Principal among these is the variance samples can exhibit. It is often the case to encounter a substantial degree of variation even for replicates of the same biological sample. A further complication comes in the form of label noise,as the classes which correspond to diagnoses on the biological samples are not always accurate. Robustness towards these factors is important.

1.4 Structure of the thesis

The rest of the thesis is structured as follows.

1. *Chapter 2* provides the necessary background on max-margin learning and feature selection assumed throughout the rest of the thesis.

2. *Chapter 3* examines variants of stability selection better tailored towards classification and introduces the greedy maximization of kernel target alignment as a model selection criterion.

3. *Chapter 4* presents a randomized feature selection algorithm for nonlinear feature selection and provides empirical comparisons with other non-linear feature selection variants.

4. *Chapter 5* studies the use of feature selection on representation learning, providing very encouraging results.

5. *Chapter 6* concludes the thesis by summarizing the presentation and identifying avenues for future research.

Chapter 2

Background: Enabling Technologies

This chapter establishes the background and introduces the necessary notation for the following chapters of the thesis. We begin with the introduction of linear learning. We establish some characteristics of linear classification such as margins, as well as illustrating how kernels are utilised to address the limitations of linear learning. Going back to feature selection, algorithms that exemplify the approaches mentioned in the previous chapter are presented. Finally, we introduce metrics and the experimental pipeline for evaluation of the chapters to follow.

2.1 Linear Learning

We consider the supervised learning problem of modelling the relationship between a $m \times n$ input matrix $\mathbf{X}$ and a corresponding $m \times n'$ output matrix $\mathbf{Y}$. The archetypical instance of such a problem is binary classification where the objective of the learning problem is to learn a function $f : \mathbf{x} \rightarrow \mathbf{y}$ mapping input vectors $\mathbf{x}$ to the desired outputs $\mathbf{y}$. In the binary case we are presented with a $m \times n$ matrix $\mathbf{X}$ and a vector of outputs $\mathbf{y} \in \{+1, -1\}^m$ Limiting the class of discrimination functions to linear classifiers we wish to find a classifier :

$$f(\mathbf{x}) = \sum_i w_i x_i + b = \langle \mathbf{w}, \mathbf{x} \rangle + b \tag{2.1}$$

where the weight vector $\mathbf{w}$ and bias b are parameters derived by the learning algorithm.

Beneath the simplicity of the above equation lies the most fundamental problem for most machine learning applications. The problem is that in most practical settings, learning is ill-conditioned. In simple terms this can mean that the number of samples is insufficient to guarantee the discovery of an accurate prediction rule. In the presence of noise, aiming to perfectly fit the training set often leads to overfitting, whereby the prediction function accounts for noise in the training set instead of learning a meaningful representation of the underlying problem.

Simple linear algebra guarantees that with enough variables it is possible to perfectly fit any dataset. This raises the question of how to choose from competing hypotheses that 'look good' on the data set at hand, and how well do these competing hypotheses mirror the properties of the underlying process. Regularization is an answer to the first question. Regularization places additional restrictions to the weight vector, with the most common approach being to favour solutions that have small norms. Learning theory attempts to answer the latter question, and SVMs combine these two insights to create an effective

algorithm.

2.2 Max margin classification and support vector machines

The signed distance of a point $\mathbf{x}$ from a hyperplane $(\mathbf{w}, b)$ is $\langle \mathbf{w}, \mathbf{x} \rangle + b$. The margin is the minimum distance between the convex hulls of the negative and positive training points. We denote as $\mathbf{x}^+, \mathbf{x}^-$, two samples belonging to margins of the positive and negative class respectively. For a weight vector $\mathbf{w}$ realising a functional margin of 1 on the positive point $\mathbf{x}^+$ and the negative point $\mathbf{x}^-$ we have

$$\langle \mathbf{w}, \mathbf{x}^+ \rangle + b = 1$$

$$\langle \mathbf{w}, \mathbf{x}^- \rangle + b = -1$$

To compute the geometric margin γ, $\mathbf{w}$ must be normalised. The geometric margin γ is the functional margin of the resulting classifier

$$\gamma = \frac{1}{2}(\langle \frac{\mathbf{w}}{||\mathbf{w}||_2} \mathbf{x}^+ \rangle - \langle \frac{\mathbf{w}}{||\mathbf{w}||_2} \mathbf{x}^- \rangle) = \frac{1}{||\mathbf{w}||_2} \tag{2.2}$$

Equation (2.2) introduces the relation between the weight vector $\mathbf{w}$ and achieved margin γ. In the linearly separable case, the SVM algorithm attempts to recover the maximal margin hyperplane $\mathbf{w}$ through the following optimization problem:

$$\begin{gathered} \textbf{minimize}_{w,b} \frac{1}{2}||\mathbf{w}||^2 \\ \\ \text{subject to} \\ y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \\ i = 1, ..., m \end{gathered} \tag{2.3}$$

Introducing slack variables to the above optimization problem we can obtain the soft-margin SVM formulation [CV95] :

$$\begin{gathered} \textbf{minimize}_{w,b,\xi} \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{n} \xi_i \\ \\ \text{subject to} \\ y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \\ i = 1, ..., m \\ \xi_i \geq 0 \end{gathered} \tag{2.4}$$

In the above formulation, the regularization parameter C controls the balance between goodness-of-fit and the weights of the recovered solution. Considering that the slack variables ξ_i are equivalent to the hinge loss function $L_{hinge} = \max(0, 1 - y_i \langle w, x_i \rangle)$, equation (2.4) exemplifies an approach we will encounter again, namely minimising a loss function under additional weight restrictions. It is important to note that for most practical scenarios, a large number of the slack variables ξ will be zero. The discovered classification rule only depends on the few samples that lie on the margin, for which the

inequality constraint is active and therefore $a_i > 0$. These samples are called support vectors, lending their name to the algorithm.

An examination of the max-margin approach would be incomplete without some mention of the wealth of learning theory relating the margin γ with generalization. Indeed, a large body of work is dedicated to establishing bounds for generalization performance of support vector machines in terms of the margin distribution. In [STBWA98], it was shown that margins and the number of support vectors can be used to estimate how well the recovered classifier relates to the underlying statistical process. A functionally tighter, more data-dependent bound in variational form was presented in [LST02]. This work was further refined in [McA03], where an explicit solution to the variational problem posed in [LST02] is provided. These results, as well as a large number of publications that followed, justify the max-margin approach, providing theoretical justifications for its generalization accuracy.

2.3 Non-linear maps and kernels

The obvious drawback of linear learning is that richer, non-linear representations are often required in practical applications. Through the use of a non-linear feature map $\phi(\mathbf{x})$, the linear learning formulation can be generalized to deal with non-linear settings. This leads to the kernelized formulation:

$$f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle = \langle \sum_i a_i y_i \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle = \sum_i a_i y_i k(\mathbf{x}_i, \mathbf{x}) \tag{2.5}$$

Where $\kappa(\mathbf{x}_i, \mathbf{x}) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle$ is the kernel function for $\mathbf{x}_i$ and $\mathbf{x}$. The kernel formulation is enabled by the representer theorem. The representer states that the weights $\mathbf{w}$ can be expressed as a convex combination of the projections of the input vectors.

$$\begin{gathered} \mathbf{w} = \textstyle\sum_i a_i \mathbf{y}_i \mathbf{x}_i \\ \text{or in the case of using a feature map } \phi(\mathbf{x}) \\ \mathbf{w} = \textstyle\sum_i a_i \mathbf{y}_i \phi(\mathbf{x}_i) \end{gathered} \tag{2.6}$$

2.3.1 SVM Dual

Through use of the representer theorem we obtain the dual SVM problem:

$$\max W(a) = \textstyle\sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \mathbf{y}_i \mathbf{y}_j \alpha_i \alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j)$$

subject to

$$\begin{gathered} \textstyle\sum_{i=1}^m \mathbf{y}_i \alpha_i = 0 \\ \alpha_i \geq 0, i = 1, ..., m \end{gathered}$$

Where $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ is the kernel between samples $\mathbf{x}_i$ and $\mathbf{x}_j$.

2.3.2 Kernel Target Alignment

Finally a key measure of interest in our work is kernel target alignment (KTA) [CMR12] which is defined as:

Definition 2.3.1.

$$a(C_x, C_y) = \frac{\langle C_x, C_y \rangle_F}{\|C_x\|_F \|C_y\|_F} = \frac{\sum_{i,j} c_{x_{ij}} c_{y_{ij}}}{\sum_{i,j} \|c_{x_{ij}}\| \sum_{i,j} \|c_{y_{ij}}\|}$$

The matrices C_x and C_y correspond to centred kernels on the features X and outputs Y and are computed as:

$$C = \left[I - \frac{11^T}{m} \right] K \left[I - \frac{11^T}{m} \right]$$

where 1*, in the above equation denotes the m-dimensional vector with all entries set equal to one.*

KTA can be seen as an extension of correlation to kernel induced feature spaces, effectively measuring the degree of agreement between two kernels. Centering confers a number of benefits over the use of the original, uncentered kernels. It removes the effect of having a large expected value in the input matrix and it additionally facilitates estimating the alignment of datasets with imbalanced class proportions, as the interclass ratio can affect the estimated covariance of the input kernel with the kernel defined on the labels when the kernel matrices are uncentered. This was illustrated in [CMR12](p. 20 fig 1), where it was contrasted to the uncentered definition of alignment. The denominator normalization in the alignment estimation is useful for using unbounded kernels, as scaling can affect their alignment score. It is not however necessary when dealing with bounded kernels such as the gaussian. Additional discussion of centering, as well as its necessity for convergence of alignment to the covariance operator in feature space can be found in [SSG+12].

2.4 Feature Selection For SVMs

Chapter 1 provided motivations for employing feature selection approaches and gave an overview of algorithms commonly used in practice. The following sections will cover how some representatives of the feature selection approaches can be combined with SVMs. After covering some baseline approaches we provide some criteria for their systematic evaluation and briefly describe the experimental pipeline used in their comparison.

We have already mentioned that finding the minimal approximately sufficient feature set is a NP-hard problem. Depending on the different criteria to optimize an approximate solution and the search strategies employed to achieve that objective we identify three principal approaches towards solving the feature selection problems, namely *Filter Methods*, *Wrappers*, and *Embedded Methods*.

The following sections present three methods representative of each approach. Our aim, is to provide sufficient intuition as to how these broad classes of algorithms work and provides a few simple benchmarks for practical comparison with the methods we will be proposing.

2.4.1 Filtering

Statistical tests are often used by filter methods. Simple statistical approaches that employ Correlation Coefficients as a measure of linear dependence between variables such as [vVDVDV+02] are ubiquitous in the biological domain.

Definition 2.4.1. *For a set of samples $(x_1, y_1), \ldots, (x_N, y_N)$ of variables **X**, **Y**, with an empirical estimate of population means $\bar{x}$ and $\bar{y}$ respectively the pearson correlation coefficient, denoted by r is:*

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 (y_i - \bar{y})^2}}$$

The result, $r \in [-1, +1]$ denotes the signed magnitude of linear codependence between the variables. This can be used effectively as a simple feature selection mechanism by ignoring variables that display a lower degree of covariance.

Assuming that we have an $m \times n$ dimensional matrix of inputs $\mathbf{X}_{train}$ and corresponding output patterns $\mathbf{Y}_{train}$ as well as validation and test patterns $\mathbf{X}_{val}$, $\mathbf{Y}_{val}$ and $\mathbf{X}_{test}$ $\mathbf{Y}_{test}$ we would like to find the combination of features that empirically results in the highest accuracy while relying upon the smallest number of features.

We can attempt to obtain this combination of features empirically by the following procedure:

For each variable $\mathbf{X}_i$ in the training set $\mathbf{X}_{train}$calculate the magnitude of its correlation with the target $\mathbf{Y}_{train}$ as:

$$r_i = \left| \frac{\sum_j (x_{ij} - \bar{x}_i)(y_j - \bar{y})}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2 (y_j - \bar{y})^2}} \right|$$

From a computational standpoint, the calculation of correlation coefficients is very efficient having a complexity of $O(nm)$. However using this method is effective when the decision rule is linear and does not account for colinearities or other types of variable interactions which is a limitation in terms of applicability.

2.4.2 Wrappers

A typical example of a wrapper method is *Recursive Feature Elimination*, [GWBV02] which is a considerably better fit for svms. Starting with the full set of features, RFE at each iteration greedily eliminates the variable that has the least contribution in the classification rule. In the primal case, this is equivalent to removing the variable that has been assigned the least weighting from the learning rule. When using kernels, the sensitivity to each variable is calculated as the following:

Definition 2.4.2.

$$\left| \|\boldsymbol{w}\|^2 - \|\boldsymbol{w}^{(i)}\|^2 \right| = \frac{1}{2} \left| \sum_{k,j} a_k^* a_j^* \boldsymbol{y}_k \boldsymbol{y}_j \kappa(\boldsymbol{x}_k, \boldsymbol{x}_j) - \sum_{k,j} a_k^{*(i)} a_j^{*(i)} \boldsymbol{y}_k \boldsymbol{y}_j \kappa^{(i)}(\boldsymbol{x}_k, \boldsymbol{x}_j) \right|$$

Where $\kappa^{(i)}(\boldsymbol{x}_j, \boldsymbol{x}_k)$ is the kernel function value for samples $\boldsymbol{x}_j$ $\boldsymbol{x}_k$ when the i_{th} feature is removed.

The sensitivity of the learning rule to the removal of a specific variable is used in practice by removing the variable that least affects the learning rule at each iteration of the algorithm. Using this we can obtain a simple methodology that ranks variables by the sequence in which they are removed through the following method:

Having ranked the variables according to the sequence they are removed by the above procedure, an empirical estimate of the algorithm's performance can be obtained in a similar way to the one indicated

Algorithm 1 RFE

Input: input data X, labels Y, kernel function $\kappa(x, x')$ and regularization parameter c

Initialize: $\bar{X} = \mathbf{X}_{train}$

repeat

Train a SVM on the set of features $\bar{X}$

for $i = 1$ **to** $|\bar{X}|$ **do**

Evaluate the criterion:

$$r_i = \left| \|\mathbf{w}\|^2 - \|\mathbf{w}^{(i)}\|^2 \right| = \frac{1}{2} \left| \sum_{k,j} a_k^* a_j^* \mathbf{y}_k \mathbf{y}_j \kappa(\bar{\mathbf{x}}_k, \bar{\mathbf{x}}_j) - \sum_{k,j} a_k^{*(i)} a_j^{*(i)} \mathbf{y}_k \mathbf{y}_j \kappa^{(i)}(\bar{\mathbf{x}}_k, \bar{\mathbf{x}}_j) \right|$$

end for

remove the variable $\bar{X}_i$for which r_i is minimum

until $\bar{X}$=$\{\}$

Return: Sequence of removed variables

previously for the use of correlation of coefficients. In practice, RFE variants often remove more than a single variable at a time improving the running time of the algorithm at the potential cost of arriving at an inexact solution.

A number of studies have highlighted the method's capacity to eliminate redundancy and yield substantial improvements in generalization accuracy. However, there are some important drawbacks when we consider our desiderata for an effective feature selection mechanism. First and foremost is the fact that especially in the kernelized case RFE is computationally prohibitive having a complexity of $O(max(n, m)m^2)$ compared to $O(nm)$ for Pearson correlation, while it can also be prone to overfitting. Its computational complexity has ramifications in any attempt to use RFE on a large dataset, while the possibility to overfit can affect both the generalisation ability of the resulting classification rule and the consistency of the algorithm when we consider the possibility of label noise. By label noise we mean the existence of a number of inaccurate class labels in the dataset, as a result of error in earlier parts of the experimental design.

2.4.3 Embedded Feature Selection

The Lasso algorithm [Tib96] finds a least-squares solution with additional constraints on the L_1-norm of the parameter vector and can be formulated as finding the solution to the following optimisation problem:

Definition 2.4.3.

$$minimise \|\boldsymbol{X}w - \boldsymbol{Y}\|_2^2 + \lambda \|w\|_1$$

This is another example of the interplay between goodness-of-fit and restrictions on the weight vector $\mathbf{w}$ we previously noted with the SVM algorithm. There are important distinctions to be made however. The first one is that Lasso attempts to minimise the square loss $L_{square}(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$, which is more amenable to regression problems. Secondly, but more importantly in terms of feature selection, the L1-regularization used by the Lasso, implicitly performs feature selection. This is due

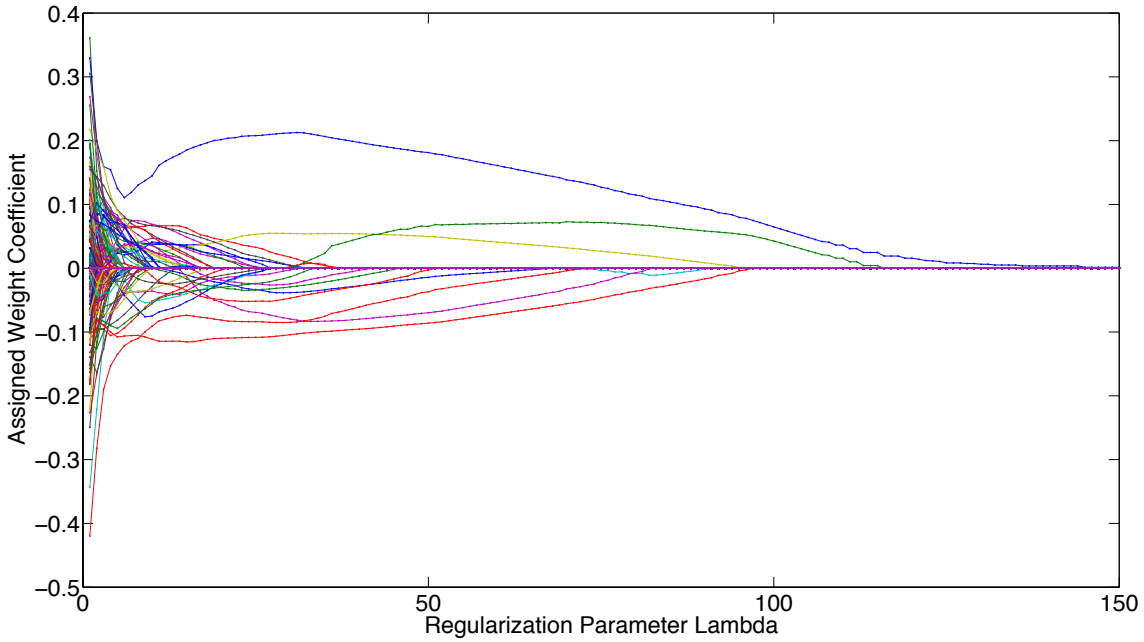

Figure 2.1: regularization path for lasso on TB dataset. The TB dataset comprises 523 features. Here, each line represents the computed weight-coefficients for each variable at varying levels of regularization. It can be seen that for increasing amounts of regularization, controlled by the parameter λ, the 1-norm of the weight $\mathbf{w}$ decreases, resulting in models that rely on fewer variables. The λ parameter essentially controls the exchange between sparsity and goodness of fit for the Lasso problem.

to its tendency to prefer sparser solutions, thus reducing the number of variables upon which the given solution is dependent. Still, this approach provides a transparent method of controlling the sparsity-accuracy tradeoff of the solution through tuning the λ parameter. The computation of the lasso solutions can be tackled with standard convex optimisation tools, or by customised solvers such as the Least Angle Regression (LARS)[Efron,2004] algorithm. An advantage of using the LARS algorithm is the fact that it can compute the entire path of solutions for every possible λ with time requirements that do not exceed those of obtaining a least squares estimate.

2.4.4 Meta-Selection

The selection consistency of the Lasso estimator has come into focus thanks to work examining the irrepresentability condition such as [DET06], and [CT07]. Work on the irrepresentability condition of the Lasso has illustrated that in settings where there are many highly correlated input variables, the Lasso estimator is only asymptotically identical to the true underlying pattern. In simple terms this means that when the experimental design includes large numbers of correlated variables, the Lasso estimator may exhibit inconcistency by only selecting the most correlated variable from a group of highly inter-related variables. A scenario from computational biology that illustrates this scenario comes from estimating the impact of peptide families for a feature selection problem. For example cytokines are a protein family which are often present in inflammatory reactions. When trying to estimate the dependence of the underlying biological process on cytokines, the Lasso estimator in this scenario could simply select

the cytokine that appears to have the largest correlation with the process, ignoring other, potentially informative peptides.

The following definition of the irrepresentability condition adheres to the exposition of [ZY06]. Let w denote the recovered weight vector, with q non-zero entries,$w_j \neq 0$, $j \in \{1, \ldots, q\}$, $w_j = 0, j \in \{q+1, \ldots, n\}$. Let $w_{(1)} = (w_1, \ldots, w_q)^T$ and denote $X_{(1)}$ and $X_{(2)}$ as the first q and $n-q$ columns of X, and let $C = m^{-1}X^TX$. By setting $C_{11} = m^{-1}X_{(1)}^TX_{(1)}$ and $C_{21} = m^{-1}X_{(2)}^TX_{(1)}$, the irrepresentable condition holds if there exists a number θ, $0 < \theta < 1$, such that

$$\|C_{21}C_{11}^{-1}sign(w_{(1)})\|_\infty \leq \theta$$

What we deem meta-selection approaches, typically address this problem by combining a sparse selection algorithm with bootstrapping. Such an approach is stability selection [MB10], which introduces a very general framework for variable selection and structure estimation. During each iteration (bootstrap) , the sparse selection algorithm is presented with a subsample of the data. By repeating this process and keeping track of the number of times each variable was used over the different iterations this framework builds an estimate for the importance of each feature.

Algorithm 2 Stability Selection

Input: input data $\mathbf{X}$, labels $\mathbf{Y}$, regularization parameter λ, and a number of bootstraps $n_{bootstraps}$

for $i = 1$ **to** $n_{bootstaps}$ **do**

randomly select a subsample $\mathbf{X}^{(i)}$ and corresponding output values, $\mathbf{Y}^{(i)}$ of the original dataset $\mathbf{X}$ and $\mathbf{Y}$

solve the lasso problem on the ith bootstrap :

$$minimise\|\mathbf{X}^{(i)}\mathbf{w} - \mathbf{Y}^{(i)}\|_2^2 + \lambda\|\mathbf{w}\|_1$$

if $w_j \neq 0$ **then**

Increment the frequency counter for the jth feature by 1.

end if

end for

Return: Inclusion frequencies for variables

Once this process is completed, model selection can be completed in the same way as with the other methods, using the selection frequency for each variable as a ranking criterion, or setting a threshold on the frequency of a variable's selection. What is particularly important however, is the fact that this approach provides a bound for the expected number of falsely included variables V. For a regularization parameter λ, the bound takes the following form [MB10]:

$$E(V) \leq \frac{1}{2\pi_{thr} - 1}\frac{q_\lambda^2}{p} \tag{2.7}$$

Where:

- $E(V)$ is the expected number of falsely included variables.
- π_{thr} is the cutoff frequency for the included variables.
- q_λ is the number of selected variables with inclusion frequency at least π_{thr}
- p is the number of variables

2.5 Evaluation

Here, we give a simple overview of how we evaluate various feature selection algorithms. The introduction outlines four particular desiderata for evaluation, namely generalization accuracy, parsimony, selection consistency and scalability. In this section, we proceed to enumerate direct, or surrogate metrics that mirror these properties.

A large number of experiments were performed on synthetic datasets to establish the empirical performance of various feature selection approaches. The use of synthetic datasets enables the production of more comprehensive results. Along with the accuracy on the test set and the sparsity we also record the precision and recall of the selection algorithms. Analogously to information retrieval, we define the precision as the number of the relevant features that were selected from the feature selection procedure over the total number of features selected and recall as the number of relevant features selected over the total number of relevant features.

Definition 2.5.1.

$$\textbf{\textit{Accuracy}} = \frac{\textit{Number of Correctly Classified Samples}}{\textit{Total Number of Samples}}$$

$$\textbf{\textit{Precision}} = \frac{\textit{Number of Correctly Identified Relevant Variables}}{\textit{Total Number of Identified Variables}}$$

$$\textbf{\textit{Recall}} = \frac{\textit{Number of Correctly Identified Relevant Variables}}{\textit{Total Number of Relevant Variables}}$$

We include two evaluation metrics that encapsulate the consistency properties of the various algorithms we have tested. The first one is based on the mean variance of the selection of individual components represented as binary indicator variables.Letting w'_{ij} be a binary indicator variable that indicates that feature j was selected during validation i we calculate the mean variance over different variables as

$$\frac{var(w'_j)}{n}$$

The mean variance however is biased towards non-sparse solutions. In order to account for this bias we also compute an upper bound on the probabilities of obtaining a particular set of selected weights in the following way.

Consider k-fold cross validation and let $s_1, \ldots, s_k$ be the size of the sets of features identified in each fold. Furthermore let s_u be the size of the union of features selected in at least one fold. For the fixed set of indices in the union, for a random choice of indicies the probability of set s_i all falling in this

set is $(s/n)^{s_i}$. Taking a union bound over all of the possible choices of the intersection of size s gives:

$$\binom{n}{s_u}(s_u/n)^{\sum_i s_i}$$

Hence log probability is (2.8)

$$\log(s_u/n)\sum_i s_i + n\log n - s_u \log s_u - (n-s_u)\log(n-s_u)$$

Where we have used Stirling's approximation for $\ln(n!) \approx n\ln(n) - n$

Figure 2.2 presents a number of scenarios to illustrate the two different methods of evaluating the consistency of selected features,where the relevant features are selected with probability $1, 0.9, \ldots, 0.1$. Figure 2.3 illustrates that mean variance is a poor indicator for the consistency of the recovered features, as solutions that include the relevant features with high probability, often have higher mean variance than their low-probability counterparts(for an example of this situation see figure 2.3). Figure 2.4 illustrates how our proposed criterion has a much better correspondence to including a consistent, small number of variables.

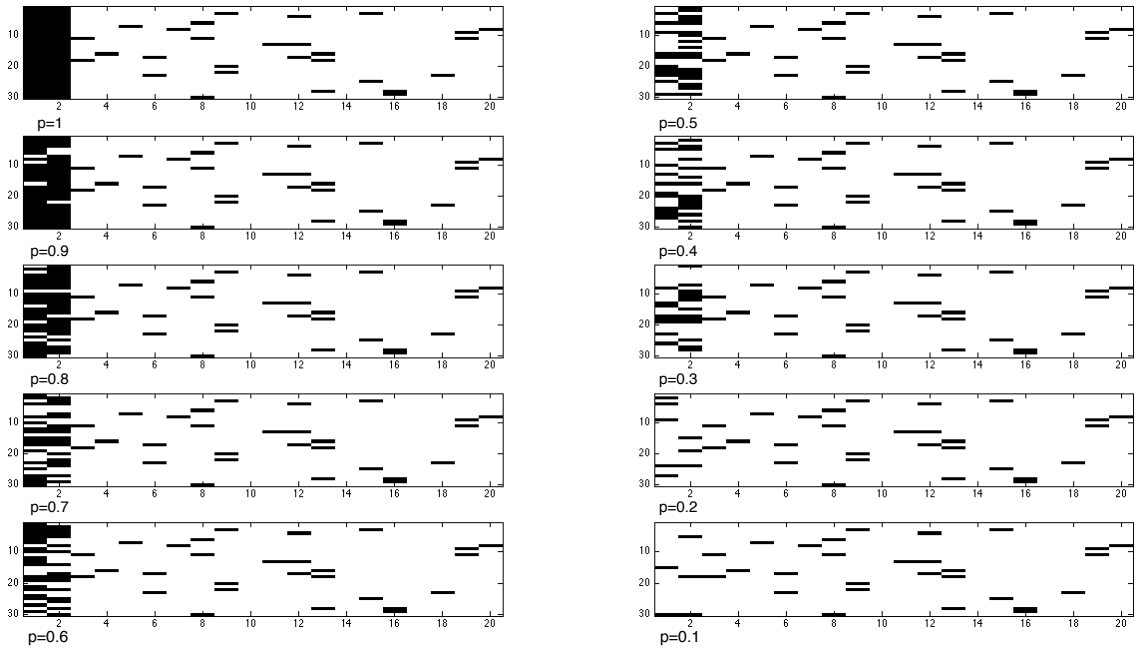

Figure 2.2: Examples of selected features over 30 cross-validation folds for a toy 20-dimensional dataset. Black bars indicate that the the feature was selected for that fold(features:left-to-right, folds: top-to-bottom). We consider the first two features to be relevant, and selected with varying probability between 1 and 0.1. The remaining, features are selected uniformly at random with a probability of 0.05.

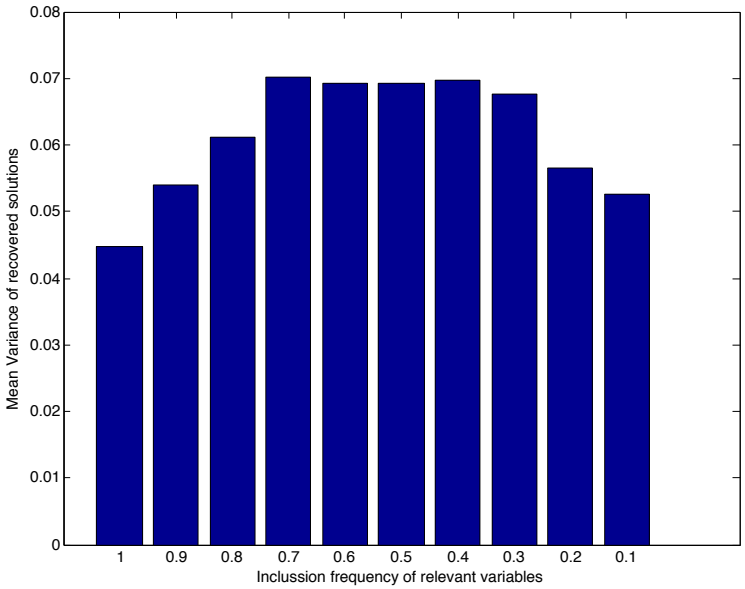

Figure 2.3: Example Mean Variance for different inclusion probabilities of the relevant features in the scenarios of figure 2.2. This figure illustrates that only estimating the variance of a feature can be misleading. In this example, selecting the two relevant features with probability 0.9 appears to have larger mean variance than selecting them with probability 0.1.

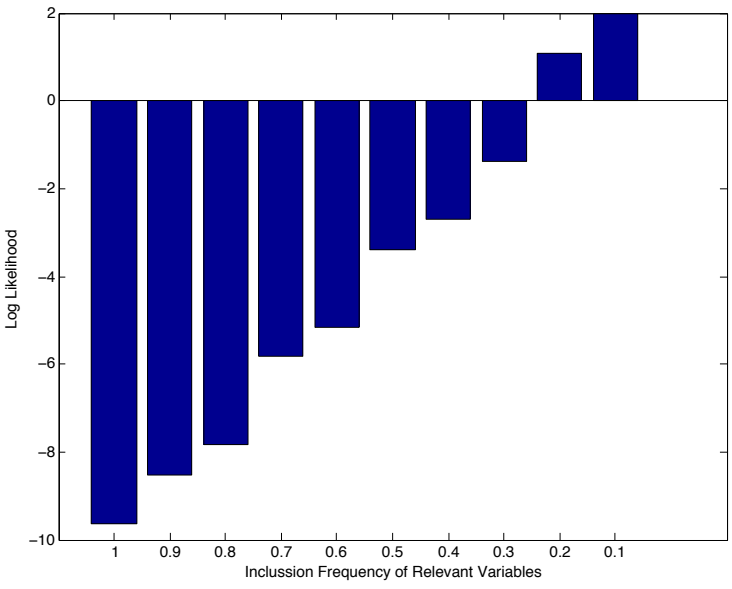

Figure 2.4: Plot for the log likelihood corresponding to the different selection probabilities of the relevant features. The correspondence between log likelihood and the high selection rate of the relevant components is substantially improved. For selection probabilities 0.2 and 0.1 the log likelihood is positive, as we are estimating an upper bound of the true probability.

2.5.1 Experimental Set Up & Cross-Validation

The following chapters will entail a substantial ammount of experimentation on both artificial and real datasets. Unless otherwise stated, the results reported for these experiments will rely on 10-fold cross-validation. For the sake of completeness we will review this approach here:

1. Split the dataset into a training and testing set for the external validation loop. Typically the data are divided in 10 folds.
2. Split the inner loop's dataset into a training and a testing set.
3. Perform feature selection on the inner-loop training set.
4. Validate model on the inner-loop validation set.
5. Repeat the inner process for 10 folds of the inner-loop data.
6. Select the model with the best accuracy. If two or more models are tied for accuracy, select the one that relies on the smaller number of variables.
7. Use the best parameters for variable selection and kernel width from the inner-loop to test on the outer loop's validation set.
8. Repeat this process for 10 folds of the outer loop.

Summary

This chapter presents a short review of max-margin learning and some of the theoretical justifications behind it. Kernels are introduced as a method to generalize max-margin learning to problems which are inherently nonlinear, while we also examine how feature selection algorithms fit into this framework. The chapter concludes with the introduction of the experimental pipeline that is utilised in most of the experiments for the following chapters as well as the evaluation metrics used.

Chapter 3

Stability Selection

This chapter presents an initial attempt at a three-stage technique for kernel machines, relying on a linear feature selection method combined with a gaussian kernel prediction rule. A number of factors justify a careful examination of this approach. Linear variable selection algorithms constitute the most common approach in many models. The simplicity of the resulting models, coupled with the fact that linear models are often competitive in terms of generalisation explain their ubiquity.

The approach comprises three steps. Initially, a linear feature selection method is utilised in order to rank the features. The resulting ranked list is then passed to a method that determines the number of features to be used in the prediction rule. Finally, the resulting reduced dataset is used in conjuction with a gaussian kernel SVM in order to infer the final prediction rule.

Typically, selecting the number of features to use is achieved through nested cross-validation. This chapter explores an alternative approach that utilises greedy maximization of Kernel Target Alignment (KTA) for the same purpose. Selecting the number of features to use in this approach is equivalent to greedily removing features from the ranked list until the alignment of a gaussian kernel defined on the remaining features is maximised. Recent publications ([GBSS05][SSG$^+$12] [CMR12]) have studied the theoretical properties of KTA suggesting numerous advantages. Here KTA is employed so as to avoid nesting in the validation phase, which constitutes a substantial overhead in the model selection phase, even for computationally inexpensive feature selection methods. Overall, this provides a significant advantage in terms of computational efficiency. What's more, our experimental comparison of KTA and nested cross-validation, illustrates improved concistency of the recovered subset of relevant variables, and competitive generalization accuracy for the various feature selection approaches we examine.

Another focal point of this chapter is stability selection, a meta-selection framework that was introduced in the previous chapter. Previous work on stability selection, focused on its strong probabilistic guarantees for controlling the false discovery rate. We expand the scope of inquiry by examining its performance in terms of the criteria we have previously outlined. What's more, we apply the framework with the use of sparse selection algorithms tailored towards classification, such as LPBoost and l_1-regularized logistic regression. We provide experimental comparisons of stability selection with the use of correlation coefficients and RFE.

An important aspect of the experiments is assessing the performance of this three-stage approach

to real world, and potentially non-linear datasets. We provide practical comparisons for the efficacy of linear feature selection algorithms in this setting, and establish the empirical performance of using them in conjunction with KTA. We employ real world datasets from computational biology, where the properties of consistency and sparsity are pivotal.

Our findings indicate that our classification oriented stability selection variants, are competitive with their more traditional counterparts. The combination of stability selection with kernel target alignment provides encouraging results. The computational efficiency of this approach, as well as the improved sparsity and consistency of the resulting models, make this approach an attractive option for the analysis of large datasets.

3.1 Model Selection with KTA

Section **2.5.1** outlined the use of nested cross validation as a model selection heuristic. In practice, cross validation is used in order to select the best model parameters from a predefined range as well as provide an empirical estimate of their impact on the generalization accuracy, through a repeated process of trial and error. When using cross-validation with a feature selection, nesting is necessary. This, significantly increases the computational requirements.

Here we briefly outline an alternative model selection criterion that utilises centred KTA [CSTEK01] [GBSS05] [CMR12]. Previous work in [CMR12] shows that KTA is sharply concentrated around its expected value, making its empirical value stable with respect to different splits of the data. This is a theoretical finding that was reproduced in practice on the datasets of this report. A significant practical advantage of using alignment instead of cross-validation is the fact that it is possible to avoid nesting through the use of KTA. We have previously defined the alignment $a(C_x, C_y)$ between two centred kernel matrices C_x and C_y in chapter 2 as:

$$\frac{\langle C_x, C_y \rangle_F}{\|C_x\|_F \|C_y\|_F},$$

where $\|C_x\|_F \|C_y\|_F$ is a normalization term, used for matrices that do not have an upper bound on the values of their entries. This approach to model selection is further justified by the following well-known observation that links kernel target alignment with the degree to which an input space contains a linear projection that correlates with the target.

Proposition 3.1.1. *Let P be a probability distribution on the product space $\mathcal{X} \times \mathbb{R}$, where $\mathcal{X}$ has a projection ϕ into a Hilbert space $\mathcal{F}$ defined by a kernel κ. We have that*

$$\sqrt{\mathbb{E}_{(\mathbf{x},y)\sim P,(\mathbf{x}',y')\sim P}[yy'\kappa(\mathbf{x},\mathbf{x}')]} =$$
$$= \sup_{\mathbf{w}:\|\mathbf{w}\|\leq 1} \mathbb{E}_{(\mathbf{x},y)\sim P}[y\langle \mathbf{w}, \phi(\mathbf{x})\rangle]$$

Proof:

$$\begin{aligned}
&\sup_{\mathbf{w}:\|\mathbf{w}\|\leq 1} \mathbb{E}_{(\mathbf{x},y)\sim P}[y\langle \mathbf{w}, \phi(\mathbf{x})\rangle] = \\
&= \sup_{\mathbf{w}:\|\mathbf{w}\|\leq 1} \left\langle \mathbf{w}, \mathbb{E}_{(\mathbf{x},y)\sim P}[\phi(\mathbf{x})y]\right\rangle \\
&= \left\|\mathbb{E}_{(\mathbf{x},y)\sim P}[\phi(\mathbf{x})y]\right\| \\
&= \sqrt{\int\int dP(\mathbf{x},y)dP(\mathbf{x}',y')\langle\phi(\mathbf{x}),\phi(\mathbf{x}')\rangle yy'} \\
&= \sqrt{\mathbb{E}_{(\mathbf{x},y)\sim P,(\mathbf{x}',y')\sim P}[yy'\kappa(\mathbf{x},\mathbf{x}')]}
\end{aligned}$$

This suggests alignment can be utilised in order to quantify the degree of agreement between a kernelised representation of a group of features and the target concept. The idea of alignment as a measure of statistical dependence was illustrated in [GBSS05], where the centred KTA is used as the empirical estimator for the the Hilbert Schmidt Independence Criterion.

In terms of kernel functions, the Hilbert Schmidt Independence Criterion can be expressed as

$$\begin{aligned}
HSIC(p_{\mathbf{xy}},\mathcal{F},\mathcal{G}) = \mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}[k(\mathbf{x},\mathbf{x}')l(\mathbf{y},\mathbf{y}')] \\
+\mathbb{E}_{\mathbf{x},\mathbf{x}'}[k(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y},\mathbf{y}'}[l(\mathbf{y},\mathbf{y}')] \\
-2\mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbb{E}_{\mathbf{x}'}[k(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y}'}[l(\mathbf{y},\mathbf{y}')]
\end{aligned}$$

where, $\mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}$ denotes expectation over independent pairs $(\mathbf{x},\mathbf{y})$ and $(\mathbf{x}',\mathbf{y}')$ and $k(\mathbf{x},\mathbf{x}')$, $l(\mathbf{y},\mathbf{y}')$ are kernel functions on $\mathbf{x}$ and $\mathbf{y}$ respectively.

This, combined with a ranking of features resulting from a feature selection procedure, leads to the simple model selection procedure described in algorithm 3, where HSIC is used to select how many of the ranked features to use. The results of the process outlined in algorithm 3 depend strongly on the selection of an appropriate parameter σ for the gaussian kernel that is used for the computation of the alignment. In practical settings, a large range of numbers is explored for selecting the parameter σ as this strongly depends on the dimensionality of the dataset under consideration. In the experimental settings the algorithms select the parameter σ which maximizes alignment. . This is illustrated in figure 3.1.

Using KTA as a model selection criterion as outlined in the above procedure has a number of advantages. From a practical point of view, it is a very simple process, that involves optimizing for a single quantity over the entirety of the training data. This adds the substantial benefit of avoiding nesting in the cross-validation phase. Compared to the overhead of nesting, this is a significant computational advantage. From a theoretical perspective this approach affords us with a bound on the empirical quantity of alignment [GBSS05]:

$$\|HSIC(p_{xy},\mathcal{F},\mathcal{G}) - HSIC(Z,\mathcal{F},\mathcal{G}\| \leq \sqrt{\frac{\log(6/\delta)}{\alpha^2 m}} + \frac{C}{m} \tag{3.1}$$

In the above equation HSIC denotes the Hilbert Schmidt Independence Criterion, a covariance operator introduced in [GBSS05]. The empirical estimator for HSIC is the KTA. A similar concentration property was later reported in [CMR12].

Algorithm 3 Greedy KTA Model Selection

Input: n-dimensional training set $\mathbf{X}_{train}, \mathbf{Y}_{train}$ and a test-set $\mathbf{X}_{test}, \mathbf{Y}_{test}$ and a ranked list of the features **r**, whose elements are sorted in order of increasing assigned importance according to a feature selection algorithm.

Initialize:

$a_{best} = 0$,

$r_{best} = \{r_1, .., r_n\}$

$\mathbf{K}_Y = \kappa_Y(\mathbf{Y}_{train}, \mathbf{Y}_{train})$ on the desired outputs $\mathbf{Y}_{train}$

$\mathbf{C}_Y = \left[I - \frac{11^T}{m}\right] \mathbf{K}_Y \left[I - \frac{11^T}{m}\right]$

for $i = 1$ **to** n **do**

Compute the centred reduced input kernel:

$\bar{\mathbf{X}} = \mathbf{X}_{train}[i : n, :]$

$\mathbf{K}_X = \kappa_X(\bar{\mathbf{X}}, \bar{\mathbf{X}})$

$\mathbf{C}_X = \left[I - \frac{11^T}{m}\right] \mathbf{K}_X \left[I - \frac{11^T}{m}\right]$

$a_i = \frac{\langle \mathbf{C}_x, \mathbf{C}_y \rangle_F}{\|\mathbf{C}_x\|_F \|\mathbf{C}_y\|_F}$

if $a_i \geq a_{best}$ **then**

$a_{best} \leftarrow a_i$

$r_{best} \leftarrow \{r_i, \ldots, r_n\}$

end if

end for

Use the reduced set r_{best} to obtain a prediction on $\mathbf{X}_{test}$

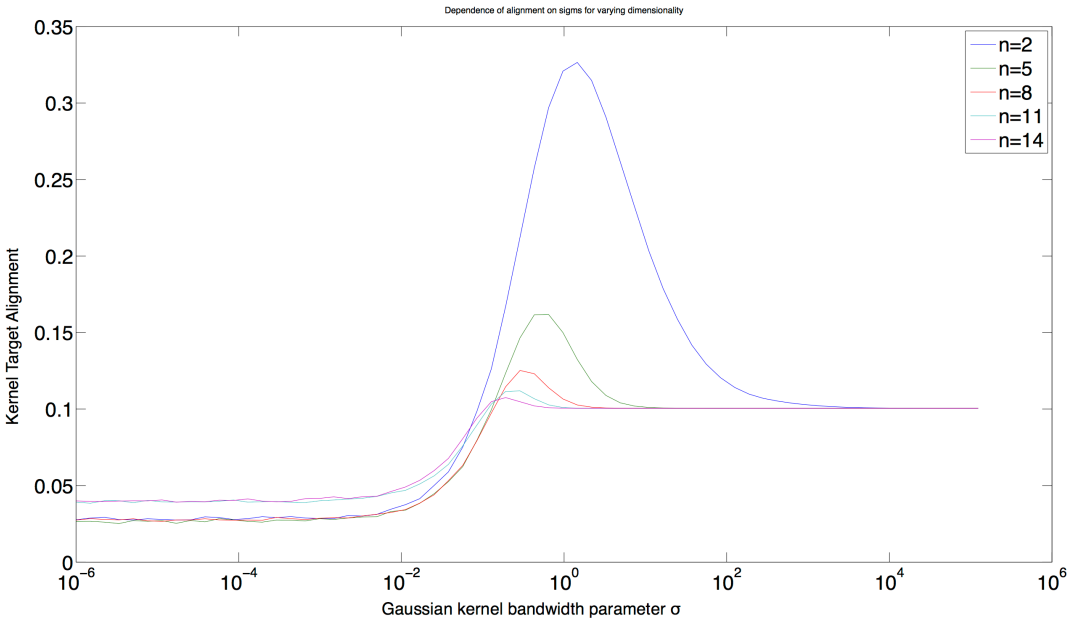

Figure 3.1: Dependence of target alignment on the σ parameter of the gaussian kernel. For increasing number of irrelevant features, smaller values of σ tend to produce higher alignment. This appears to stem from increasing the effective dimensionality, and has been also observed in ([SFG+09], sec. 5).

3.2 Extending Stability Selection

This section presents some immediate extensions to the stability selection framework, which was presented in section **2.4.4**. To reiterate, stability selection utilises bootstrapping in combination with a sparse selection algorithm and uses the inclusion frequency of the selected variables over a number of bootstraps as a feature selection criterion. In the original presentation, stability selection relies on the Lasso for sparse selection.

3.2.1 loss functions for classification

The Lasso follows the familiar pattern of minimising the sum of a loss function, $\|Xw - Y\|_2^2$ with the addition of a constraint on the 1-norm of the recovered solution $\lambda\|w\|_1$, where λ is a regularization parameter. The choice of loss function should reflect the real world semantics of the problem. For example, the square loss is a sensible choice in regression problems, where the minimisation of the discrepancy between real number values and the predicted output is the target.

The target in classification is optimizing the 0-1 loss, which is equivalent to maximising the accuracy. As direct optimization of the 0-1 loss is computationally intractable, in practice a proxy loss which is easier to optimise is used. The square loss is a valid choice of proxy for the 0-1 loss. However, as classification problems comprise a large number of the problems encountered in practice, sparse selection algorithms that are better tailored towards classification are highly desirable. Two such examples are sparse logistic regression and LPBoost.

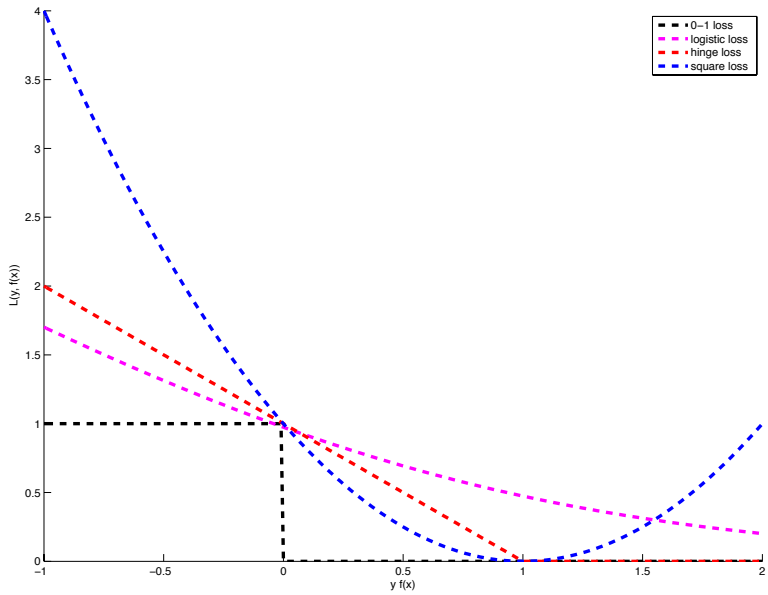

Figure 3.2: Different loss functions for classification. The lasso minimises the square loss (blue). The 0-1 loss function(black), which is directly related to accuracy, is NP-hard to optimise. Two loss functions commonly applied to classification problems are the logistic loss (magenta), and the hinge loss(red), that act as proxies for the 0-1 loss.

l_1 regularized logistic regression

For the purposes of classification we can consider the logistic model, which has the form:

$$p(y_i|\mathbf{x}_i) = \frac{\exp(y_i(\mathbf{x}_i^T\mathbf{w} + b))}{1 + \exp(y_i(\mathbf{x}_i^T\mathbf{w} + b))}$$

Where:

- $p(y_i|\mathbf{x}_i)$ is the conditional probability of y_i given $\mathbf{x}_i$
- $\mathbf{w}$ is the weight vector
- b is the intercept, or bias term

The parameters of the logistic regression function are optimised by maximising the likelihood of the training data in the model. This is equivalent to maximising the log likelihood given by the expression

$$\frac{1}{m}\sum_{i=1}^{m}\log(1 + \exp(-y_i(\mathbf{x}_i^T\mathbf{w} + b)))$$

Similarly to the approach taken for LASSO, sparsity in the case of penalised logistic regression can be enforced by adding an l_1-regularization term in the objective function, thus obtaining the following problem:

$$\text{minimize}\frac{1}{m}\sum_{i=1}^{m}\log(1 + \exp(-y_i(\mathbf{x}_i^T\mathbf{w} + b))) + \lambda\|\mathbf{w}\|_1 \tag{3.2}$$

LPBoost

Another approach suitable for classification problems would be trying to minimize the hinge loss l_{hinge}:

$$L_{hinge} = \max(0, 1 - y_i\langle\mathbf{w}, \mathbf{x}_i\rangle)$$

LPBoost [DBST02], is a sparse selection algorithm that minimises the hinge loss and can be used in the stability selection framework. LPBoost attempts to build a classification rule as a weighted sum of weak learners $h(x)$. A weak learner is a classifier that depends on a small subset of features.As an example consider the stump hypothesis on a single thresholded feature j

$$h_j(\mathbf{x}_{ij}) = \begin{cases} 1 & \text{if } x_{ij} \geq \theta \\ -1 & \text{otherwise} \end{cases}$$

Denoting by H the matrix with elements $h_{ij} = h_j(\mathbf{x}_{ij})$ LPBoost can be expressed as the following linear program:

$$\begin{gathered} \textbf{minimize}_{u,\beta}\beta \\ \text{subject to} \\ \textstyle\sum_{i,j} u_i\mathbf{y}_i H_{ij} \leq \beta \\ \textstyle\sum u_i = 1 \end{gathered} \tag{3.3}$$

By using individual features as weak predictors, LPBoost is functionally equivalent to the 1-norm, soft margin SVM.

$$\begin{gathered}\textbf{minimize}_{w,b,\gamma,\xi} -\gamma + C\sum_{i=1}^{n}\xi_i \\ \\ \text{subject to} \\ y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq \gamma - \xi_i \\ i = 1, ..., m \\ \xi_i \geq 0 \\ \|\mathbf{w}\|_1 = 1\end{gathered} \tag{3.4}$$

Where C a regularization parameter.

3.3 Experiments & Results

This section is concerned with the empirical performance characteristics of our proposals. Preceding sections presented theoretical intuitions for classification oriented variants of stability selection. Additionally, they provide some theoretical justification for the use of KTA as a model selection criterion, particularly in terms of computational economy and statistical properties. In all cases the linear selection methods are used to derive a set of features that are subsequently used in a Gaussian kernel. Here, we provide a number of experiments on artificial and real world datasets that attempt to characterize these approaches in terms of accuracy, sparsity and consistency.

To this end, we conducted a range of experiments employing Lasso stability selection and our proposed classification-oriented stability selection variants relying on l_1-logistic regression and LPBoost. Our benchmarks also employ correlation coefficients and linear RFE. In order to benchmark these approaches, we include two baseline comparisons. The first involves using the KTA model selection on a random ordering of features, and secondly a fixed randomly chosen set of 10% of the features. It must be noted that despite some small theoretical motivations, the approach of using linearly ranked features to order features for a nonlinear selection algorithm remains a heuristic, as these motivations do not guarantee the optimality of this approach.

In the case of the stability selection variants, we employed the approach for 1000 bootstraps, employing a range of different regularization parameters. We report two sets of results, resulting from our two different model selection procedures, and compare the experimental performance of the two approaches. In one approach, the final model parameters where set through nested cross-validation, while the other employs kernel target alignment for the same purpose, by choosing the kernel that maximises alignment, over the ordered set of features for a user-specified range of kernel parameters σ.

3.3.1 Synthetic Data

We generated six synthetic datasets in order to carefully illustrate the properties of the different feature selection algorithms. All six synthetic datasets contain 300 samples with a dimensionality of 200 features. With the exception of the XOR dataset, all other datasets were designed to contain 10 relevant variables for simplicity.

Along with the accuracy on the test set and the sparsity, we also record the precision and recall of

the selection algorithms. Analogously to information retrieval, we define the precision as the number of the relevant features that were selected from the feature selection procedure over the total number of features selected and recall as the number of relevant features selected over the total number of relevant features. We include results on the consistency of selected features for the synthetic experiments. Finally table 3.2 compares the decrease in time requirements for different algorithms when using KTA in place of cross-validation.

Table 3.1: Class proportions for synthetic data.

Dataset	+1	-1
Fake Class	49.33 (148)	50.67 (152)
Linear Zhang - Feature Noise	51.67 (155)	48.33 (145)
Linear Zhang - Sample Noise	48.00 (144)	52.00 (156)
Linear Weston	53.67 (161)	46.33 (139)
Non-Linear Weston	48.67 (161)	51.33 (154)
Xor	48.00 (144)	52.00 (156)

3.3.1.1 Fake Class

The Fake Class dataset benchmark (figure 3.3), can be thought of as a negative control, as no supervised feature selection method is expected to find a meaningful relationship. As such, the low performance of all methods on all fronts is to be expected. In terms of numbers of selected features for the fake class dataset, none of the two methods to selected the number of variables appears to have a distinct advantage in terms of sparsity. Precision-wise, all methods, with the exception of linear RFE used in conjuction with nested cross-validation are indistinguishable, and no better than random guessing. Most methods perform similarly with stability selection in conjunction with cross validation having a slight edge where recall is concerned. Finally in terms of mean variance and log likelihood of the selected features for the fake class dataset, models resulting from KTA appear more stableand less random than their nested cross validation counterparts. Comparing the alignment criterion on a random ranking of the features with randomly selecting 10% of the features, it appears that selecting for alignment in a random ordering includes a substantially larger number of variables, without any noticeable effects on performance.

3.3.1.2 Linear Zhang With Feature Noise

The performance picture is much improved in the linear Zhang dataset with feature noise (figure 3.4). In terms of generalization accuracy, most methods are indistinguishable, with all methods performing substantially better than randomly selecting 10% of the features, including using alignment to select a model using a random ranking of variables. The number of selected features tends to be substantially smaller for the alignment based methods, with the linear RFE producing the sparsest pattern. The reliance on larger number of features, leads the cross-validation based variants to diminished precision, whereas the alignment-based models achieve perfect precision. Conversely however, this means that the recall of the

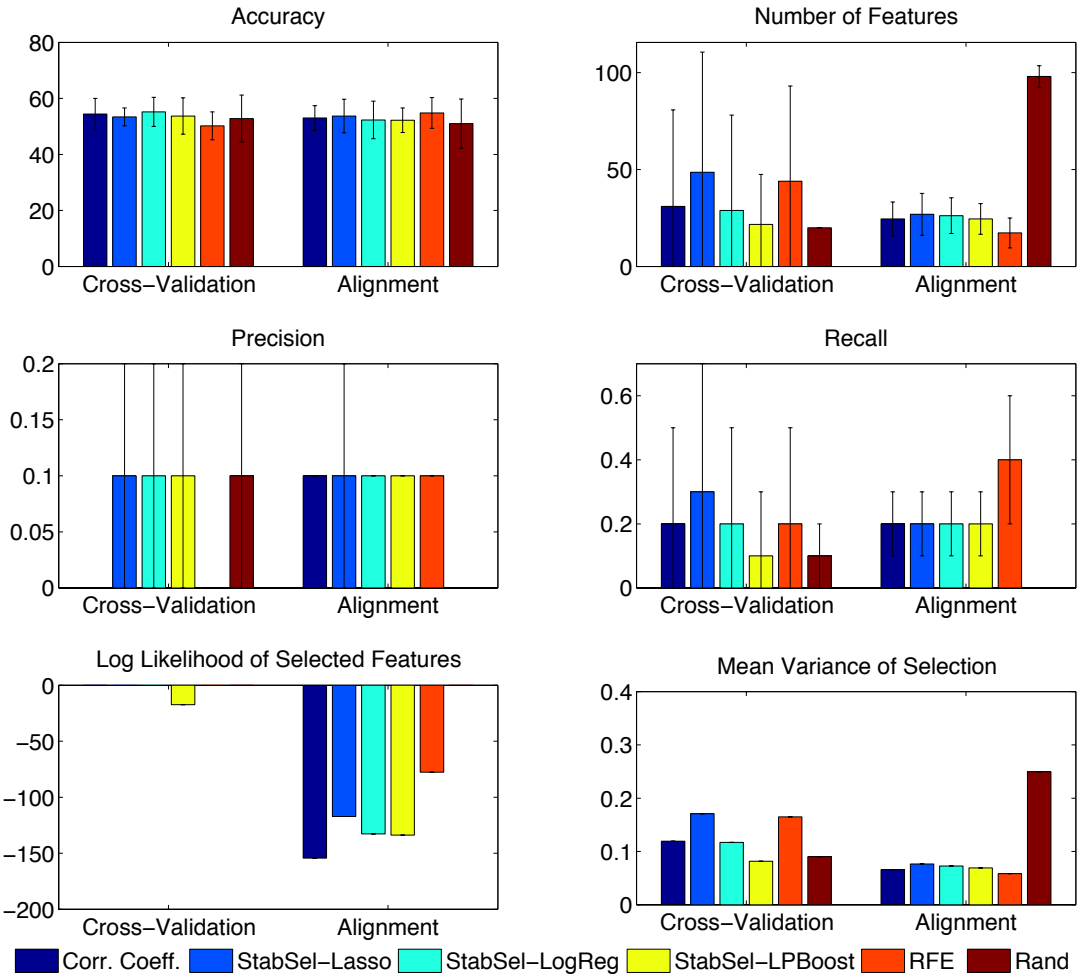

Figure 3.3: Results for the fake class dataset. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case.

alignment based models, is decreased compared to their CV-derived counterparts. Finally, KTA based models outperform their cross-validation counterparts with a large margin, in terms of consistency, and appear to have very similar variance and likelihood.

3.3.1.3 Linear Zhang - Sample Noise

The linear Zhang dataset with sample noise(figure 3.5, paints a similarly indistinguishable picture in terms of classification accuracy. Yet again, cross validation relies on more features on average and includes irrelevant features in the final model. On the other hand, alignment produces sparser models, and in three cases, (correlation coefficients, l_1-logistic regression & LPBoost) produces models with perfect precision and recall. Lasso-Stability selection and linear RFE produced sparser models, a fact which also appears to have a very small effect in their generalization performance. Alignment based models appear more stable with three methods consistently identifying all the relevant features without variation.

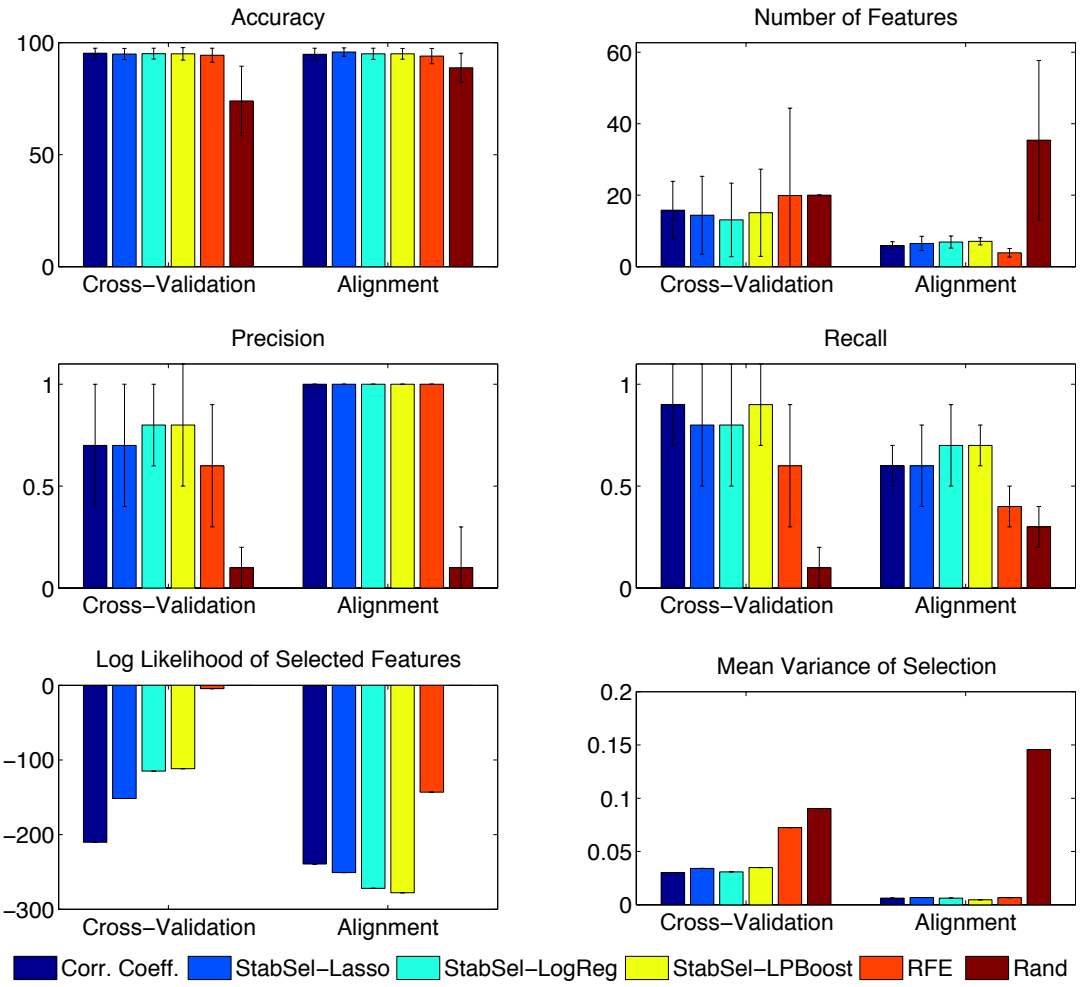

Figure 3.4: Results for the Linear Zhang with Feature Noise. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case.

3.3.1.4 Linear Weston

The linear Weston dataset (figure 3.6) provides an example were the aggressive resulting sparsity for the alignment models can have a detrimental effect in generalization accuracy. In particular, correlation coefficients and linear RFE are sparser, producing models relying on 1.3 and 1.1 variables on average respectively. However, this comes at the cost of reduced accuracy. Once again, alignment leads to perfect precision, a feat that only LPBoost stability selection reproduced in the cross validation setting. It is also interesting to note that in this benchmark, KTA based correlation coefficients and linear RFE models have substantially less consistent behaviour than their cross-validation based counterparts.

3.3.1.5 Non Linear Weston

The two nonlinear datasets provide a more interesting test scenario. In the case of the nonlinear Weston dataset (figure 3.7), with the exception of RFE, all methods perform abysmally on all captured metrics. RFE produced the sparsest solutions in both the cross validation and alignment setting, while managing to identify some of the relevant variables. In conjuction with alignment based model selection, RFE

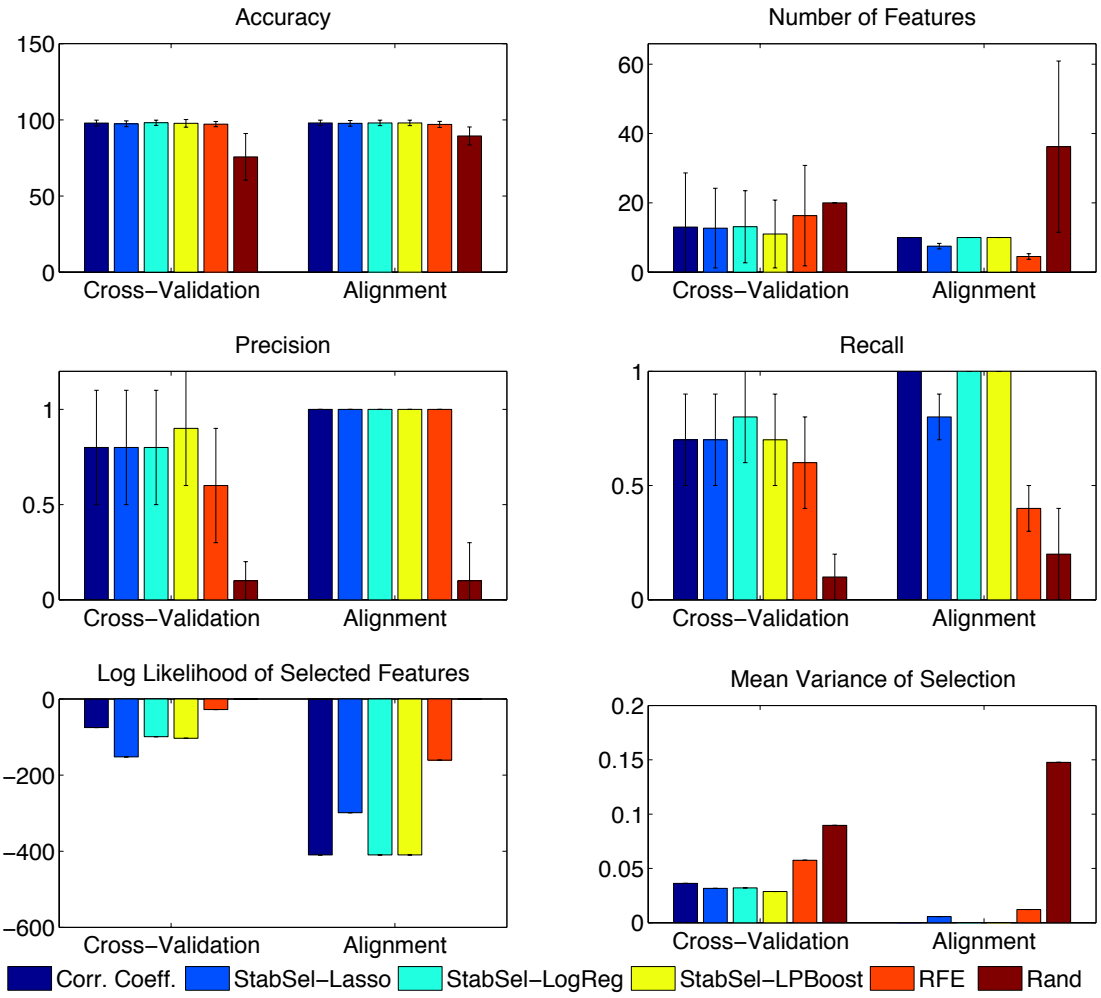

Figure 3.5: Results for the linear Zhang dataset with sample noise. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case.

benefits in terms of precision, something which is reflected in the fact that this is the clear winner in terms of both sparsity and generalization accuracy. Interestingly, using KTA on randomly ranked features appears to capture some of the dependencies, however the models also include many irrelevant features, with the higher signal to noise-ratio meaning that it fails to produce the generalization accuracy of RFE.

3.3.1.6 XOR

The case of the XOR dataset (figure 3.8) illustrates the potential benefits of using alignment as a model selection criterion. The accuracy for the models relying on alignment is significantly higher than their nested cross validation counterparts. Using alignment contributes to improved recall, which is a necessary condition for capturing the interdependence between features the model is trying to predict. While most of the nested cross-validation models are not substantially more precise from selecting variables at random, this problem is improved through the use of KTA. In contrast to the non-linear Weston dataset, RFE is the worst performer for this dataset. All approaches exhibit substantial variance, however KTA

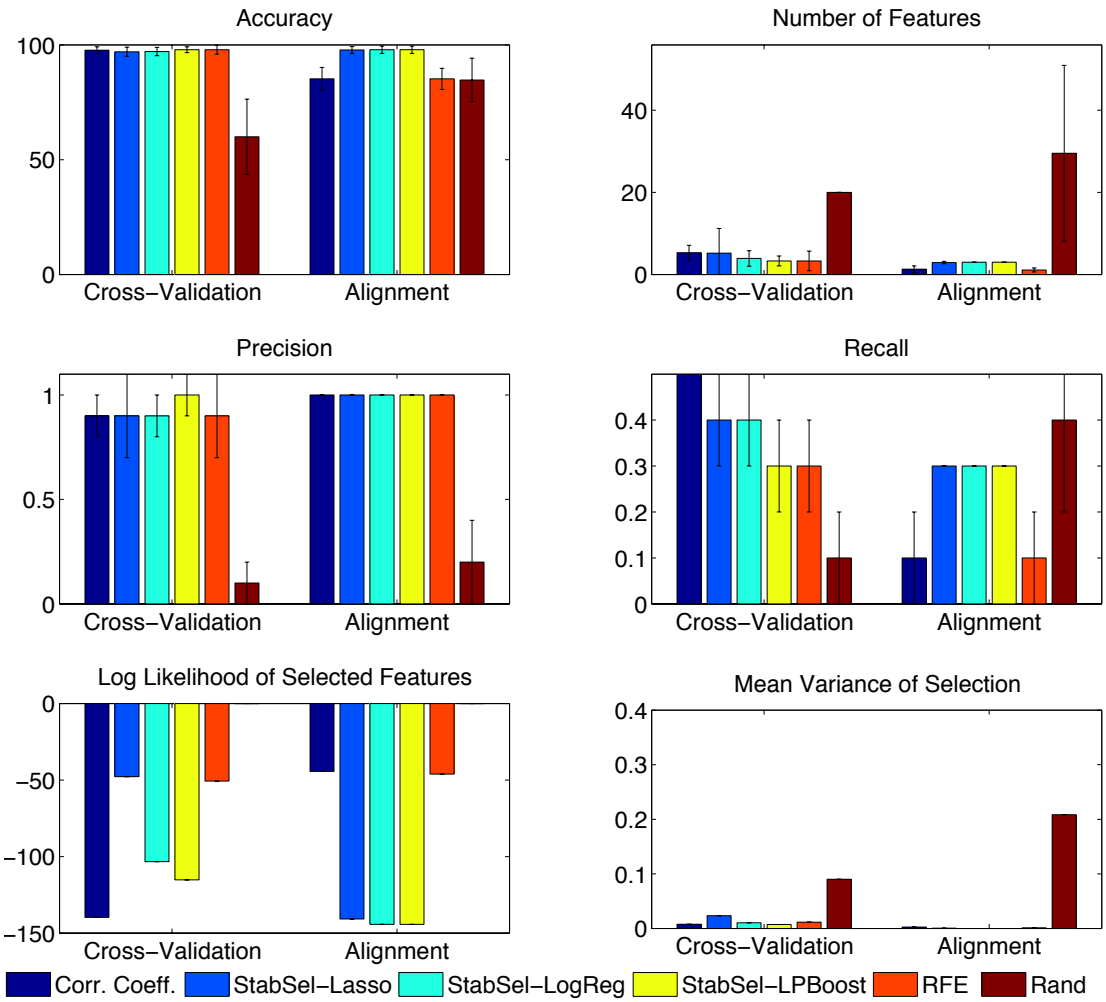

Figure 3.6: Results for the linear Weston dataset. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case.

based methods indicating a slightly improved behaviour, with LPBoost Stability Selection being the overall most stable.

Overall, KTA nearly universally improves the consistency of the recovered solutions, showing significantly smaller variance. This is strongly illustrated in the linear Zhang with sample noise and linear Weston datasets where l_1-logistic regression and LPBoost stability selection show no variation. In the case of Linear Zhang, the same behavior is exhibited by correlation coefficients. There are two important exceptions to this trend, in the case of the linear Weston (figure 3.6) dataset, where linear RFE and correlation coefficients exhibit a switching behavior in terms of the solutions they recover compared to their counterparts resulting from cross-validation. Here, the aggresive sparsity and switching behavior has catastrophic results in the generalization accuracy.

Table 3.2 provides the fold-change when employing KTA instead of nested cross-validation for model selection purposes. All algorithms benefit substantially with the stability selection variants close to the 10-fold theoretical improvement in speed.

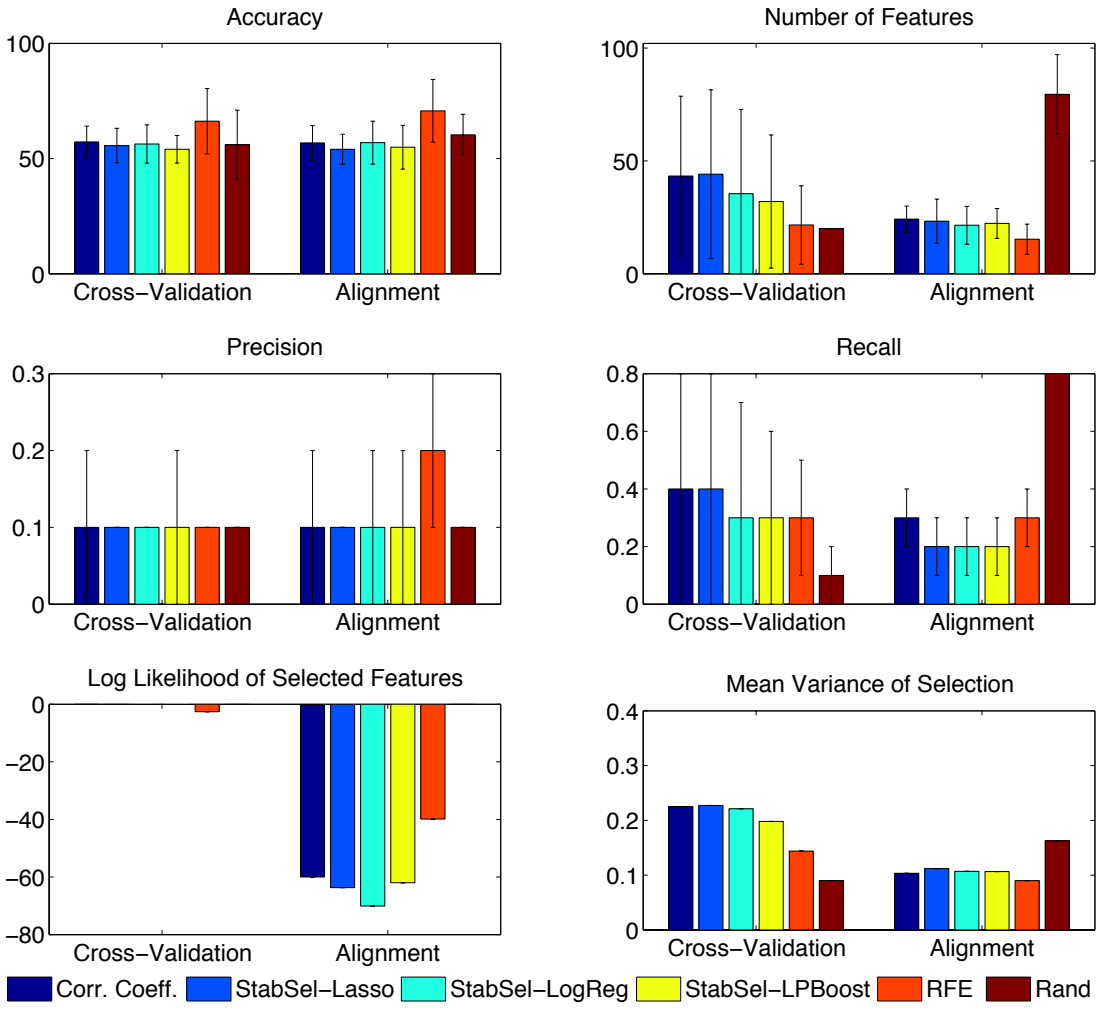

Figure 3.7: Results for the non-linear Weston dataset. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case.

Table 3.2: Fold Decrease for the time requirements of different methods when using alignment instead of cross validation

Algorithm	**Fold Decrease**
Corr Coeff.	6.05 ± 0.29
StabSel-Lasso	9.38 ± 0.70
StabSel-LogReg	9.32 ± 0.32
StabSel-LPBoost	9.9737 ± 1.26
Linear RFE	7.6779 ± 0.05

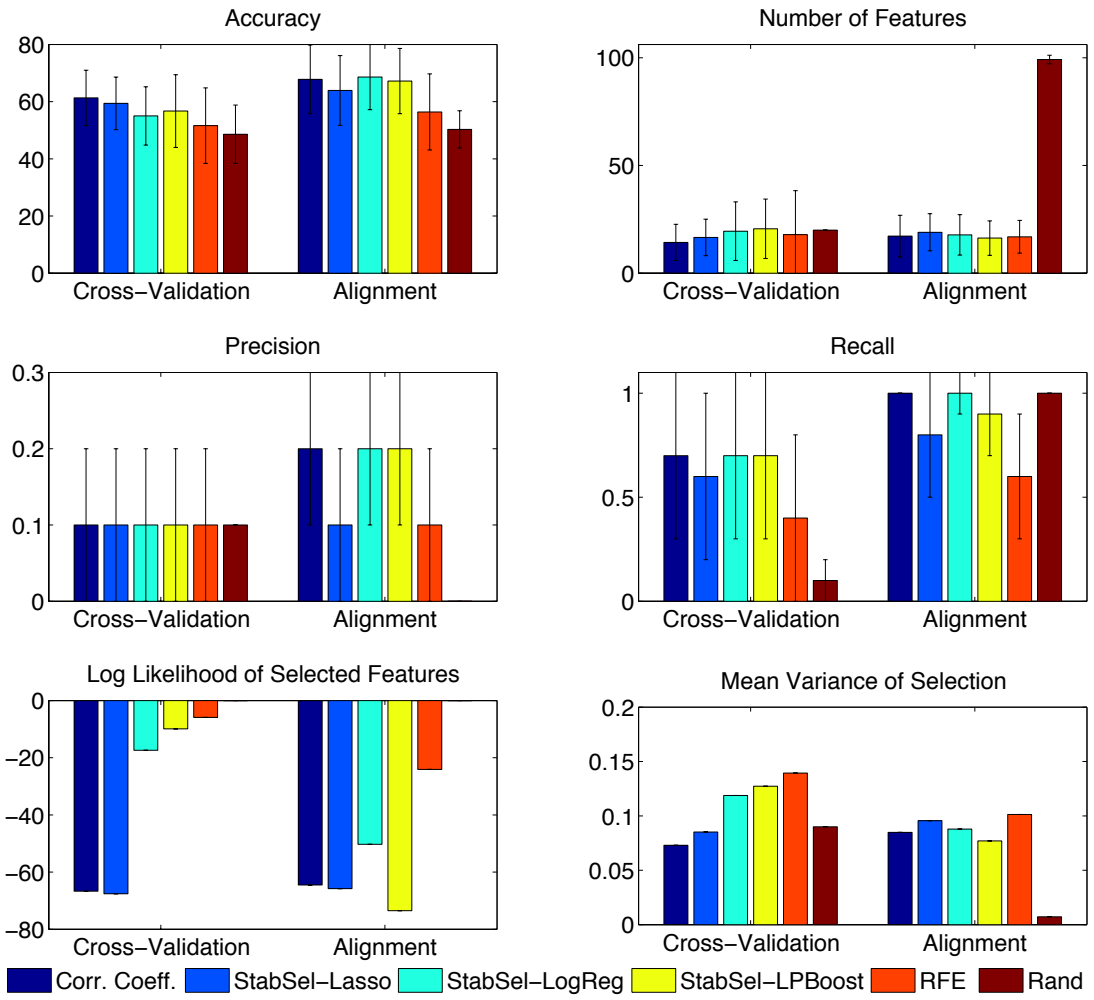

Figure 3.8: Results for the XOR dataset. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case.

3.3.2 Real Data

3.3.2.1 TB - Task 1

In the first TB Classification task(figure 3.9), most methods perform similarly in terms of generalization, with LPBoost stability selection relying on cross validation for model selection being the winner in terms of generalization accuracy. It is also worth noting that randomly selecting 10% of the variables gives a non-trivial result. This has largely to do with the fact that a large number of variables exhibit near equicorrelation with the target output, however using KTA for model selection on the randomly ranked features does outperform the random 10% benchmark in terms of generalization. LPBoost stability selection is also the winner in terms of sparsity among the models resulting from cross-validation relying on 90.4 features on average. The models relying on alignment, are in many cases competitive in terms of generalization performance, relying however on an order of magnitude fewer features, which is a key element in assisting interpretation. Correlation coefficients and linear RFE result in the sparsest models

on average.

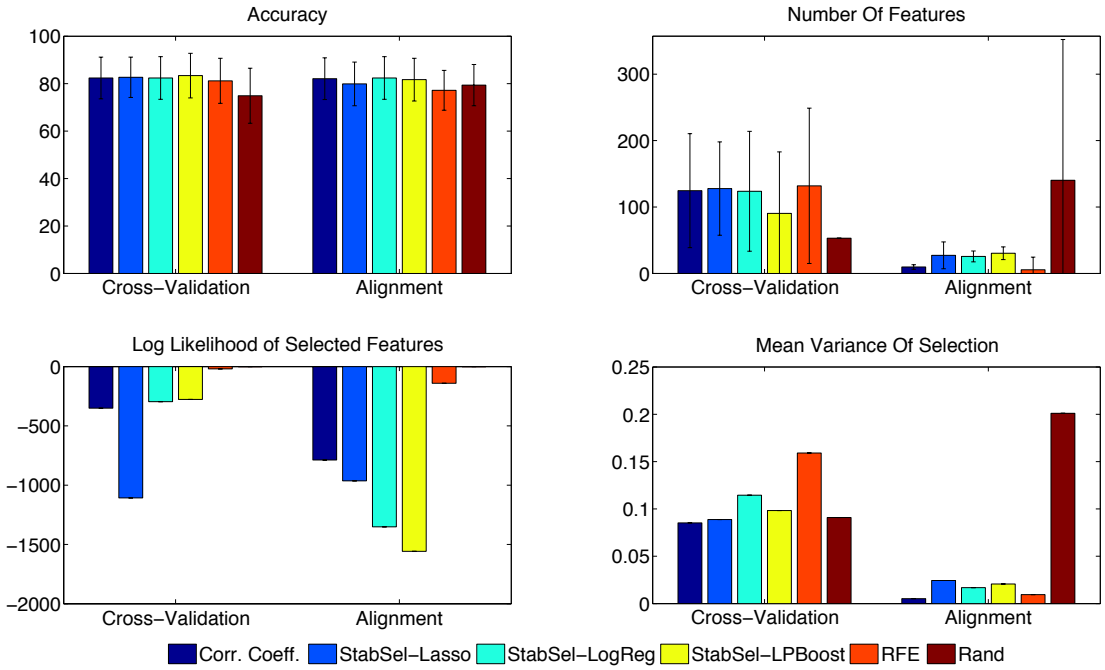

Figure 3.9: Results for the first TB task. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case.

3.3.2.2 TB - Task 2

Interestingly, on the second TB classification task (figure 3.10), the best performer in terms of accuracy is simple correlation coefficients with an average accuracy of 82.9, although most methods are indistinguishable. From the stability variants, LPBoost is yet again the best performer, with an accuracy of 81.8 when combined with alignment, being also the sparsest stability variant relying on 13.4 features for prediction on average. The sparsest overall method is yet again correlation coefficients, this time relying on an average of 6.5 features. KTA based models are universally sparser, and tend exhibit lower variance than their cross validation derived counterparts, with correlation coefficients being the overall sparsest.

3.3.2.3 TB - Task 3

On the third TB Task (figure 3.11), correlation coefficients remain the sparsest method, using on average 8.3 features. The winner in terms of generalization accuracy is Lasso-stability selection, combined with alignment with an average accuracy of 86%. Once more the other stability variants remain competitive in terms of accuracy when used in conjunction with alignment but rely on a substantially smaller number of features. KTA based models are sparser than their nested cross validation counterparts, with correlation coefficients being the overall sparsest. KTA-based LPBoost and l_1-regularized logistic regression stability selection models appear to be the most consistent. Interestingly correlation coefficients seems to be worse in terms of consistency when combined with KTA.

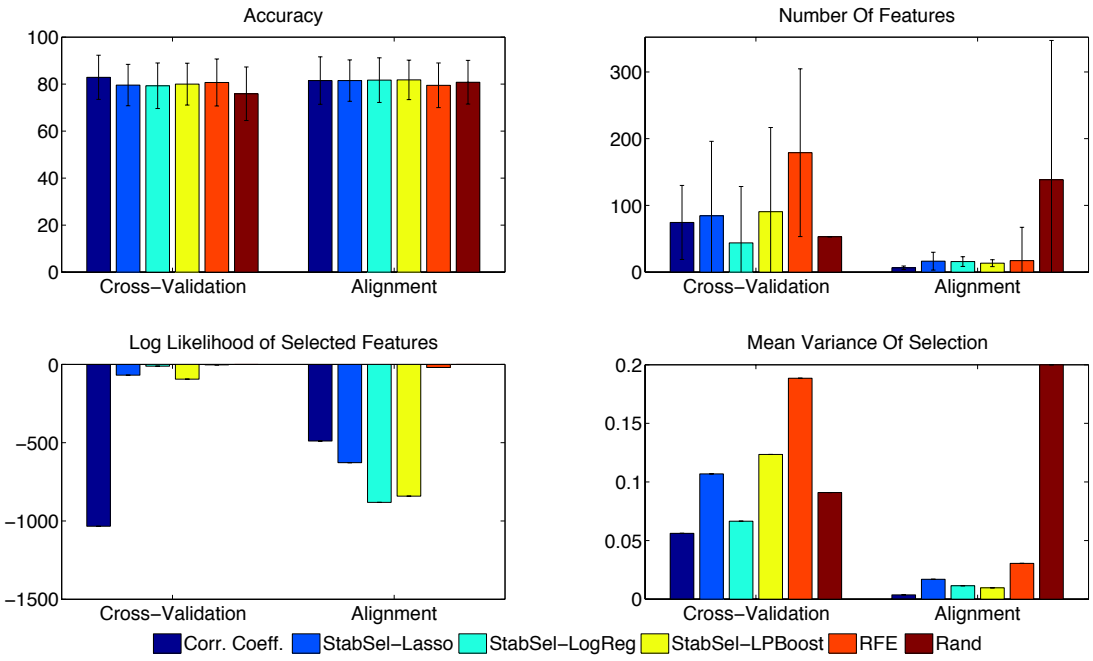

Figure 3.10: Results for the second TB task. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case.

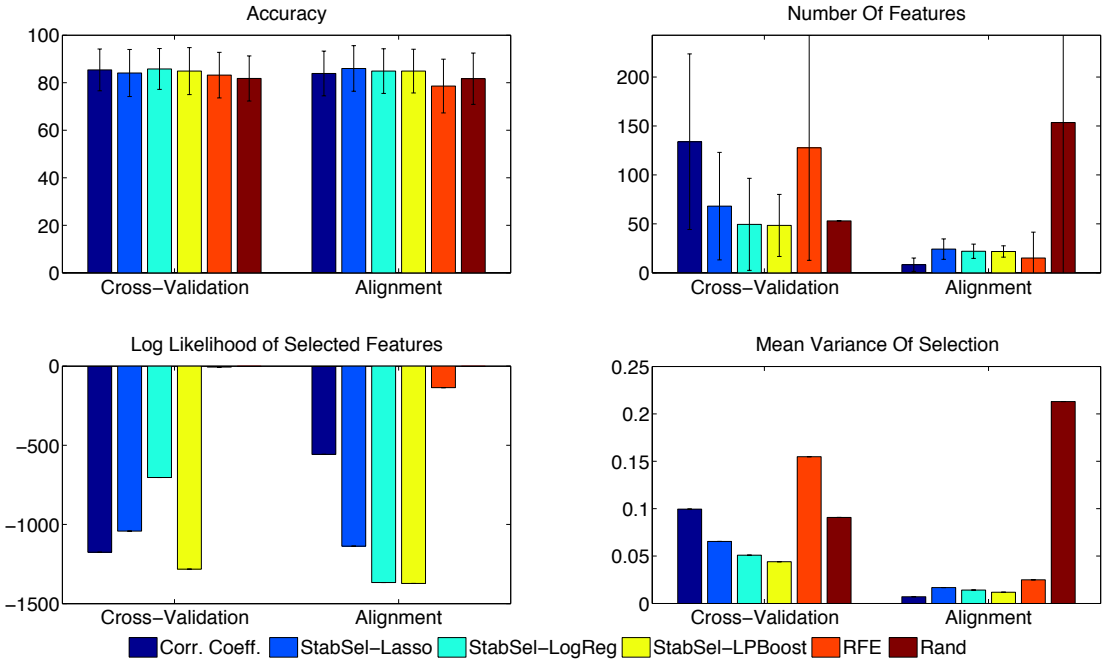

Figure 3.11: Accuracy results for the third TB task. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case.

3.3.2.4 TB - Task 4

The next two benchmarks are more interesting as they model a substantially more nuanced problem of predicting the presence of a latent TB infection in an otherwise seemingly healthy population. This explains the low classification accuracy all across the board for the fourth TB classification task (figure 3.12), as latent infection should not be expected to have a systematic impact on the differentiation of the blood-plasma proteomic profile. Accordingly most methods barely manage to achieve an accuracy of 60. In fact, only correlation coefficients combined with nested cross-validation is the only method that clearly outperforms using a random 10% subset of the features. Even though KTA-based models are sparser and more stable than their cross-validation counterparts, this property does not translate into substantial accuracy gains.

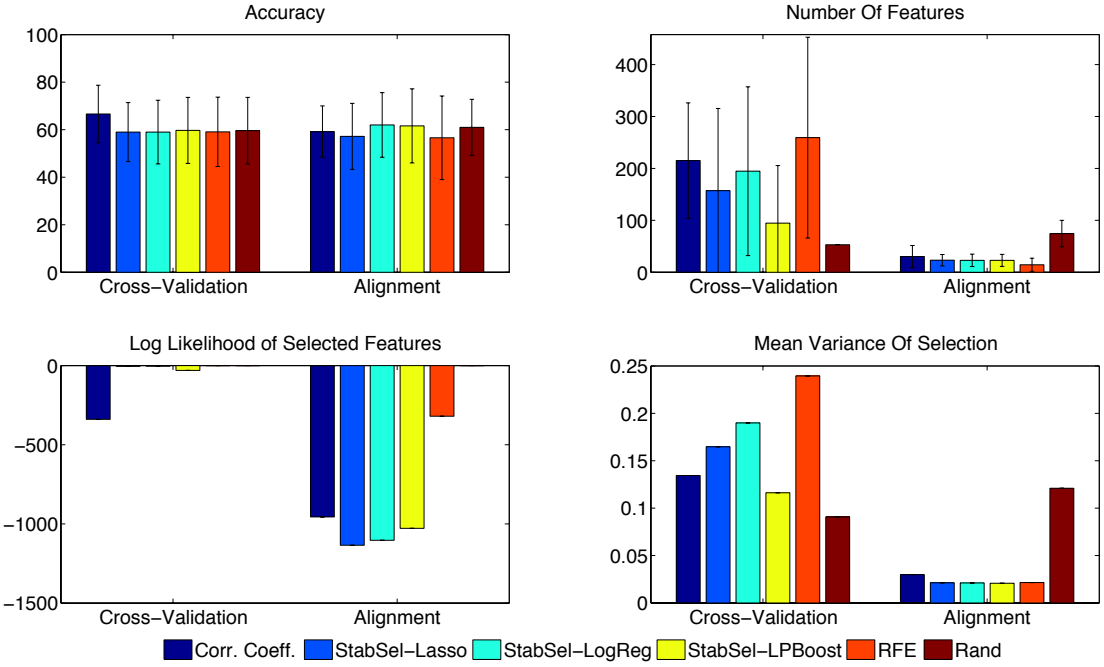

Figure 3.12: Results for the fourth TB task. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case.

3.3.2.5 TB - Micro-Array

The accuracy picture for the same classification task changes dramatically in the TB micro-array dataset (figure 3.13). Ostensibly, this stems from the better fit to the classification problem, resulting from the improved resolution of the underlying technology. This task illustrates the negative effects of aggressive sparsity, that were previously seen for some of the alignment models in the Linear Weston dataset. On average, none of the models selected through alignment use more than three features. This is reflected in the fact that the two densest models for the case of alignment, the one for logistic regression, and the linear-rfe, are the only ones that achieve a performance that touches 80.

This behavior is further reflected in the recall consistency results. In all other real world experiments, the likelihood for the alignment-based models was significantly smaller than their cross-validation

based counderparts,with the exception of the catastrophic failures for correlation coefficients and linear RFE in the linear Weston dataset. The micro-array dataset is another case where we observe this behavior, where alignment will greedily select a very small group of variables, that will be substantially different among different folds of the data. This observation while troublesome, empirically illustrates the importance of consistency, something we outlined in the desiderata. It can further be seen as validating the observation that the consistency of models through different folds of data, should be one of the guiding properties of feature selection.

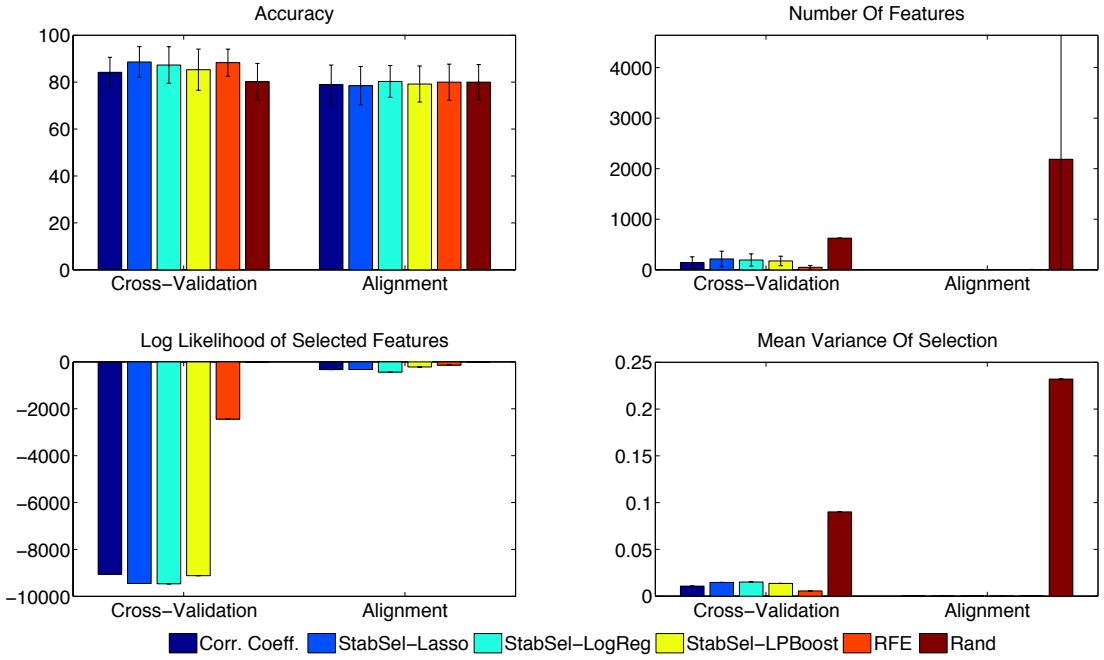

Figure 3.13: Results for the TB micro-array task. Rand selects a random 10% of the variables in the cross-validation setting, or the subset of a random ordering of variables which maximises alignment in the alignment case.

3.4 Discussion

In terms of feature selection, our experimental results indicate that stability selection is a potent framework for datasets arising in computational biology. In the performed experiments there was no clear winner in terms of generalization accuracy among the different stability selection variants. In terms of achieved sparsity the various stability selection approaches achieve somewhat similar results, with LP-Boost solutions appearing sparser in a number of settings. In terms of computational efficiency, stability selection methods outperform RFE, with the additional benefit that they are embarrassingly(this is used a lot as a term) parallel.

Using Kernel Target Alignment for model selection provides a significantly faster method and our experimental results on the synthetic benchmarks indicate substantially sparser models leading to improved precision in terms of the selected variables. Throughout our experiments, the only case where the achieved sparsity of the two approaches was comparable was the XOR dataset. It is noteworthy however

that in this case, the similar size to the KTA based model, leads to recovering both of the relevant variables. However, owing to the presence of a large number of irrelevant variables, this only led to modest gains in terms of generalization.

KTA based models were also more consistent in terms of variable selection for most of the experimental comparisons. The two datasets where the KTA derived models exhibited instability in terms of recovery are the same two datasets where the generalization accuracy for the alignment-derived models noticeably deteriorated (figures 3.6 and 3.13). This observation lends further credence to the importance of consistency for variable selection in our proposed framework.

Correlation coefficients, and linear RFE are two widely adopted feature selection methods which were used as baseline methods for the experimental comparisons. Both methods were competitive throughout all synthetic dataset benchmarks with the already noted exception of the linear Weston dataset, where both methods produced models that were too sparse when combined with KTA. Linear RFE is also the only method to successfully identify part of the structure of the non-linear Weston dataset, which explains why it was the only method to achieve an accuracy surpassing 70 when combined with alignment. Among all the methods tested, RFE appears to be the most sensitive to small changes in the dataset, something which is reflected by its being constantly the least consistent method in terms of the consistency of the recovered solutions.

Throughout most of the experiments, using KTA results in very sparse models. If a small number of variables is capable of explaining the majority of the observations, than the KTA based models have distinct advantages over using nested cross validation, with similar accuracy. However, when this condition does not hold, the generalization accuracy can diminish substantially. Interestingly in the two notable cases of the linear Weston dataset, and the TB micro-array dataset, this is accompanied with diminished consistency for the KTA based models, in contrast to the experiments with nearly all other datasets. This fact suggests that the consistency of the selected models to some degree can act as a proxy for their generalization ability.

This chapter attempts to establish the ability of the three step approach to generalise to real world and potentially non-linear datasets. The methods we examined are not designed to cope with non-linear datasets, something that the non-linear synthetic benchmarks strongly illustrated. The same experiments however exhibited somewhat improved behavior when using the alignment of a nonlinear kernel as the model selection criterion. This observation further underlines the possibility of directly optimising the alignment of a non-linear kernel function, something which is the central theme of the next chapter.

Chapter 4

Randomised Feature Selection

Experiments with the linear feature selection methods in the previous chapter illustrated their inability to effectively identify nonlinear combinations of features. Failing to identify such interrelation has important ramifications in various practical settings, where a single variable may exhibit weak influnence on a classification rule, but its effect when combined in groups of two or more independent variables becomes significantly more pronounced. An example of this in the context of genome-wide association studies is [KTS+09], where it is shown that three-variable interactions can identify switching mechanisms of two genes' expressions under the influence of a third gene. Some results in the synthetic datasets indicated that using the alignment of a combination of features in a RKHS, can yield some information and potentially identify non-linear interactions. This chapter examines methods that try to optimize directly the alignment of a subset of variables.

Kernel methods excel in modelling non-linear relations, and consequently a number of kernel-based feature selection algorithms have been proposed. Early propositions, such as RFE [GWBV02] can be computationally prohibitive. This is also true of more recent approaches for non-linear feature selection employing greedy optimisation of Centred Kernel Target Alignment(KTA). Although such methods exhibit strong results in terms of generalisation accuracy and sparsity, their application to high-dimensional datasets can become computationally prohibitive. Meanwhile, attempts to learn a convex combination of low-rank kernels may fail to encapsulate nonlinearities in the underlying relation[Cortes11]. Recent approaches using explicit kernel approximations can capture such non-linear relations[Bach, 2008], but increase the storage and computational requirements. The successful use of a kernel-based feature selection methods is a matter of balance.

This chapter proposes a randomised feature selection algorithm, with attractive scaling properties. It examines the use of centred KTA as a measure of dependence on RKHS, and reiterates the properties of the deterministic, greedy approaches in [SSG+12]. It then proceeds to show how randomization and bootstrapping can be combined with KTA to great effect by introducing a randomized algorithm that takes variance information into consideration.

An outline analysis suggests the possibility of deriving rigorous guarantees on the performance of the algorithm. The experimental results on real and artificial data, show that the method successfully identifies informative combinations of features, often outperforming its deterministic counterparts.

4.1 Related Work

Previous chapters introduced the Hilbert Schmidt Independence Criterion(HSIC). As a cross-covariance operator, HSIC functions analogously to covariance, but in reproducing kernel hilbert spaces. In statistics, covariance functions as a measure of statistical dependence, quantifying the extent to which two random variables $\mathbf{x}, \mathbf{y}$ vary together. In the case that $\mathbf{x}$ and $\mathbf{y}$ are independent, their covariance vanishes.

$$cov(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\}\{\mathbf{y} - \mathbb{E}[\mathbf{y}]\}] = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbf{xy}] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}]$$

And so $\mathbf{x}, \mathbf{y}$ are independent if

$$\mathbb{E}[\mathbf{xy}] = \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}] \text{ or alternatively } cov(\mathbf{xy}) = 0$$

The use of Hilbert-Schmidt norms as a measure of statistical dependence was introduced in [GBSS05]. In terms of kernel functions, the Hilbert Schmidt Independence Criterion can be expressed as

$$\begin{aligned} HSIC(p_{\mathbf{xy}}, \mathcal{F}, \mathcal{G}) = \mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}[k(\mathbf{x}, \mathbf{x}')l(\mathbf{y}, \mathbf{y}')] \\ +\mathbb{E}_{\mathbf{x},\mathbf{x}'}[k(\mathbf{x}, \mathbf{x}')]\mathbb{E}_{\mathbf{y},\mathbf{y}'}[l(\mathbf{y}, \mathbf{y}')] \\ -2\mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbb{E}_{\mathbf{x}'}[k(\mathbf{x}, \mathbf{x}')]\mathbb{E}_{\mathbf{y}'}[l(\mathbf{y}, \mathbf{y}')] \end{aligned}$$

where, $\mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}$ denotes expectation over independent pairs $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$ and $k(\mathbf{x}, \mathbf{x}')$, $l(\mathbf{y}, \mathbf{y}')$ are kernel functions on $\mathbf{x}$ and $\mathbf{y}$ respectively. The empirical estimator for this quantity is the centred KTA, whose computation in terms of previously computed gram matrices was introduced in **definition 2.3.1**. Furthermore, **section 3.1**, introduced a key property for the concentration of the empirical estimate, which is reiterated here:

$$\|HSIC(p_{xy}, \mathcal{F}, \mathcal{G}) - HSIC(Z, \mathcal{F}, \mathcal{G})\| \leq \sqrt{\frac{\log(6/\delta)}{\alpha^2 m}} + \frac{C}{m}$$

The above suggests that the empirical estimate will converge to the population estimate at rate $1/\sqrt{m}$, a property that guarantees the detection of statistical dependence with high probability. This property is illustrated for varying degrees of dependence in figure 4.1. [SSG$^+$12] illustrates that greedy optimization of the HSIC is an effective methodology for feature selection, proposing two variants, FoHSIC and BaHSIC relying on forward and backward selection respectively.

FoHSIC (algorithm 4), starts with an empty set of variables and at each iteration the variable whose inclusion maximises the KTA to the output kernel is selected and added to the set. FoHSIC repeats this procedure until it has included the entire set of variables. BaHSIC (algorithm 5), in essence is the functional converse of FoHSIC. Starting with the full set of features, at each iteration the algorithm removes the feature whose exclusion maximises, the KTA to the output kernel, repeating the process until all variables have been removed from consideration. FoHSIC is more efficient in terms of computation. This however, comes at a price, as unlike BaHSIC features are assessed individually and without the context of all the other present features. When considering inter-relations between variables this can be significantly detrimental, something that will be shown in the experimental results.

T

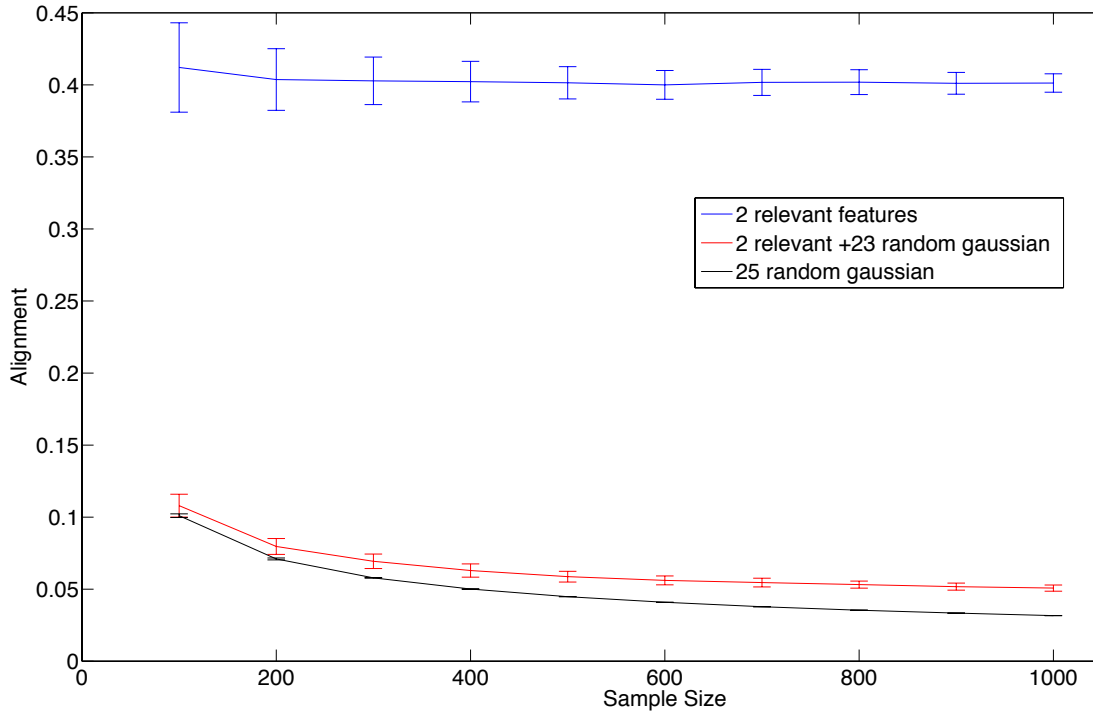

Figure 4.1: Mean alignment as a function of sample size. For each sample size the mean alignment is computed on 500 bootstraps of the data. Black line corresponds to a a random 25-dimensional multivariate gaussian. Red line depicts the mean alignment of 2 relevant variables generated according to the [WMC$^+$00], and 23 gaussian probes. Green depicts the alignment for only the two relevant features. The influence of self-interaction terms (the diagonal terms $c_{x_{ii}} c_{y_{ii}}$) decreases for larger sample sizes, leading to the displayed drop here.

Algorithm 4 FohSic

Input: Full set of variables S, input kernel function $\kappa(x, x')$, output kernel K_Y and and kernel bandwidth parameters Ξ

Initialize: $S^\dagger = \{\}$

repeat

$\sigma \leftarrow \Xi$

$I \leftarrow arg\max_I \sum_{j \in I} HSIC(S^\dagger \cup \{j\}, K_Y, \sigma), I \subset S$

$S \leftarrow S \setminus I$

$S^\dagger \leftarrow (S\dagger, I)$

until S={}

Return: Sequence of included variables $S^\dagger$

Algorithm 5 BahSic

Input: Full set of variables S, input kernel function $\kappa(x, x')$ output kernel K_Y and kernel bandwidth parameters Ξ.

Initialize: $S^\dagger = \{\}$

repeat

$\sigma \leftarrow \Xi$

$I \leftarrow arg\max_I \sum_{j \in I} HSIC(S^\dagger \setminus \{j\}, K_Y, \sigma), I \subset S$

$S \leftarrow S \setminus I$

$S^\dagger \leftarrow (S\dagger, I)$

until S={}

Return: Sequence of dropped variables $S^\dagger$

4.2 A randomized algorithm for feature selection

In principle feature selection using HSIC in combination with greedy forward or backward selection is an effective strategy for the elucidation of nonlinear feature interactions. However, in practice greedy schemes for variable selection, and especially those relying on backward elimination, become computationally prohibitive for large numbers of features. The following section develops some key ideas, that can be combined to produce a weighting scheme for feature selection employing HSIC. Instead of relying on a greedy scheme, this proposal employs randomization to elicit information on the relevance of features. Combining the resulting fast weighting scheme with backward elimination leads to a fast algorithm for nonlinear feature selection.

4.2.1 Development of key ideas

The core of the proposed approach remains the use of centred kernel target alignment as an empirical estimator for HSIC. Substantial evidence in support of HSIC as a feature selection criterion are presented in [SSG+12], along with the two greedy optimization procedures of FoHSIC and BaHSIC.

The main novelty of our proposal is the use of randomization in order to produce an estimate of the importance of each variable. At each stage, the algorithm randomly splits the variables in two groups of equal size. This random splitting is repeated over a number of bootstrapped samples of the data, and the algorithm computes the alignment for each resulting random half of the variables over these bootstraps.

The final constituent of the method is bootstrapping. Certainly, the use of subsampling is widespread in statistical modelling where it is commonly used for the estimation of confidence intervals and statistical testing. Its use in this proposal however, is motivated by its prominence in the stability selection framework [MB10]. Subsampling lies at the core of stability selection where it is combined with sparse selection algorithms in order to detect dependence relations that consistently occur in a large fraction of the resulting selection sets. What's more, figures 4.2 strongly suggests that for samples corresponding to higher signal-to-noise ratios, the variance of the resulting alignment seems to be much higher, something which is demonstrated visually by their substantially increased spread in both figures.

This work, proposes the synthesis of the aforementioned approaches through a randomised fea-

ture selection algorithm based on estimating the statistical dependence between random subspaces of bootstraps of the dataset in RKHS. Having completed introduction of the individual constituents the following section will examine the properties of the resulting algorithm.

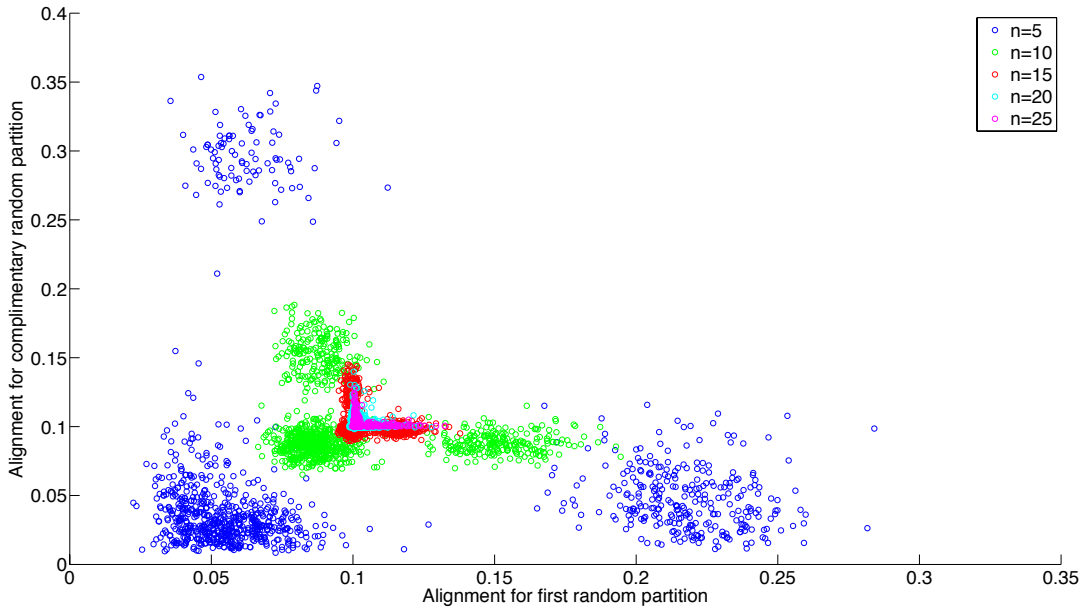

Figure 4.2: Scatter plot of the resulting alignment over random splits of variables for an XOR dataset with the additional inclussion of a varying number of irrelevant variables (blue n=5, green n=10, red n= 15, light blue n= 20, magenta n = 25, bootstrap size = 100 samples). Over the different number of dimensions, a common pattern emerges: Samples concentrated over the lower left corner correspond to estimates in which each random split of the variables contains a single relevant feature, resulting in lower alignment for both splits. The bottom-right and top-left corners contain cases where one split contains both of the relevant variables, resulting in a visible hike in alignment for that split. It is also particularly instructive to notice that for samples corresponding to higher signal-to-noise ratios, the variance of the resulting alignment seems to be much higher, further justifying subsampling.

4.2.2 A randomized algorithm for feature selection

The approach we will take will be based on observations that link kernel target alignment with the degree to which an input space contains a linear projection that correlates with the target, such as Propositions *3*.1 and *3*.1.1. These propositions suggest that measuring kernel target alignment can detect useful representations. For non-linear functions the difficulty is to identify which combination of features creates a useful representation. Section 4.3 illustrates how this problems can be addressed through randomization and sampling. Here, we solidify the notion sampling subsets S of features and assessing whether on average the presence of a particular feature i contributes to an increase c_i in the average centred kernel target alignment. In this way we derive an empirical estimate of a quantity we will term the contribution.

Definition 4.2.1. *The* contribution c_i *of feature* i *is defined as*

$$\begin{aligned} c_i &= \mathbb{E}_{S\sim\mathcal{S}_i}[\mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}[\kappa_S(\mathbf{x},\mathbf{x}')\kappa'(\mathbf{y},\mathbf{y}')] + \mathbb{E}_{\mathbf{x},\mathbf{x}'}[\kappa_S(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y},\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')] \\ &\quad -2\mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbb{E}_{\mathbf{x}'}[\kappa_S(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')] \\ &\quad -\mathbb{E}_{S'\sim\mathcal{S}_{\setminus i}}[\mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}[\kappa_{S'}(\mathbf{x},\mathbf{x}')\kappa'(\mathbf{y},\mathbf{y}')] + \mathbb{E}_{\mathbf{x},\mathbf{x}'}[\kappa_{S'}(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y},\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')] \\ &\quad -2\mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbb{E}_{\mathbf{x}'}[\kappa_{S'}(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')], \end{aligned}$$

where κ_S *denotes the (non-linear) kernel using features in the set* S *(in our case this will be a Gaussian kernel with equal width),* κ' *is a kernel defined on the target outputs,* $\mathcal{S}_i$ *the uniform distribution over sets of features of size* $\lfloor n/2\rfloor + 1$ *that include the feature* i*,* $\mathcal{S}_{\setminus i}$ *the uniform distribution over sets of features of size* $\lfloor n/2\rfloor$ *that do not contain the feature* i*, and* n *is the number of features. We further introduce the notation*

$$\begin{aligned} A(S) &= \mathbb{E}_{S\sim\mathcal{S}_i}[\mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}[\kappa_S(\mathbf{x},\mathbf{x}')\kappa'(\mathbf{y},\mathbf{y}')] + \mathbb{E}_{\mathbf{x},\mathbf{x}'}[\kappa_S(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y},\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')] \\ &\quad -2\mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbb{E}_{\mathbf{x}'}[\kappa_S(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')] \end{aligned}$$

for the centred kernel target alignment of the kernel defined on a set S *of variables. Hence, we can write*

$$c_i = \mathbb{E}_{S\sim\mathcal{S}_i}\left[A(S)\right] - \mathbb{E}_{S'\sim\mathcal{S}_{\setminus i}}\left[A(S')\right] = A_i - A_{\setminus i}, \tag{4.1}$$

where $A_i = \mathbb{E}_{S\sim\mathcal{S}_i}\left[A(S)\right]$ *and similarly for* $A_{\setminus i}$*.*

Note that the two distributions over features $\mathcal{S}_i$ and $\mathcal{S}_{\setminus i}$ are matched in the sense that for each S with non-zero probability in $\mathcal{S}_{\setminus i}$, $S\cup\{i\}$ has equal probability in $\mathcal{S}_i$. This approach is a straightforward extension of the idea of BaHsic [5].

We will show that for variables that are independent of the target this contribution will be negative. On the other hand, provided there are combinations of variables including the given variable that can generate significant correlations then the contribution of the variable will be positive. We now consider the problem of specifying whether a random variable is irrelevant. Ideally we would define a random variable to be irrelevant if it was independent of the target outputs Y, given the values of all other relevant variables. However in order to make the analysis tractable we make a stronger assumption in terms of irrelevance.

Definition 4.2.2. *We will define an* irrelevant *feature to be one whose value is statistically independent of the label and of the other features.*

We have previously noted that in terms of kernel functions, the Hilbert Schmidt Independence Criterion can be expressed as:

$$\mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}[\kappa(\mathbf{x},\mathbf{x}')\kappa'(\mathbf{y},\mathbf{y}')] + \mathbb{E}_{\mathbf{x},\mathbf{x}'}[\kappa(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y},\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')] - 2\mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbb{E}_{\mathbf{x}'}[\kappa(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')].$$

We would like an assurance that irrelevant features do not increase centred kernel target alignment. This is guaranteed for the Gaussian kernel by the following result.

Proposition 4.2.3. *Let P be a probability distribution on the product space $\mathcal{X} \times \mathbb{R}$, where $\mathcal{X}$ has a projection ϕ_S into a Hilbert space $\mathcal{F}$ defined by the Gaussian kernel κ_S on a set of features S. In addition, define the kernel κ' on the targets y. Suppose a feature $i \notin S$ is irrelevant. We have that*

$$\begin{aligned}\mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}[\kappa_{S\cup\{i\}}(\mathbf{x},\mathbf{x}')\kappa'(\mathbf{y},\mathbf{y}')] + \mathbb{E}_{\mathbf{x},\mathbf{x}'}[\kappa_{S\cup\{i\}}(\mathbf{x},\mathbf{x}')\mathbb{E}_{\mathbf{y},\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')] - 2\mathbb{E}_{\mathbf{x},\mathbf{y}}[\kappa_{S\cup\{i\}}(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')] \\ \leq \mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}[\kappa_S(\mathbf{x},\mathbf{x}')\kappa'(\mathbf{y},\mathbf{y}')] + \mathbb{E}_{\mathbf{x},\mathbf{x}'}[\kappa_S(\mathbf{x},\mathbf{x}')\mathbb{E}_{\mathbf{y},\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')] - 2\mathbb{E}_{\mathbf{x},\mathbf{y}}[\kappa_S(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')]\end{aligned}$$

Proof. Since the feature is independent of the target and the other features, functions of these features are also independent. Hence,

$$\begin{aligned}&\mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}[\kappa_{S\cup\{i\}}(\mathbf{x},\mathbf{x}')\kappa'(\mathbf{y},\mathbf{y}')] + \mathbb{E}_{\mathbf{x},\mathbf{x}'}[\kappa_{S\cup\{i\}}(\mathbf{x},\mathbf{x}')\mathbb{E}_{\mathbf{y},\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')] - 2\mathbb{E}_{\mathbf{x},\mathbf{y}}[\kappa_{S\cup\{i\}}(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')] \\ &= \alpha\left(\mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}[\kappa_S(\mathbf{x},\mathbf{x}')\kappa'(\mathbf{y},\mathbf{y}')] + \mathbb{E}_{\mathbf{x},\mathbf{x}'}[\kappa_S(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y},\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')] - 2\mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbb{E}_{\mathbf{x}'}[\kappa_S(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')]\right) \\ &\leq \mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}[\kappa_S(\mathbf{x},\mathbf{x}')\kappa'(\mathbf{y},\mathbf{y}')] + \mathbb{E}_{\mathbf{x},\mathbf{x}'}[\kappa_S(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y},\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')] - 2\mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbb{E}_{\mathbf{x}'}[\kappa_S(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y}'}[\kappa'(\mathbf{y},\mathbf{y}')]\end{aligned}$$

for $\alpha = \mathbb{E}_{(\mathbf{x},y)\sim P,(\mathbf{x}',y')\sim P}[\exp(-\gamma(x_i - x_i')^2)] \leq 1$. □

In fact the quantity α is typically less than 1 so that adding irrelevant features decreases the alignment. Our approach will be to progressively remove sets of features that are deemed to be irrelevant, hence increasing the alignment together with the signal to noise ratio for the relevant features. Figure 4.3 shows how progressively removing features from a learning problem whose output is the XOR function of the first two features both increases the alignment contributions and helps to highlight the two relevant features.

We now introduce our definition of a relevant feature.

Definition 4.2.4. *A feature i will be termed* η-influential *when its contribution $c_i \geq \eta > 0$.*

So far have only considered expected alignment. In practice we must estimate this expectation from a finite sample. This part of the analysis is an application of U-statistics that ensures that with high probability for a sufficiently large sample from $\mathcal{S}_i$ and $\mathcal{S}_{\setminus i}$ and of samples from P (whose sizes depend on η, probability δ, the number k of η-influential variables and the number T of iterations) an empirical estimate of the contribution of an η-influential variable will with probability at least $1-\delta$ be greater than 0 for all of the fixed number T of iterations of the algorithm.

We begin by bounding the effect of taking an empirical estimate for the expectations in definition 4.1. Consider n_s randomly chosen sets of $\lfloor n/2 \rfloor + 1$ variables that include feature i, we denote these sets $S_1, \ldots, S_{n_s}$. We denote the empirical mean alignment as:

$$\hat{A}_i = \frac{1}{n_s}\sum_{j=1}^{n_s} A(S_j)$$

Note that these estimates are averages of the true alignments which we will not be able to estimate exactly. We will consider their estimation later. Similarly, consider a sample of size n_s of $n/2$ features that exclude feature i. We denote these sets $S_1', \ldots, S_{n_s}'$ and the empirical estimate of $A_{\setminus i}$:

$$\hat{A}_{\setminus i} = \frac{1}{n_s}\sum_{j=1}^{n_s} A(S_j')$$

T

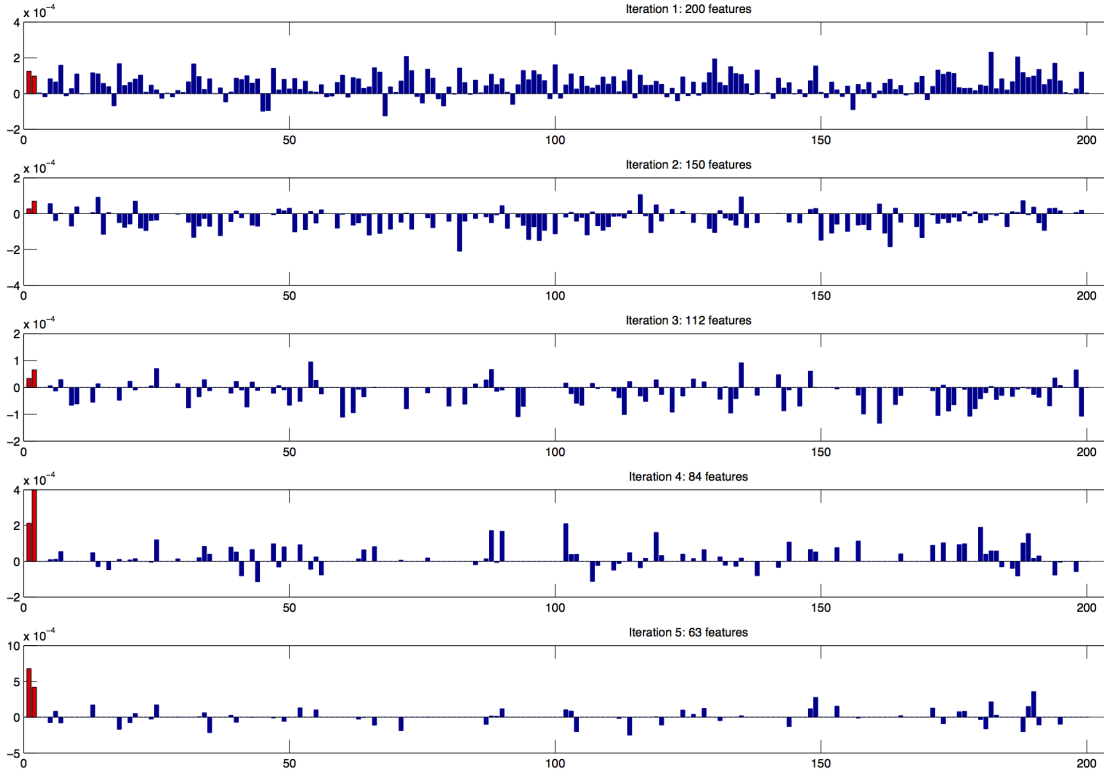

Figure 4.3: 200-dimensional XOR classification problem. The expected contribution of the two relevant features is in red. It can be seen that as more of the noise features are removed in latter iterations of the method, the expected contribution of the two relevant variables rises substantially, in contrast to the contribution of the other features.

Finally, denote by $\hat{c}_i$ the empirical estimate of the contribution of feature i computed as

$$\hat{c}_i = \hat{A}_i - \hat{A}_{\setminus i}.$$

Proposition 4.2.5. *Consider the empirical estimate $\hat{c}_i$ of the contribution of a feature i. The probability of a deviation larger than $\frac{\eta}{6}$ from the true contribution of feature i can be bounded as:*

$$P(|c_i - \hat{c}_i| \geq \frac{\eta}{6}) \leq 4 \exp\left(-\frac{1}{72} n_s \eta^2\right) \tag{4.2}$$

Proof. Using Hoeffding's inequality, we have:

$$P(|\hat{A}_i - A_i| \geq t) \leq 2 \exp\left(-\frac{2n_s^2 t^2}{\sum_{j=1}^{n_s}(1-0)}\right) \leq 2 \exp\left(-2n_s t^2\right)$$

Setting $t = \frac{\eta}{12}$

$$P(|\hat{A}_i - A_i| \geq \frac{\eta}{12}) \leq 2 \exp\left(-\frac{1}{72} n_s \eta^2\right) \tag{4.3}$$

A further application of Hoeffding's inequality on $A_{\setminus i}$ gives:

$$P(|\hat{A}_{\setminus i} - A_{\setminus i}| \geq \frac{\eta}{12}) \leq 2 \exp\left(-\frac{1}{72} n_s \eta^2\right) \tag{4.4}$$

Since $c_i = A_i - A_{\setminus i}$,

$$|c_i - \hat{c}_i| \leq |\hat{A}_i - A_i| + |\hat{A}_{\setminus i} - A_{\setminus i}|,$$

and hence we can use equations (4.3) and (4.4) to bound the deviation between c_i and $\hat{c}_i$ as:

$$P(|c_i - \hat{c}_i| \geq \frac{\eta}{6}) \leq 4 \exp\left(-\frac{1}{72} n_s \eta^2\right) \tag{4.5}$$

□

Corollary 4.2.6. *Provided that we use a sample size $n_s = \frac{72}{\eta^2} \ln\left(\frac{8nT}{\delta}\right)$, then with probability $1 - \frac{\delta}{2}$ there is no deviation larger than $\frac{\eta}{3}$ for the entire set of n variables over all T time periods.*

Proof. Equation (4.5) calculates the probability of an a deviation larger than $\frac{\eta}{6}$ for a single variable i. An upper bound for the number of deviations of magnitude larger than $\frac{\eta}{6}$ for the entire set of n variables can be computed by taking the the union bound over the n different variables yielding:

$$\begin{aligned} P\left(\exists i : |c_i - \hat{c}_i| \geq \frac{\eta}{6}\right) &\leq \sum_{i=1}^{i=n} P\left(|c_i - \hat{c}_i| \geq \frac{\eta}{6}\right) \\ &\leq 4 \sum_i \exp\left(-\frac{1}{72} n_s \eta^2\right) \leq 4n \exp\left(-\frac{1}{72} n_s \eta^2\right) \end{aligned}$$

We would like this to hold with high confidence for all T iterations. Setting the left hand side of the equation to $\frac{\delta}{2T}$, in equation (4.6), we obtain

$$\frac{\delta}{2T} = 4n \exp\left(-\frac{1}{72} n_s \eta^2\right) \tag{4.6}$$

Solving for n_s in equation (4.6) we obtain an estimate for n_s

$$n_s = \frac{72}{\eta^2} \ln\left(\frac{8nT}{\delta}\right), \text{ as required.} \tag{4.7}$$

□

We now consider estimating the alignment $A(S)$ for a set S of variables from a sample $B = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{|B|}, y_{|B|})\}$. We will denote the empirical estimate of this quantity as $\hat{A}(S, B)$, where

$$\hat{A}(S, B) = \frac{2}{n(n-1)} \sum_{1 \leq i,j \leq |B| : i \neq j} \kappa(\mathbf{x}_i, \mathbf{x}_j) y_i y_j. \tag{4.8}$$

Proposition 4.2.7. *Consider a fixed set S of variables, and the gaussian kernel $\kappa_S(\mathbf{x}, \mathbf{x}')$ defined on this set. Let B be an iid sample drawn from the distribution P. It follows that*

$$P(|\hat{A}(S, B) - A(S)| \geq t) \leq 2 \exp\left(\frac{2|B|t^2}{4}\right). \tag{4.9}$$

Proof. We will apply Hoeffding's concentration bound for U-Statistics from Theorem A.1.1. We define X_i to be $(\mathbf{x}_i, y_i)$ and

$$g(X_i, X_{i+1}) = g((\mathbf{x}_i, y_i), (\mathbf{x}_{i+1}, y_{i+1})) = y_i y_{i+1} \kappa_S(\mathbf{x}_i, \mathbf{x}_{i+1}). \tag{4.10}$$

We must verify U is equal to the empirical alignment, that is

$$U = \frac{2}{n(n-1)} \sum_{i,j \leq |B|: j \neq i} \kappa_S(\mathbf{x}_i, \mathbf{x}_j) y_i y_j.$$

It will then follow that $\mathbb{E}(U) = A(S)$ and the result will follow from Theorem A.1.1. Since U is an average of evaluations of expressions $g(\cdot, \cdot)$, the result follows from the observation that all the expressions $g((\mathbf{x}_i, y_i), (\mathbf{x}_j, y_j))$, for $i \neq j$, occur equally often by the symmetry of the permutation group. □

Corollary 4.2.8. *Fix $\delta > 0$, with probability $1 - \frac{\delta}{2}$, provided that we use samples $m_a \geq \frac{72}{\eta^2} \frac{\ln 4 n_s n T}{\delta}$ for estimating $A(S)$ for all $2 n_s n T$ occasions, all estimates will be within $\frac{\eta}{6}$.*

Proof. We apply Proposition 4.2.7 $n_s n T$ times with $t = \frac{\eta}{6}$. Substituting m_a for $|B|$ in the right hand side of equation (4.9), we obtain $\frac{\delta}{2 n_s n T}$, and the result follows. □

Theorem 4.2.9. *Consider estimating the alignment by sampling n_s sets of features $S_1^i, \ldots, S_{n_s}^i$ according to $\mathcal{S}_i$. For each set S_j^i of features , an iid sample B_j^i of size m_a from P, for each feature $i = 1, \ldots, n$, where $n_s = \frac{72}{\eta^2} \ln\left(\frac{8nT}{\delta}\right)$ and $m_a \geq \frac{72}{\eta^2} \frac{\ln 4 n_s n T}{\delta}$. Similarly sample n_s sets of features from $\mathcal{S}_{\setminus i}$. Denote the estimated alignments by*

$$\hat{\hat{A}}_i = \frac{1}{n_s} \sum_{j=1}^{n_s} A(S_j^i, B_j^i), \quad \hat{\hat{A}}_{\setminus i} = \tfrac{1}{n_s} \textstyle\sum_{j=1}^{n_s} A(S_j^{\setminus i}, B_j^{\setminus i}),$$

with $\hat{\hat{c}}_i = \hat{\hat{A}}_i - \hat{\hat{A}}_{\setminus i}$, for $i = 1, \ldots, n$ Then, with probability $1 - \delta$, $|\hat{\hat{c}}_i - c_i| \leq \frac{\eta}{2}$, $\forall i = 1, \ldots, n$, for all iterations of the algorithm.

Proof. For all i, and all iterations, we can bound $|\hat{\hat{c}}_i - c_i|$ as follows:

$$\begin{aligned} |\hat{\hat{c}}_i - c_i| &\leq |\hat{\hat{c}}_i - \hat{c}_i| + |\hat{c}_i - c_i| \\ &\leq |\hat{\hat{c}}_i - \hat{c}_i| + \frac{\eta}{6} \end{aligned}$$

with probability at least $1 - \frac{\delta}{2}$, by applying corollary 4.2.8. Furthermore, with probability at least $1 - \frac{\delta}{2}$ for all iterations:

$$\begin{aligned} |\hat{\hat{c}}_i - \hat{c}_i| &\leq |\hat{\hat{A}}_i - \hat{A}_i| + |\hat{\hat{A}}_{\setminus i} - \hat{A}_{\setminus i}| \\ &\leq \frac{1}{n_s} \sum_{j=1}^{j=n_s} |A(S_j^i, B_j^i) - A(S_j^i)| + \sum_{j=1}^{j=n_s} |A(S_j^{\setminus i}, B_j^{\setminus i}) - A(S_j^{\setminus i})| \\ &\leq \frac{1}{n_s} \sum_{j=1}^{j=n_s} \frac{\eta}{6} + \frac{1}{n_s} \sum_{j=1}^{j=n_s} \frac{\eta}{6} \\ &\leq \frac{2\eta}{6}. \end{aligned}$$

Hence with probability at least $1 - \delta$, $|\hat{\hat{c}}_i - c_i| \leq \frac{\eta}{2}$, for all $i = 1, \ldots, n$ for all iterations. □

Our final ingredient is a method of removing irrelevant features that we will term culling. It is inspired by the following theorem which states that we can reliably remove irrelevant vartiables. After an iteration of the algorithm the contributions of all of the features are estimated using the required sample size and those features with contributions $\hat{\hat{c}}_i < \frac{\eta}{2}$ are removed.

Theorem 4.2.10. *Fix $\eta > 0$. Suppose that there are k η-influential variables and all other variables are irrelevant. Fix $\delta > 0$. After 1 iteration of the algorithm, given sufficiently many samples $n_s = \frac{72}{\eta^2} \ln\left(\frac{8n}{\delta}\right)$ and $m_a \geq \frac{72}{\eta^2} \frac{\ln 4n_s n}{\delta}$ as described in Theorem 4.2.9, removing all features with estimate contribution $\hat{\hat{c}}_i < \eta/2$, will with probability at least $1 - \delta$ only preserve all k relevant features, and remove all irrelevant features.*

Proof. For all irrelevant variables i, $\hat{\hat{c}}_i \leq \frac{\eta}{2}$ with probability at least $1 - \delta$. Conversely, for all relevant variables will be within $\hat{\hat{c}}_j \geq \frac{\eta}{2}$, with probability at least $1 - \delta$. Therefore, provided sufficiently many samples as described in theorem 4.2.9, removing all variables with contribution $\hat{\hat{c}}_i < \eta/2$ will only remove irrelevant variables, and preserve all relevant variables. □

This section has provides a theoretical study of a randomised feature selection approach employing KTA, however it does not provide a theoretical comparison with its deterministic counterparts, which is an open topic for future research. In addition, the sample size required to achieve the probabilistic guarantees of Theorem 4.2.10, are too large for most practical settings. For this reason in our experimentation, we proceed to iteratively cull a smaller percentage of the bottom-contributing features at the end of each iteration. For example, most of the experiments in this chapter were performed with culling 25% of the features after the end of each iteration.

4.2.3 Properties of the algorithm

We now define our algorithm for randomised selection (randSel). Given a $m \times n$ input matrix X and corresponding output matrix Y, randSel proceeds by estimating the individual contribution of features by estimating the alignment of a number of random subsamples that include $\frac{n}{2}$ and $\frac{n}{2} + 1$ randomly selected features. This leads to an estimate for the expected alignment contribution of including a feature. The algorithm is parametrized by the number of bootstraps n_s, a bootstrap size m_a and a percentage $z\%$ of features that are dropped after n_s bootstraps. The algorithm proceeds iteratively until only two features remain. Optionally the algorithm can be further parametrized by permanently including features which were ranked in the top percentile $a\%$ on at least a number t occasions. This option enhances the probability of detecting non-linear dependencies between variables, should they be present.

There are a number of benefits to this approach, aside from the tangible probabilistic guarantees. RandSel scales gracefully. Considering the computation of a kernel $k(x, x')$ for samples x, x' atomic, the number of kernel computations for a single iteration are $m_a^2 n_s$, which for a sensible choice of m_a and n_s can be substantially smaller than the $m^2 n$ complexity of the deterministic HSIC variants, which have a quadratic dependence on the sample size. For example setting $m_a = \sqrt{m}$ and $n_s = n$ an iteration would require mn kernel element computations.

4.2.4 Model Selection

The final ingredient to the methods we examined is an effective approach for model selection. FoHSIC and BaHSIC by design return an ordered list, where the variables are ranked by their sequence of inclusion/exclusion respectively. This fits with the cross validation framework outlined in section 2.5.1. RandSel is somewhat different in this respect from its deterministic siblings, as at each iteration it pro-

Algorithm 6 randSel

Input: input data X, labels Y, number of iterations n_s subsample size m_a, number of features n, drop percentile proportion z, top percentile proportion a, number of occasions t

repeat

for $i = 1$ **to** n_s **do**

$B \leftarrow$ random subsample from of size m_a from X

$S_i \leftarrow$ random subset of $\frac{n}{2} + 1$ variables, including the ith variable.

$S_{\setminus i} \leftarrow$ random subset of $\frac{n}{2}$ variables, excluding the ith variable.

$a_i \leftarrow \hat{A}(S_i, B)$

$a_i^{(+)} \leftarrow \hat{A}(S_{\setminus i}, B))$

end for

for $j = 1$ **to** n **do**

$\hat{\hat{c}}_j \leftarrow \hat{\hat{A}}(S_j, B) - \hat{\hat{A}}(S_{\setminus j}, B)$

end for

cull the $z\%$ bottom-contributing features

save the $a\%$ top-contributing features

if fixing features **then**

if j top-contributor for t consecutive times **then**

fix feature j

end if

end if

until no features left to fix, or only 2 features remain

Return Sequence of estimated contributions and Fixed Variables

duces a list of the expected contributions for the inclusion of each remaining variable. While the expected contribution does afford us with a ranking criterion that can readily be used in the framework we previously introduced, there's also a simpler option, which is a more natural fit for the procedure.

Instead of trying to produce a total ranking of the variables, we can look at the expected contributions at the end of each iteration of the algorithm, and only include the features which are expected to have a positive contribution. Further to integrating additional information made available by the algorithm, this process greatly reduces the computational burden, as the individual models resulting from this process is typically substantially smaller.

4.3 Experiments & Results

This section presents experiments comparing randSel to other nonlinear feature selection approaches, specifically, RFE, FoHSIC and BaHSIC. The experiments follow closely the experimental setup in section *3*.3, and are designed to capture the evaluation metrics that were outlined in section *2*.5. For the input data, the gaussian kernel is used throughout the experiments, while we use the simple linear kernel for the output data. The same range of gaussian kernel bandwidths was explored in all algorithms and

the resulting final classifiers employed a regularisation parameter of $c = 1$. Specific details relating to the different algorithms used are included in the following sections.

For the synthetic benchmarks we used randSel using 3000 bootstraps of size $m/4$ of the dataset, culling the bottom 25% of variables in terms of expected contribution after the end of each iteration. The same 25% proportion of features was dropped for the two other backward elimination methods, RFE and BaHSIC. We allowed FoHSIC to run for 150 iterations, adding a single variable at the end of each loop.

For the mass spectrometry tasks we used randSel using 5000 bootstraps of size $m/3$ of the dataset, culling the bottom 25% of variables in terms of expected contribution after the end of each iteration. We used similar parameters for the Micro-array dataset but with an increased number of bootstraps of 10000 in order to account for the substantially higher dimensionality of the data. Again, no variables were fixed and the algorithm iterated until only two variables remained.

4.3.1 Synthetic Data

4.3.1.1 Fake Class

In the fake class benchmark (figure 4.4), the generalization accuracy achieved by the different methods is not significantly different from choosing features at random, and the number of identified features changes substantially between folds, for most methods. As expected they are all unable to establish a meaningful dependence relation between the selected features and the output, explaining the poor recall and precision. The estimated set of relevant variables changes substantially between folds for all methods, which the log likelihood of the recovered sets of variables strongly reflects.

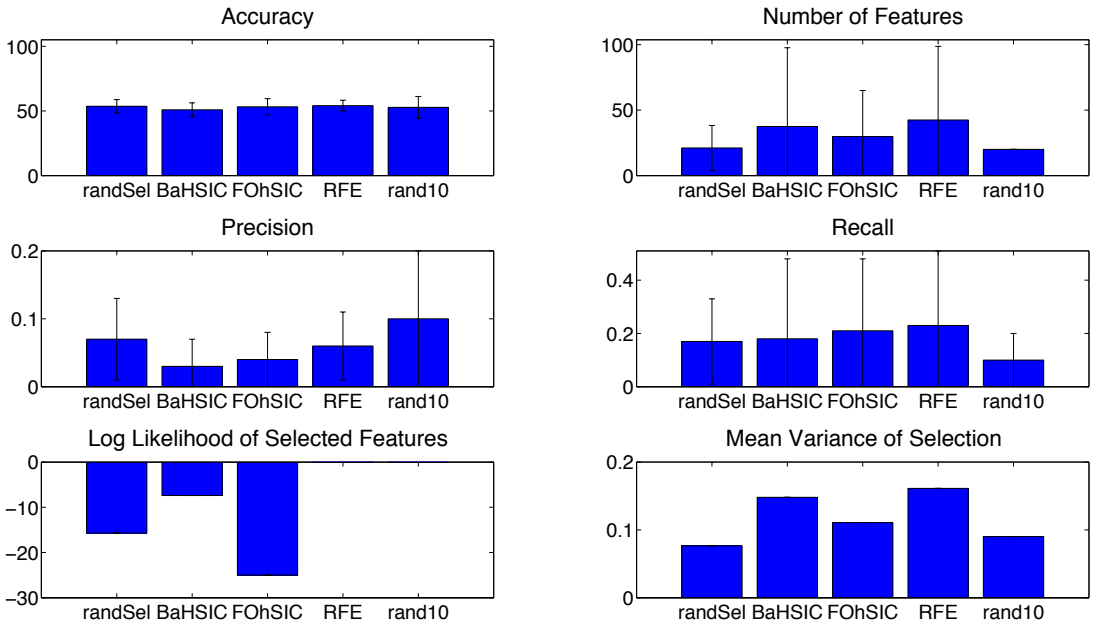

Figure 4.4: Results for the fake class dataset.

4.3.1.2 Linear Zhang With Feature Noise

This situation is reversed in the linear Zhang dataset with feature noise(figure 4.5). In terms of generalization accuracy, the resulting models are almost indistinguishable, with RFE having a minor edge.

In terms of selected features, all methods falsely identify a substantial number of probe variables as relevant, with RFE being the worst offender with an average precision of 0.49. This is also an issue for randSel, although on average it achieves the highest precision of 0.71 and is the only method that achieved perfect recall. This however comes at a cost in terms of the concistency of the recovered solution, where it is very close to RFE. The two greedy HSIC variants tend to be sparser and much more stable on average, with FoHSIC being the sparsest, and BaHSIC the all-around more consistent method.

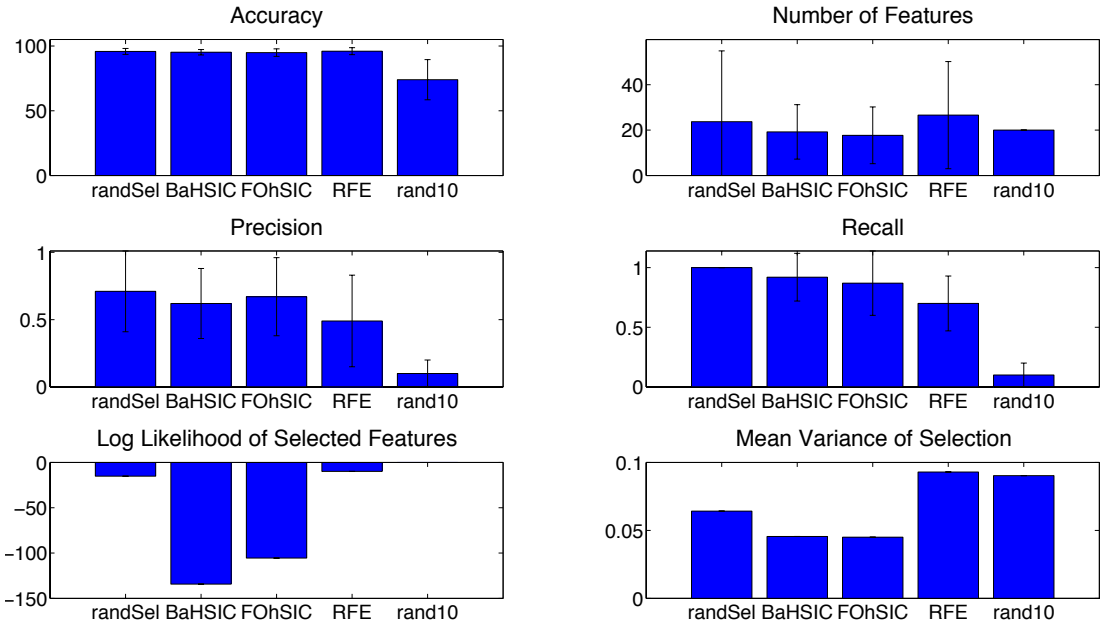

Figure 4.5: Results for the linear Zhang with feature noise dataset.

4.3.1.3 Linear Zhang With Sample Noise

The linear Zhang dataset with sample noise (figure 4.6), paints a much improved picture in terms of selection precision for most algorithms. Classification performance is indistinguishable among the different algorithms. In terms of precision and sparsity, FoHSIC is a clear winner, however this comes at the cost of a slightly diminished performance in terms of recall, where randSel fairs slightly better than all competing algorithms. FoHSIC also appears to be the most stable in terms of the solution it recovers, followed by randSel and BaHsic. RFE, yet again picks a number of probe variables, which reflects negatively in its stability profile. FoHSIC is the most consistent according to both metrics of consistency, with randSel a close second.

4.3.1.4 Linear Weston

In the linear Weston dataset (figure 4.7)the precision of all HSIC variants is further improved, with randSel achieving near perfect selection precision. All three HSIC variants on average produce very sparse and stable solutions with randSel illustrating a marginal advantage. RFE although generally competitive in terms of classification accuracy, for some folds of the data fails catastrophically something that is reflected in the large deviation in the number of variables it selects. The same fact is responsible for the substantially diminished stability profile of the algorithm, however RFE outperforms the other

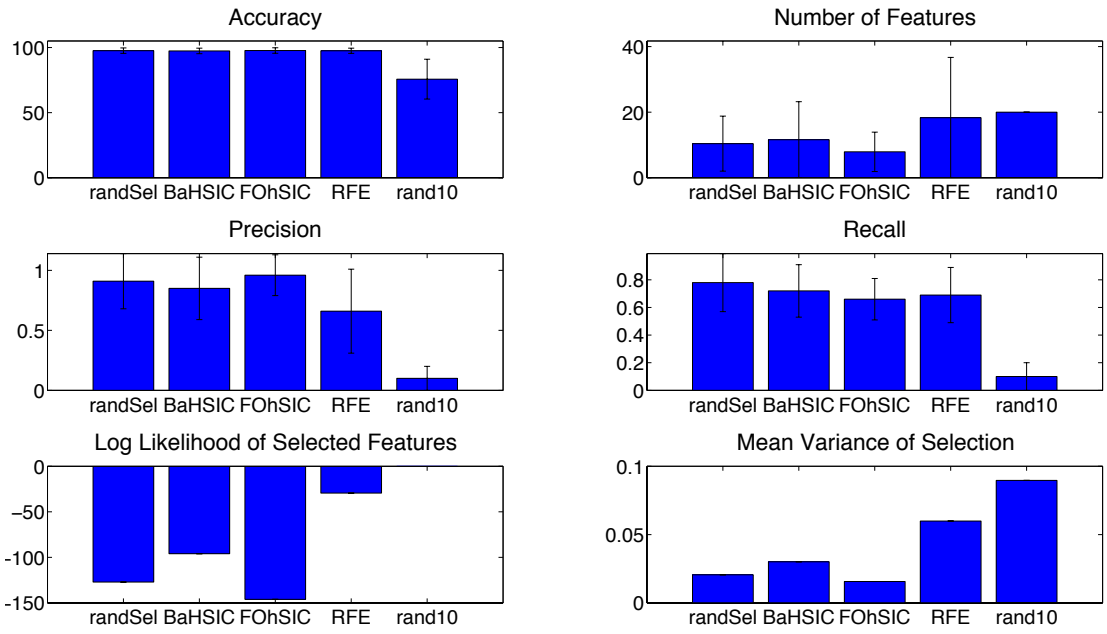

Figure 4.6: Results for the linear Zhang with sample noise dataset

algorithms in terms of recall. In terms of selection consistency, randSel appears to be more stable than competing approaches.

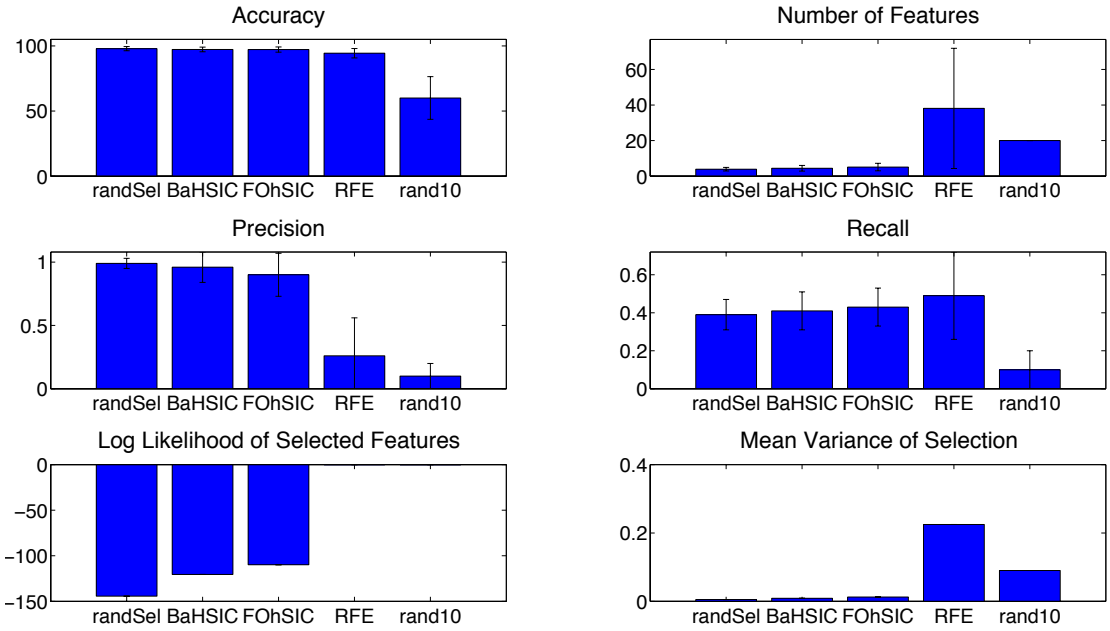

Figure 4.7: Results for the linear Weston dataset.

4.3.1.5 Non Linear Weston

In the case of the non linear Weston dataset (figure 4.8), all algorithms perform exceptionally, with RFE eeking a 0.1% advantage in terms of generalization accuracy. RandSel and FoHSIC achieve perfect precision in terms of recovered variables, with FoHSIC being slightly sparser on average, and randSel being the most stable in terms of recovered features. Yet again, RFE achieves the highest recall, followed

by randSel and the other HSIC variants. RandSel and FoHSIC are very similar in terms of consistency, with randSel having a slight advantage, as illustrated by the log likelihood.

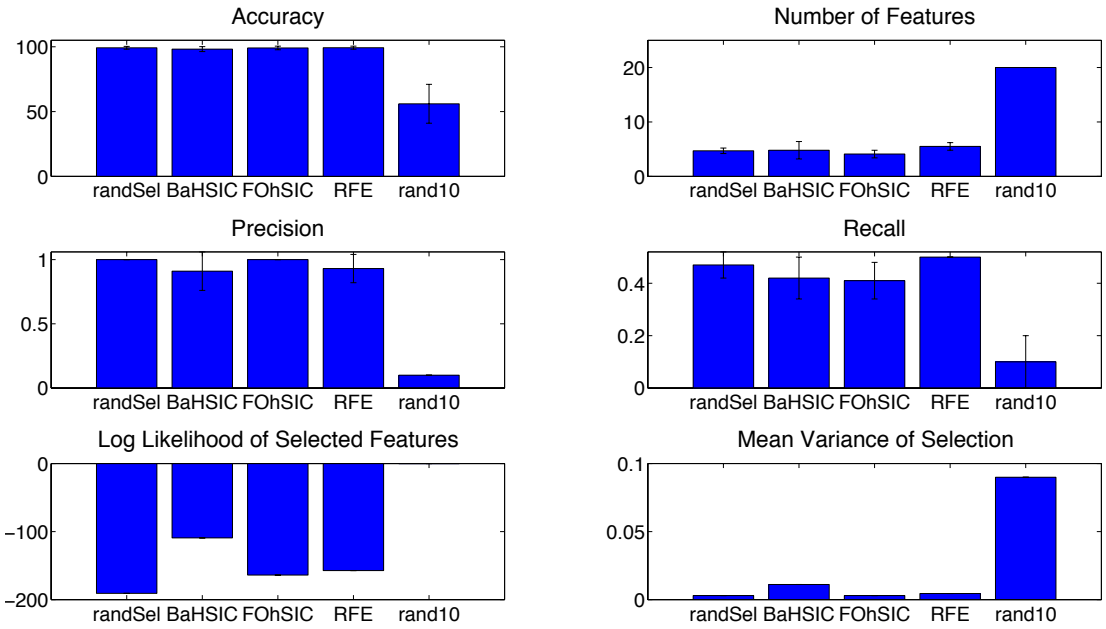

Figure 4.8: Results for the non linear Weston dataset.

4.3.1.6 XOR

Finally the XOR benchmark (figure 4.9) illustrates a case where the myopic strategy employed by FoHSIC, wherein the algorithm fails to consider a variable in the context of other included variables backfires catastrophically. FoHSIC is the only variant that fails catastrophically across the board, with randSel and BaHSIC being indistinguishable and RFE following in accuracy and number of features. Interestingly randSel and BaHSIC achieve identical performance on every captured metric, being tied in accuracy, sparsity, precision, recall and accordingly the stability of their recovered solutions.

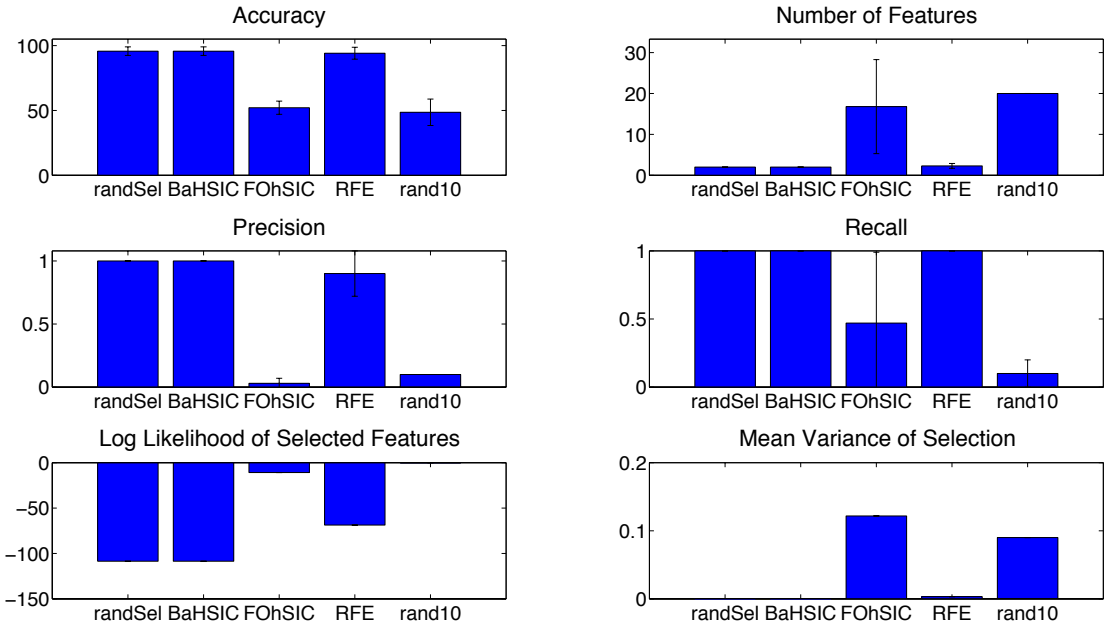

Figure 4.9: Results for the XOR dataset.

4.3.2 Real Data

4.3.2.1 TB - Task 1

In the first TB Classification task (figure 4.10), most methods perform similarly in terms of generalization, with randSel being the winner in terms of generalization accuracy. RandSel is also the winner in terms of sparsity among the different selection strategies. In terms of stability FoHSIC appears to be better, followed by randSel.

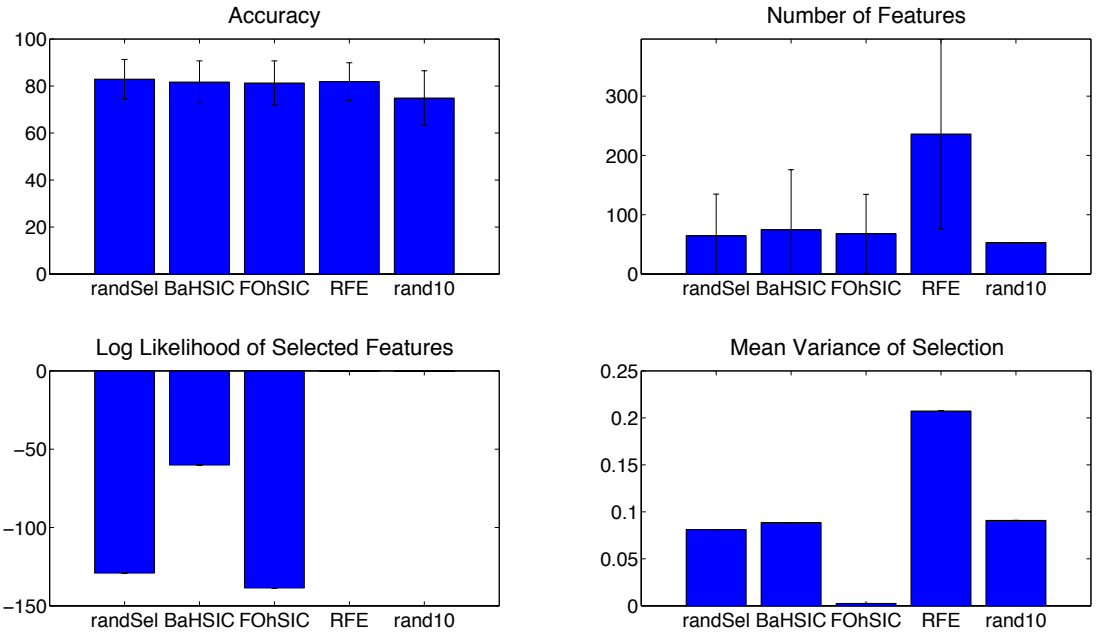

Figure 4.10: Results for the first TB task.

4.3.2.2 TB - Task 2

On the second TB task (figure 4.11), randSel is again the winner in terms of accuracy. In terms of sparsity and consistency, FoHSIC fairs substantially better than most of its competitors, followed by BaHSIC. RFE typically results in the selection of a larger group of variables, with substantial variance in the selection consistency for the resulting group of features. This is reflected in its substantially diminished consistency.

4.3.2.3 TB - Task 3

On the third TB Task (figure 4.12), the HSIC variants are indistinguishable in terms of generalization accuracy. The sparsest solutions are on average produced by RFE, although a substantial ammount of variation enters into the set of recovered solutions. In terms of stability, BaHSIC and FoHSIC outperform the other algorithms.

4.3.2.4 TB - Task 4

The fourth TB Task (figure 4.13) concludes the mass spectrometry benchmarks, with BaHSIC being the only method to reach an accuracy over 60%. Most selection methods have abysmal consistency, with FoHSIC's greedy forward approach affording the algorithm a modicum of consistency.

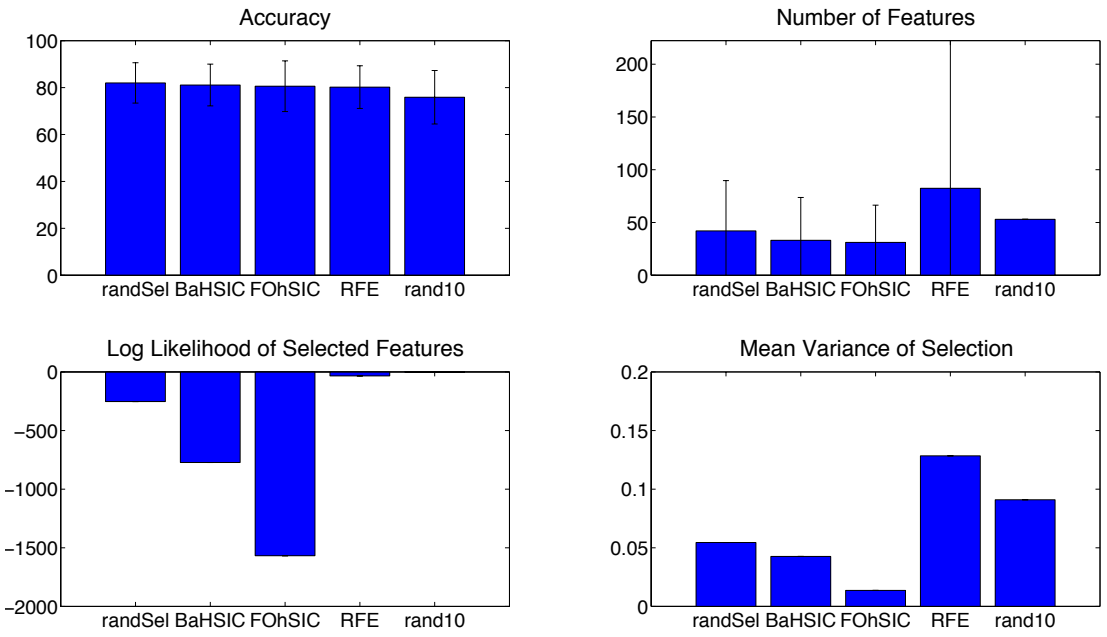

Figure 4.11: Results for the second TB task.

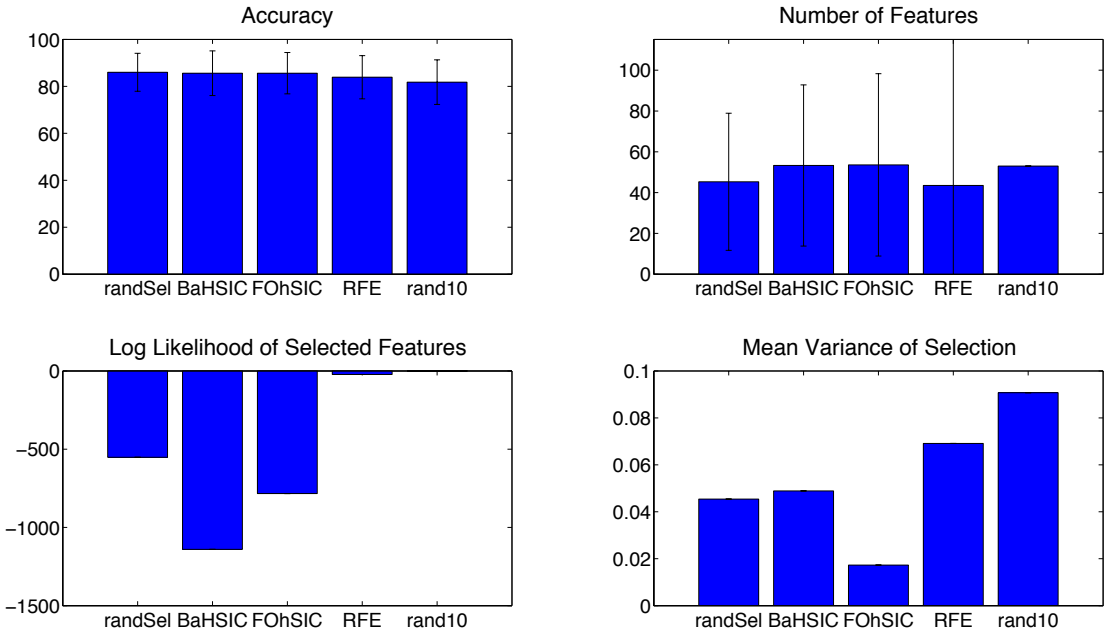

Figure 4.12: Results for the third TB task.

4.3.2.5 TB - Micro-Array

Finally on the TB Micro-Array dataset (figure 4.14), randSel surpasses the other methods in terms of accuracy. RandSel follows FoHSIC in terms of sparsity, although FoHSIC typically results in much more consistent solutions. BaHSIC is second in terms of accuracy, and stability, and is close to the other HSIC variants in terms of sparsity. Finally, RFE is between FoHSIC and BaHSIC in terms of performance, but performs abysmally in terms of sparsity and consistency.

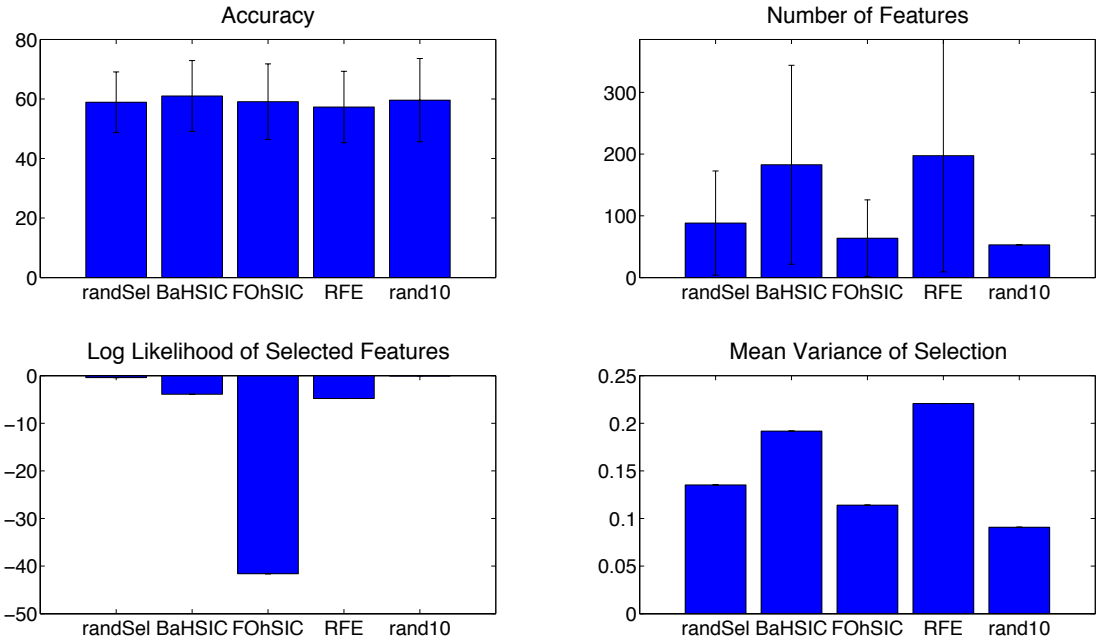

Figure 4.13: Results for the fourth TB task.

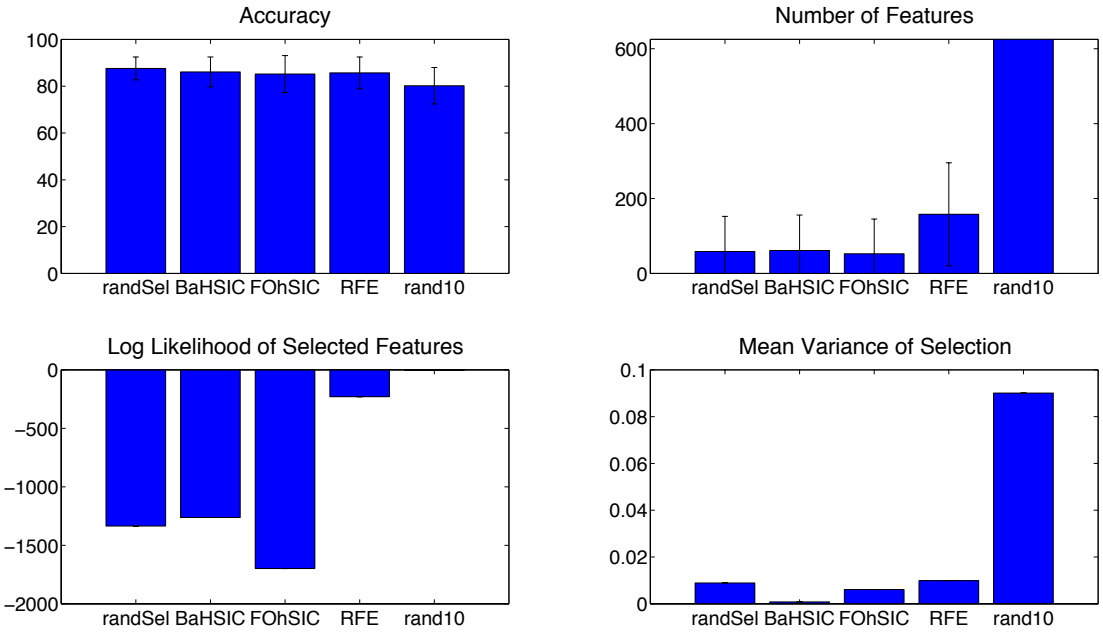

Figure 4.14: Results for the TB Micro-Array task.

4.4 Discussion

This chapter proposes randSel, a new algorithm for non-linear feature selection based on randomised estimates of HSIC. RandSel, stochastically estimates the expected importance of features at each iteration, proceeding to cull a large proportion of uninformative features at the end of each iteration. We compare randSel, with FoHSIC and BaHSIC which also utilise HSIC for feature selection. In addition we included gaussian kernel based RFE in our experimental comparisons. For most of the experiments the HSIC variants produced competitive results. On many instances RFE exhibited bad experimental results.

This behavior is interesting as RFE is closely linked to HSIC. However, it is not surprising. Whereas, other HSIC variants utilise the entirety of the dataset and operate as filters. RFE is a wrapper method that is tightly coupled with the use of a support vector machine. As a result of this, RFE only considers the set of support vectors in estimating the relevance of a feature. Given that the set of support vectors is comprised of more atypical examples (for which the constraint of the convex optimization problem is active), using them to infer the relevance of variables is marred with a larger degree of bias. Furthermore, RFE requires the setting of a regularization parameter C, a choice which substantially affects the support vectors and the resulting ranking of variables by RFE.

None of the experiments on real datasets presented in this chapter play to HSIC's strengths. The real-world datasets we employ arise in infectious disease modelling. Consequently, all experiments performed in this section are binary classifications for small sample sizes where strong linear relations can potentially overshadow non-linear interactions between the observed variables and the target classes. This echoes findings in [SSG$^+$12], where a linear method exhibited the best performance in the majority of the extensive experimental evaluations. It appears that this is also the case for the experiments presented in this section, with linear methods capturing the majority of information to be found in most datasets. Nonetheless, [SSG$^+$12] illustrated a small number of cases where the identification of non-linear interactions was beneficial, a property which will be illustrated in the following chapter. This observation means that for a large number of datasets, surpassing the performance of established linear methods is not in any way trivial. Comparing the results in this chapter with the cross validation results in section *3.3* there is no clear winner in terms of generalization performance, although the HSIC variants tend to produce sparser solutions on average.

A further issue in nonlinear modelling for the sample sizes examined in the last two chapters is the fact that the gaussian kernel offers a much richer representation in comparison to simple linear modelling. Such a representation in combination with the small number of samples can adversely effect the inferred relations. A clear example of this is comparing the performance of linear and nonlinear RFE on the TB Micro-Array dataset, on average resulting in significantly more accurate, interpretable and stable decision rules.

These observations do not indict the use of HSIC variants, or nonlinear feature selection. On the contrary, they simply suggest cautious examination of the modelling assumptions. HSIC variants more than hold their own in most experimental comparisons, and performed remarkably well in the datasets

where nonlinear dependency detection was a requirement. HSIC provides a versatile approach, with good finite sample behaviour that also generalizes naturally to structured domains. If detecting arbitrary nonlinear dependence or accounting for structure is a requirement, then the approach advocated in this chapter is more than sensible.

The true strengths of HSIC based methods are the fact that they can infer dependence relations on arbitrary domains and provide finite sample estimates, while only requiring a set of kernel hyper parameters, in terms of parameter-tuning. From a theoretical standpoint, randSel inherits many of the properties of HSIC. The theoretical analysis suggests strong guarantees for the expected performance of this procedure which is was demonstrated by testing on a number of real and artificial datasets. This, combined with the algorithm's attractive scaling properties make randSel a strong proposition for use in application areas where the volume of data increases at a frantic pace. RandSel's exact time requirements depend on user supplied parameters, however the use of sampling provides a more transparent exchange between expected performance and time requirements, that is trivial to parallelize. The next chapter gives an example where randSel's strengths are instrumental to producing improved results.

4.5 A weighting scheme for randomized feature selection

A prior attempt to leverage randomized splitting of the features, was based on exploiting the anticorrelation in the alignment of random splits of the variables. This section briefly summarises this approach. Assume that the kernel $\mathbf{K}$ on input variables can be written as a combination of l kernels, $\mathbf{K}^{(1)}, \ldots, \mathbf{K}^{(l)}$, each defined on a small subset of the variables. In the case of a Gaussian kernel this combination would correspond to the product of the component kernels. We consider here the simpler case of taking the sum of the kernels in order to illustrate the approach:

$$\mathbf{K} = \sum_{j=1}^{l} \mathbf{K}^{(j)}.$$

In this case the unnormalised empirical alignment for the kernel $\mathbf{K}$ is equal to

$$\mathbf{y}^T \mathbf{K} \mathbf{y} = \sum_{j=1}^{l} \mathbf{y}^T \mathbf{K}^{(j)} \mathbf{y}.$$

Let $\mathbf{a}_1$ and $\mathbf{a}_2$ be vectors whose entries correspond to the resulting alignment of randomly partitioning the data into 2 separate sums n_s times. For each of the n_s random partitions we have $\hat{a} = \sum_j \mathbf{y}^T \mathbf{K}^{(j)} \mathbf{y} = \sum_j b_j^{(1)} \mathbf{y}^T \mathbf{K}^{(j)} \mathbf{y} + \sum_j b_j^{(2)} \mathbf{y}^T \mathbf{K}^{(j)} \mathbf{y} = a_{1i} + a_{2i}$, for $b^{(1)}$ and $b^{(2)}$, the binary vectors defining the two parts of the partition. There are two simple outcomes:

- If there is a negative correlation between $\mathbf{a}_1$ and $\mathbf{a}_2$, it would suggest that some of the input variables exhibit a statistical dependence on the target output and that is reflected in the corresponding alignment. Accordingly this suggests a mechanism through which the presence or absence of relevant variables in each random set might be detected .

- If no statistically significant correlation between $\mathbf{a}_1$ and $\mathbf{a}_2$ can be detected, it would suggest that no meaningful statistical dependence can be inferred between individual variables and the target output.

Indeed, it is straightforward to prove in our simplified case that $\mathbf{a}_1$ and $\mathbf{a}_2$ must be anticorrelated, unless the features are independent of the target output.

Proof. For every iteration i we have $a_{1i} + a_{2i} = \hat{a}$, where $\hat{a}$ is a constant. The covariance of $a_1$and a_2 is

$$\begin{aligned}
\frac{1}{n_s}\sum_{i=1}^{i=n_s}(a_{1i} - E(a_1))(a_{2i} - E(a_2)) &= \frac{1}{n_s}\sum_{i=1}^{i=n_s}(a_{1i} - E(a_1))(\hat{a} - a_{1i} - E(\hat{a} - a_1)) \\
&= \frac{1}{n_s}\sum_{i=1}^{i=n_s}(a_{1i} - E(a_1))(\hat{a} - E(\hat{a}) - a_{1i} + E(a_1)) \\
&= \frac{1}{n_s}\sum_{i=1}^{i=n_s} -(a_{1i} - E(a_1))^2 \\
&\leq 0,
\end{aligned}$$

using $E(\hat{a} - a_1) = E(\hat{a}) - E(a_1)$, by linearity of expectation, and $E(\hat{a}) = \hat{a}$, by definition. □

Having established the anticorrelation property for different sets of variables, this approach breaks down to two steps. The first step entails generating n_s random partitions and for each iteration estimating the resulting difference $z_i = a_{1i} - a_{2i}$ in alignments between them. This is illustrated in algorithm 7.

Algorithm 7 createRandomEstimates

Input: n-dimensional dataset X, number of iterations n_s, input kernel function $\kappa(x, x')$, output kernel K_Y, kernel bandwidth σ.

for $i = 1, ..., n_s$ **do**

$p \leftarrow$ random permutation of the vector $(1, \ldots, n)$

$p \leftarrow \{p_1, ..., p_{n/2}\}$

$p' \leftarrow \{p_{n/2+1}, ..., p_n\}$

$a_1 \leftarrow$ HSIC($X^{(p)}, K_Y$,σ)

$a_2 \leftarrow$ HSIC($X^{(p')}, K_Y$,σ)

$z_i \leftarrow a_1 - a_2$

for $j = 1, ..., n/2$ **do**

$b_{i,p_j} = 1$

$b_{i,p'_j} = -1$

end for

end for

Return: Partitioning indicator matrix $\mathbf{B}$, per iteration differences in alignment, $\mathbf{z}$.

The final step of this approach, is to estimate which variables account for substantial differences in alignment between the two partitions. To achieve this, we rely on an ϵ-insensitive regression, that acts as a threshold to the deviations z_i, effectively ignoring any difference in alignment which is below a user-specified threshold. This is achieved by solving the following convex program:

$$\textbf{minimize} \sum_{i=1}^{i=n} \max(|\langle \mathbf{b}_i, \mathbf{w} \rangle - z_i| - \epsilon, 0) + \lambda \|w\|_1$$
$$\text{subject to}$$
$$w_i \geq 0$$

where ϵ is the parameter for the ϵ-insensitive loss function, and λ a regularization parameter. The non-zero entries in the solution vector $\mathbf{w}$ indicate the active variables that should be selected.

When using this approach with a gaussian kernel two problems arise. Firstly, although for the gaussian kernel, randomised splitting and estimation of alignment, does in general lead to anticorrelated estimates for the alignment of random subsets, the condition that $\hat{a} = a_1 + a_2$, does not hold, as the gaussian is a product, rather than sum of component kernels. In addition, initial experimentation indicated substantial sensitivity to the parameters ϵ and λ.

Chapter 5

Deep(er) Learning

This chapter represents a shift in focus from work in previous sections. Whereas previous chapters focused exclusively on feature selection, this chapter illustrates some of the characteristics that make randSel an attractive proposal in the somewhat novel context of deep learning. In recent years, deep learning has risen to prominence as deep architectures show a substantial degree of success in many application areas. A key strength of deep architectures is their ability to learn multiple levels of nonlinear representations of their input. This property however, comes at the cost of increased complexity. This chapter advocates a simpler approach, relying on a simple, "good-but-not-optimal" learned representation and leveraging non-linear feature selection to improve prediction.

This approach ranked third out of 218 competing proposals in the recent ICML black box learning competition. However problems in the biological domain remain the subject of the thesis. To this end, this chapter introduces a novel application of representation learning to biological sequence prediction. The proposed architecture is applied to the Signal Peptide Cleavage Site prediction problem, with substantial statistical performance gains over the original representation.

5.1 Introduction

Often the original features may not be sufficiently expressive for many tasks. The idea of introducing feature maps $\phi(x)$ that transform inputs that may be more useful for prediction is not new. In fact, it is the foundation of kernel methods. A clever choice of features enables superior predictions to using the original representation. Representation learning takes this notion a step further, allowing for the feature map to be parametrised by a set of values that may be trained and tuned.

The ability of representation learning to encapsulate substantial relational information among the observed variables is pivotal for deep learning. Representation learning techniques, such as RBMs [HOT06] and sparse auto-encoders[MRBL07], have flourished in recent years. Recent literature provides numerous instance of successful applications of representation learning in novel areas. Despite the recent success stories however, the successful application of representation learning somewhat remains a black art. In large part this stems from the large number of parameters and variety of components that are available to the practitioner. Current literature provides scant actionable guidance for integrating the large number of available components and tuning their respective parameters.

The lack of guidance is particularly pronounced in the field of deep learning where multiple layers of representation learning mechanisms are present.

To the uninitiated, current results in literature may often appear confusing, even contradictory, with minor parameter tuning leading to dramatic changes in the performance of seemingly similar algorithms. While the reported results underline the performance gains associated with representation learning, its effective application may often necessitate a substantial investment on parameter tuning. This issue has been noted in the literature by a number of practitioners, as for example in the review article [BCV13], or [NKC+11].

This chapter proposes a different approach, that is motivated by overall simplicity. The proposed architecture, utilises sparse filtering4 [NKC+11], a simple algorithm that optimizes the sparsity of ℓ_2-normalized learned representations. Instead of learning a complicated deep representation, the architecture learns a representation that is "good enough" but not optimal. The main advantage of sparse filtering is that it only requires a single parameter, which is the number of features to learn. In this way, the approach trades some degree of sub-optimality in the learned representation for simplicity. Issues arising from the sub-optimality of the learned representation are rectified further down the pipeline, by careful application of feature selection and a strong kernel based classifier.

The confluence of representation learning and kernel methods is the subject of a growing body of work with many reported encouraging results [WRMC12], [SH07]. Our proposed architecture, illustrated in figure 5.1, leverages numerous aspects of this confluence and comprises the following three steps:

1. Learn a sparse, over-complete representation of the observed data.

2. Select a series of informative non-linear combinations of learned features, which are progressively fine-grained in relation to the learning problem.

3. Learning an ensemble classifier, using kernels defined on the previously selected non-linear combinations of features.

The rest of this chapter is organised as follows: Section 2 presents Sparse Filtering and its use in our architecture. Section 3 reviews the use of randSel in conjunction with the prediction mechanism. Section 4 presents results on the ICML Black Box learning challenge. Section 5 introduces the signal peptide cleavage site prediction problem, and the corresponding experimental pipeline and results. A brief discussion section concludes the chapter.

5.2 Feature Selection for learned representations

Unsupervised feature learning algorithms such as sparse filtering are typically used to learn an over-complete representation of the data. Over-completeness refers to learned representations that learn a number of features that exceeds the dimensionality of the original dataset.

Sparse filtering learns such a representation by optimizing the sparsity of ℓ_2-normalized learned features. Concretely let $f_i^{(j)}$ represent the i^{th} learned feature value for the j^{th} example, with $f_i^{(j)} =$

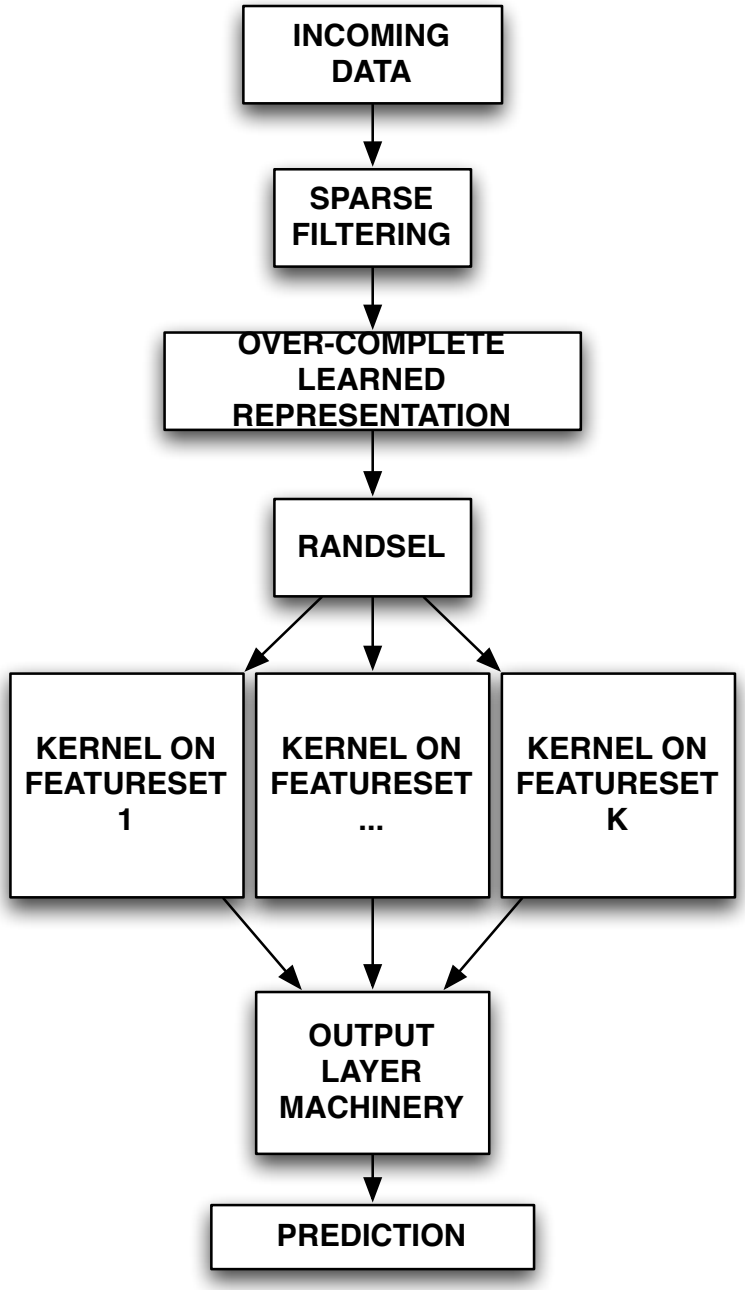

Figure 5.1: Overall architecture; randSel is applied on the features learned by sparse filtering, producing a number of nonlinear combinations of learned features of increasing granularity. A number of kernels is defined on these nonlinear combinations of features, and multiple kernel learning is used for the overall prediction.

$w_i^T x_j$. The method iterates through the following three steps until convergence:

1. Normalize each learned feature value by its ℓ_2-norm across all examples. This for the i^{th} feature becomes:

$$\tilde{f}_i = f_i / \|f_i\|_2$$

2. Normalize these values by sample. For the j^{th} sample this becomes:

$$\hat{f}^{(j)} = \frac{\tilde{f}^{(j)}}{\|\tilde{f}^{(j)}\|_2}$$

3. Optimize what is termed population sparsity, For a dataset comprising m samples:

$$\text{minimize} \sum_{j=1}^{m} \|\hat{f}^{(j)}\|_1 = \sum_{j=1}^{m} \left\| \frac{\tilde{f}^{(j)}}{\|\tilde{f}^{(j)}\|_2} \right\|_1$$

   Here the population sparsity limits the magnitude of learned features for the j^{th} example. This step reduces the number of learned features that are active for the j^{th} sample, leading to only a small number of features being used for the representation of each sample in the dataset.

The representations learned according to sparse filtering are then passed to randSel for feature selection. The depth of a deep learning architecture refers to the composition of different levels of non-linear operations in the learned function. This suggests that the feature selection component of the architecture, which is utilised to refine the initial learned representations, would substantially benefit from capturing non-linear interactions between the learned features.

The previous chapter's survey of non-linear feature selection methods provides a number of options. The decision of utilising randSel for feature selection is predicated on the following properties:

- *S*calability. From the methods examined in the previous chapter, randSel is the most readily applicable to a large sample size, a key property for the signal peptide dataset, which involves roughly 100,000 samples. The only other non-linear method that can cope with this sample size on standard commodity hardware is RFE, owing to the fact that the active set of support vectors is substantially smaller, comprising roughly 5,000 of the original 100,000 samples.

- *S*tructure Relevancy. The algorithm is readily applicable to domains that have some structure, such as the multi-class structure of the black box challenge. This, property is shared among all the HSIC-variants.

- *G*ranularity. Granularity refers to a property that was illustrated when considering model selection strategies for randSel. At the end of each iteration the algorithm returns a list of the remaining features and their expected contributions, which in our architecture leads to a series of kernels of increased granularity. Arguably a similar strategy could be developed for other non-linear feature selection algorithms, however randSel is the only algorithm that readily provides kernels of increasing granularity, which can be a highly desirable when using MKL for the final prediction.

5.3 Prediction

The feature selection layer produces progressively fine-grained combinations of features. A prediction mechanism effectively utilising the increasingly granular combinations of features comprises the last step of our approach, where we take a boosting approach based on LPBoost-MKL[FHST10]. LPBoost-MKL is a simple extension of the LPBoost algorithm presented in section 3.2.1, where the t-th weak learner employs a kernel to produce a prediction of the form:

$$h_t(x_j) = \sum_{i=1}^{m} u_i^{(t)} y_i \kappa_{s_t}(\mathbf{x}_i, \mathbf{x}_j),$$

and the vector $\mathbf{u}^{(t)}$ corresponds to the optimal dual parameters of the optimization program 5.1, and s_t is the index of the kernel selected at iteration t.The overall classification rule has the form of a linear combination of weak predictors

$$f(x) = \sum_t a_t h_t(\mathbf{x}).$$

The architecture proceeds by building a number of Gaussian kernels. Each kernel is parametrised by the different sets of features S_i they are defined on, and a kernel bandwidth parameter σ. Thus each individual kernel has the form

$$\kappa_{(S_i,\sigma)}(\mathbf{x}, \mathbf{x}') = \exp(-\sigma \|\mathbf{x}^{(S_i)} - \mathbf{x}'^{(S_i)}\|^2),$$

where, $\mathbf{x}^{(S_i)}, \mathbf{x}'^{(S_i)}$ are vectors containing only variables included in the set S_i. For small sample sizes, as was the case with the contest dataset, it is perfectly possible to store all these different kernels in memory. However, storing a single approximately $100,000 \times 100,000$ kernel matrix for the signal peptide dataset is prohibitive in most modern commodity hardware. The use of kernel ridge regression with subsampling addresses this limitation. Rather than use LPBoost-MKL, we precompute a set of weak learners derived by applying kernel ridge regression to balanced subsamples of the training data. By using kernel ridge regression each individual weak predictor takes the form:

$$h_t(\mathbf{x}) = \sum_i a_i \kappa_{s_i,\sigma}(\mathbf{x}_i, \mathbf{x}),$$

where $\mathbf{a}$ is the solution to the ridge regression problem $(\mathbf{K}_{s_t,\sigma} + \lambda \mathbf{I})\mathbf{a} = \mathbf{y} \Rightarrow \mathbf{a} = (\mathbf{K}_{s_t,\sigma} + \lambda \mathbf{I})^{-1}\mathbf{y}$, and λ a ℓ_2 regularization parameter.

The algorithm then computes the classification rule through the following linear program:

$$\begin{aligned} &\text{minimize } \beta \\ \text{s.t. } &\sum_{i=1}^{m} u_i y_i h_{ij} \leq \beta \\ &\sum_{i=1}^{m} u_i = 1 \\ &0 \leq u_i \leq D \end{aligned} \tag{5.1}$$

Where D is a ℓ_1 regularization parameter. Provided a sensible range of kernel bandwidths σ is specified, the final LPBoost classifier only requires tuning the regularization parameter D. In our search for

simplicity, this is a tangible benefit, substantially reducing the search space of parameter combinations, to tuning this single regularization parameter.

There is one remaining concern that the prediction mechanism needs to address. The weak learners' predictions are highly correlated between them. LPBoost is a sparse selection algorithm. Work on the irrepressentability condition of the lasso and related sparse selection methods [DET06] has shown that this can lead to recovering an inconsistent solution. Our proposal addresses this problem by a simple use of randomisation. Concretely, we train a number of LPBoost-based classifiers. For each classifier we randomly omit half of the weak learners. We select as our predicted class, the majority class as voted by the randomised classifiers.

5.3.1 Results on the ICML Black Box Learning Challenge

We used our architecture in the recent ICML 2013 Challenges in Representation Learning Black Box Learning challenge [Cha13]. The dataset used in the challenge was an obfuscated subset of the Street View House Numbers dataset [NWC+11]. The original where projected down to 1875 dimensions by multiplication with a random matrix, and the organizers did not reveal the source of the dataset until the competition was over. The training set comprises only 1,000 labelled samples, while an additional 130,000 samples where provided for the purposes of unsupervised pre-training.

For our submissions, cross validation was used to select the number of features to learn with sparse filtering, with our best solution using a set of 625 learned features. Randsel was then used to select combinations of the 625 learned features dropping 12.5% of the least contributing features at the end of each iteration. The resulting set of 34 different sets of features was combined with 75 different σ parameters to result in 2550 weak learners. The output of a randomly selected half of these weak learners was used to train 500 different LPBoost-MKL classifiers, and the majority vote of these 500 classifiers resulted in our final set of predictions. The regularisation parameter D was also set through cross validation. This approach led to a generalisation accuracy of 68.44% on the public and 68.48% on the private leaderboards, ranking third in both cases out of a total of 218 teams.

5.4 Application to cleavage site prediction

This section examines the application of the proposed architecture to the signal peptide cleavage problem. Signal peptides are amino-acid sequences found in transported proteins that selectively guide the distribution of the proteins to specific cellular compartments. Often referred to as the zip-code sequences, owing to their role in sub-cellular localization, a substantial body of work is devoted to predicting the cleavage site of signal sequences. Current literature establishes the importance of a number of physico-chemical properties of the signal sequence in determining the cleavage site location. The experimental pipeline presented in this section further supplements this approach, by learning a feature representation of multiple physicochemical property encodings.

5.4.1 Experimental Pipeline

The Predisi dataset [HGS+04] of eukaryotic signal sequences was used for experimentation. Initial filtering produced a dataset of 2,705 unique signal peptide sequences, with a sequence length of 50

amino-acids. The approach used for cleavage site prediction breaks each individual sequence into smaller windows. Cross validation was used to estimate the parameters relating to the window size. The resulting convention was to use windows that contain 9 aminoacids prior to what we deem the target of the window and 2 aminoacids following that position. For an individual prediction to be considered accurate, the window predicted as most likely to contain the cleavage site in its target position, must coincide with the actual window containing the cleavage target site for the sequence.

With the window parameters chosen through cross validation, this results in each individual sequence of 50 aminoacids producing 39 windows with a length of 12 aminoacids each. The resulting process generates a dataset comprising of 105,495 windows. The entirety of 54 distinct physicochemical encodings offered by the Matlab bioinformatics toolbox was used for numerical representation of each sequence window, which is then represented by a 648-dimensional vector of physicochemical properties.At this point, sparse filtering learns an overcomplete representation comprising of 1500 learned features. The complete process, illustrated in figure 5.2 generates a dataset comprising of 105,495 1500-dimensional samples.

B

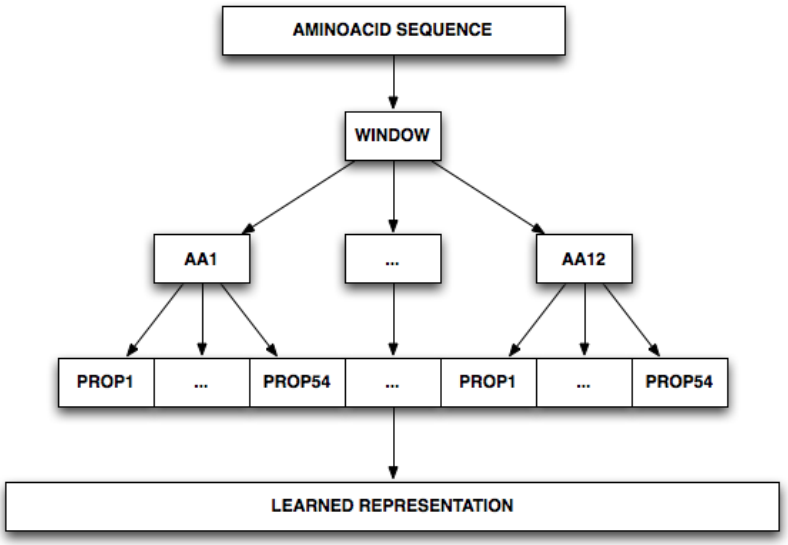

Figure 5.2: How the learned representation is generated. The amino-acid sequence is broken into smaller windows. Each amino-acid in the window is represented by its 54 distinct physicochemical properties. Sparse filtering is used to learn a representation for this encoding.

5.5 Experiments

There are three interesting questions which the experiments where designed to address. Concretely, the experimental comparisons are designed to establish the performance gains from using a learned representation, over learning in the original feature space, using randSel for feature selection as opposed to other possible feature selection methods, and finally establishing the importance of multiple kernel

learning, used for prediction.

To this end, a number of competing solutions were implemented. The shallow approach uses the original physichochemical properties for prediction. For comparing the performance of different feature selection algorithms on learned representations, ℓ_1-logistic regression stability selection and nonlinear SVM-RFE are used in addition to randSel. Finally we compare the performance of a prediction rule relying on a single gaussian kernel SVM, to the performance of randomized-MKL.

The large size of the dataset, as well as the fact that it is highly imbalanced make for some challenges. Stability selection can readily be applied to a problem of this dimensionality. While deterministic HSIC-variants are ill-equipped to deal with the size of the resulting kernel in most current commodity hardware, the use of sampling in randSel largely alleviates the problems related with size. In order to address the issue of the imbalance, subsamples where both classes are equally represented were used. The possibility of using RFE for non-linear feature selection is of particular interest. At over 100,000 samples, the size of the full dataset is prohibitive, however at each fold the size of the active set is substantially smaller, averaging aproximately 5,100 support vectors. The substantially smaller active set makes RFE feasible.

In terms of producing the actual prediction the experiments examine two options. The first is using a chunking, non-linear SVM, which is the same approach that enables the use of RFE. Using vanilla LPBoost-MKL for prediction is prohibitive, owing to the memory requirements of storing the kernel matrices. The approach to rectify this problem is to use subsampling from the negative class, combined with kernel ridge regression as a weak predictor in our boosting framework. This results in a mixed-norm (as the individual weak learners are ℓ_2-normalized, and the overall prediction rule is ℓ_1-normalized) MKL formulation which effectively addresses the limitation of not being able to store the kernel matrices. For the l_2 regularization parameter λ of individual weak predictors, a very small range of parameters was used. The l_1 regularization parameter D for the LPBoost prediction rule was set through cross-validation.

5.5.1 Results

Figure 5.3 summarizes the results for the different attempted approaches. Using the original feature representation with a non-linear SVM leads to a generalization accuracy of 67.2%. This is substantially smaller than all the approaches that rely on the learned feature representation. This suggests that there are performance gains to be had in using a learned representation.

In terms of using feature selection on the learned representation, the results indicate that randSel has an edge over the other methods, with RFE also performing marginally better than l_1-regularized logistic regression-based stability selection and randSel employing a linear kernel. The learned representation offers a case where it is reasonable to suspect benign non-linear collusion between features, something that both RFE and randSel are designed to take advantage of, and the large sample size allows for increased confidence when inferring such relationships. The fact that randSel outperforms RFE repeats some of the observations of the previous chapter. RFE's reliance on the support vectors for feature selection can negatively bias the feature selection procedure. Figure 5.4 indicates substantial differences for the correlation of learned features to the output. There is enough order information for the method

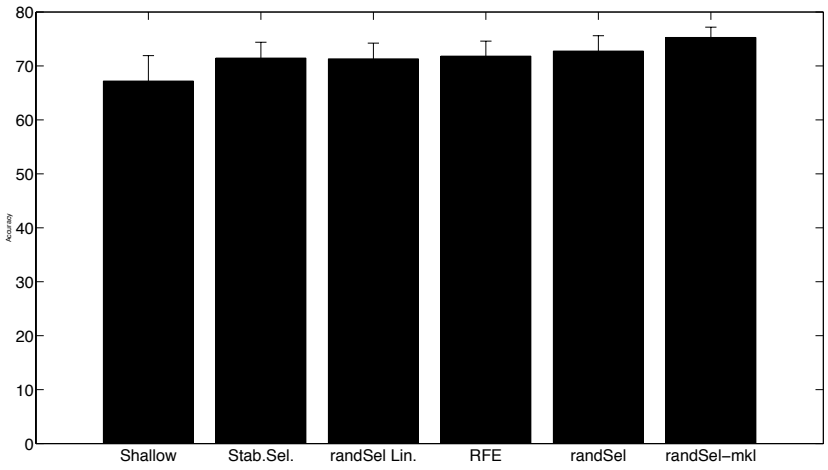

Figure 5.3: Accuracy of different feature selection and representation approaches on the signal peptide dataset. All feature selection approaches that operate on the learned representation clearly outperform the original features. Methods employing a single kernel for prediction result in similar accuracy. Combining randSel with MKL outperforms all other approaches.

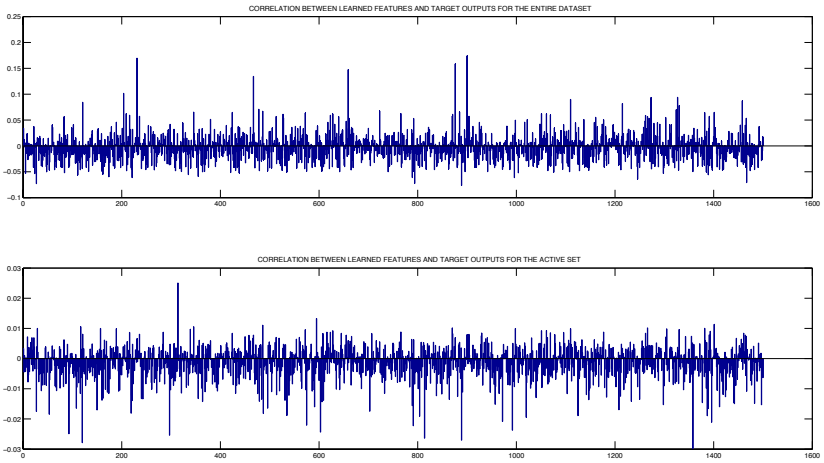

Figure 5.4: Substantial differences in correlation between full dataset (top row) and the support vectors (bottom row). This is largely due to the support vectors comprising of more atypical examples, for which the constraint of the SVM optimization problem is active. It is not visually obvious, but the two rows are very correlated, however some ordering information, which is important for feature selection, is lost in the active set. This means that while the ranking the features by correlation to the target output is largely similar between the active set and the entire set of variables, the differences in rank are substantial enough to affect the behavior of RFE.

to make useful inferences about the relevance of each variable, but as the results show, the employed approach with randSel is better suited for feature selection in this dataset. Finally, the use of MKL for prediction further improves the results. Direct comparisons to state-of-the art methods for cleavage prediction is difficult as the reported accuracy highly depends on the dataset and modelling assumptions,

such as the original sequence length. It does appear however that the method proposed here outperforms SignalP's [PBvHN11] reported accuracy of 72.9% for eukaryotic sequences.

5.6 Discussion

This chapter presented a simple system that produces a classification rule based on non-linear learned feature combinations of increasing granularity. The architecture of the system comprises a fast, unsupervised feature learning mechanism, randomised non-linear feature selection and a multiple kernel learning based classifier. The guiding principle of this approach is to use simple components that require minimal parameter tuning, with components further down the pipeline making up for the potential shortcomings upstream. Indeed, the three different constituents of this architecture, require minimal parameter tuning and scale gracefully, and the experimental results on both datasets appear to validate the approach.

While current literature already reports many encouraging results on the intersection of representation learning and kernel methods, the field remains relatively young. This simple fact encourages further exploration and opens many avenues for improvements. Such an opportunity for improvement which is particularly pronounced in the proposed architecture stems from the fact that the individual components are to a large degree insular. Improving feedback between layers would allow for improved performance at the feature learning level, in a way that is more directly related to the classification objective. The growing number of convex optimization approaches for representation learning, such as sparse coding [LBRN07] provide strong candidates that address this condition. This approach suggests the possibility of deeper overall convex architectures, with the benefits convexity entails.

In terms of applying deep learning, computational biology largely appears to be unknown territory, with the exception of a small number of image processing tasks that arise in the biological domain. This is largely a matter of culture. Deep learning is still a relatively recent development, and from the perspective of a biological science practitioner, its application in various biological domains is far from obvious. The sheer number of different approaches, in combination with their apparent complexity, further hinder the adoption of representation learning. Another important reason is that in many biological application areas representation learning does very little to aid interpretation. This particularly applies to data arising from methods such as micro-arrays. In such an example, the biological significance of an inferred micro-array feature is puzzling.

Luckily, there are certain areas in computational biology where high accuracy far outweighs any other considerations. Many tasks relating to proteomics belong to this category. The signal peptide cleavage site prediction presented in this chapter exemplifies this category, with the state-of-the-art approach, [PBvHN11] hardly offering any interpretation. Currently, searching google for applications of deep representations in biological areas other than image processing returns only a handful of existing publications (see for example [DLBN$^+$12], [EC12], [LXLF13]). For applications where accuracy is paramount, representation learning should be considered. The advent of improved methods, and better theory to drive their adoption and application promises many interesting developments for both fields.

Chapter 6

Conclusions

It is often repeated that the adoption of data-driven methods has revolutionalized biology. This is particularly true for biomarker discovery, a field where feature selection methods often play a key role. This thesis showcases the machine learning approach for biomarker discovery by building on the rich theory of max-margin learning and kernel machines. Kernel methods and feature selection are important tools for many bioinformaticians with good reason. Most chapters focus on a simple approach comprising feature selection and prediction through a nonlinear SVM. This approach is vindicated by strong experimental results even for methods that are considered as baselines in the context of this thesis, such as correlation coefficients and RFE. Throughout our work, the use of kernel target alignment features prominently, either as a model selection criterion in chapter 3, or as an empirical estimator for an independence criterion that powers the majority of the non-linear feature selection approaches in chapter 4. Furthermore, chapter 5 illustrates a case where there is a strong synergy between feature selection methods and representation learning. However, as is often the case for research, the work presented here opens up many new questions and avenues for experimentation.

6.1 Future Work

The use of kernel target alignment, often in the guise of HSIC permeates this thesis. It is therefore logical that future iterations take the latest developments in this area into consideration. For example, the vast majority of our work employing HSIC relies on the presentation of [GBSS05] and relies on the biased estimator which following the authors' conventions we denote HSIC_0. HSIC is central to a growing body of work, and some refinements for its estimation have been presented, such as the use of HSIC_1 in the work of [SSG$^+$12]. The important aspect of HSIC_0 for the feature selection work presented here is supplying order information for the relevance of variables. For most experiments the results do not indicate substantially diminished performance that can directly be attributed to the bias of the HSIC_0 estimator. However, given the fact that using HSIC_1, does not complicate most computations and is more sound from a theoretical point of view, its adoption is a reasonable next step. More pertinently, [SSG$^+$12] notes that HSIC_1, can address problems which arise from the use of diagonally dominant kernels, a property which is highly important for many applications in biology. This was also previously reported in the work of [QSST10].

Chapter 3 explores the use of HSIC as a model selection criterion in place of cross-validation. In most cases, the models resulting from HSIC produced competitive accuracy, while being sparser and more consistent in terms of recovered variables. However there are two instances in chapter 3 where optimizing for HSIC results in inconsistent variable selection, severely impacting generalization accuracy. A careful examination of the conditions leading to such inconsistencies as well as a principled method to avoid such inconsistencies could prove highly beneficial. This is an application with significant practical repercussions, as from a practitioner's point of view it can vastly simplify model selection.

Chapter 4 introduces a randomized algorithm that relies on HSIC for feature selection. Our analysis of randSel, derives some probabilistic guarantees on its performance. However, the sample sizes required for those guarantees are unrealistic for a lot of real world applications. As such, an investigation of tighter bounds is a logical next step. Another aspect of chapter 4 that merits closer examination are the definitions of relevance. The section works on the strong assumption that an irrelevant feature is one that is independent of the target output and other variables. This definition while simplifying potential analyses, is too strong, as it fails to consider interactions between variables. This is the undertaking of very recent work in [SGB13], which explores three-variable interaction. However, as the authors note, detection of higher order interactions between variables, is substantially harder to analyse.

Chapter 5 illustrated the application of feature selection on learned representations. In the search for a more principled approach to deep learning, the exploration of convex optimization approaches such as sparse coding is interesting. By providing the largest dataset studied in the thesis chapter 5 also emphasises another aspect that is of interest. The real world datasets in mass spectrometry and genomics have a relatively small sample size. This is largely due to the acquisition difficulties associated with collecting data in the biomedical domain. As acquisition costs are driven down, the performance picture for the various algorithms in this presentation may change dramatically.

Finally, we have noted that methods utilising HSIC for feature selection have distinct advantages in structured domains on a number of occasions. However, the only illustration of this property in the thesis was on the black box competition dataset. Partly responsible is the fact that most datasets in biomarker discovery are designed to answer very specific questions, typically involving a target and control class. Provided more studies begin integrating larger ammounts of information, this should prove to be a particularly fruitful avenue for future research.

Simply enumerating the applications of machine learning on biological data would be no mean feat, as it is successfully adopted in novel areas with increasing frequency. By covering feature selection for biomarker discovery and signal peptide prediction, barely scratches the surface of its potential applications. The focus on biomarker discovery is not only due to the fact that it constitutes an ideal application area, but also reflects its impact.

The study from which the TB micro-array dataset has resulted from [BGM+10], is an attempt to improve the understanding of a major global cause of mortality. According to the World Health Organization, Tuberculosis was responsible for 1.3 million deaths in 2012[1], with over 8.6 million falling

[1] http://www.who.int/mediacentre/factsheets/fs104/en/

ill. More importantly, according to the same report one third of the world's population has latent TB, which has an approximately 10% lifetime risk of becoming an active infection. Developing tools that can identify individuals who will develop active disease, has the potential to facilitate preventative therapy. A staggering body of work is devoted to identifying the underlying mechanisms of the disease, and develop better-targeted approaches for intervention. Recent approaches that attempt to directly link clinical data with biological profiling such as [RAS+13], serve to further improve the understanding of how biological data can be linked to the different clinical aspects of this disease. New screening technologies, and reduced data acquisition costs have resulted in a frantic increase in the volume of data. Feature selection methods are pivotal for making sense out of this deluge of information.

Appendix A

Appendix A

A.1 Concentration Inequalities

A.1.1 Hoeffding's inequality

Let $X_1, \ldots, X_n$ be independent random variables. Assume that the X_i are almost surely bounded; that is, assume for $1 \leq i \leq n$ that $P(X_i \in [a_i, b_i]) = 1$. We define the empirical mean of these variables $\overline{X} = \frac{1}{n}(X_1 + \cdots + X_n)$. Theorem 2 of [Hoe63] proves the inequality:

$$P(|S - \mathbb{E}(S)| \geq t) \leq 2 \exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

A.1.2 Hoeffding's concentration bound for U-Statistics

Theorem A.1.1. *Consider $n \geq r$. Let*

$$V(X_1, ..., X_n) = \frac{1}{k}\{g(X_1, ..., X_r) + g(X_{r+1}, ..., X_{2r}) + \ldots + g(X_{kr-r+1}, ..., X_{kr})\}$$

where $k = [n/r]$, the largest integer contained in n/r. Then,

$$U = \frac{1}{n!}\sum_{n,n} V(X_{i_1}, \ldots, X_{i_n})$$

Where the sum $\sum_{n,n}$ is taken over all permutations $i_1, i_2, \ldots, i_n$ of the integers $1, 2, \ldots, n$. Each term in the sum on the right is a sum of k independent random variables. Thus the expression gives a representation of U. If the function g is bounded $a \leq g(x_1, \ldots, x_r) \leq b$, then

$$P(|U - \mathbb{E}(U)| \geq t) \leq 2 \exp\left(\frac{-2kt^2}{(b-a)^2}\right) \tag{A.1}$$

A.2 Sparse Filtering Implementation

This section includes Jiquan Ngiam's implementation for spare filtering[NKC+11], which was used in the experiments of Chapter 5. More information can be found at the implementation's github page at `https://github.com/jngiam/sparseFiltering/blob/master/sparseFiltering.m`.

```
function [optW] = sparseFiltering(N, X)
```

```
    % N = # features, X = input data (examples in column)
    optW = randn(N, size(X, 1));
    optW = minFunc(@SparseFilteringObj, optW(:), ...
                   struct('MaxIter', 200, 'Corr', 20), X, N);
    optW = reshape(optW, [N, size(X, 1)]);

end

function [Obj, DeltaW] = SparseFilteringObj (W, X, N)
    % Reshape W into matrix form
    W = reshape(W, [N, size(X,1)]);

    % Feed Forward
    F = W*X;
    Fs = sqrt(F.^2 + 1e-8);
    [NFs, L2Fs] = l2row(Fs);
    [Fhat, L2Fn] = l2row(NFs');

    % Compute Objective Function
    Obj = sum(sum(Fhat, 2), 1);

    % Backprop through each feedforward step
    DeltaW = l2rowg(NFs', Fhat, L2Fn, ones(size(Fhat)));
    DeltaW = l2rowg(Fs, NFs, L2Fs, DeltaW');
    DeltaW = (DeltaW .* (F./Fs)) * X';
    DeltaW = DeltaW(:);
end
%
% The following functions are now in common/
% function [Y,N] = l2row(X) % L2 Normalize X by rows
%     % We also use this to normalize by column with l2row(X')
%     N = sqrt(sum(X.^2,2) + 1e-8);
%     Y = bsxfun(@rdivide,X,N);
% end
% function [G] = l2grad(X,Y,N,D) % Backpropagate through Normalization
%     G = bsxfun(@rdivide, D, N) - bsxfun(@times, Y, sum(D.*X, 2)./(N.^2));
% end
```

A.3 Datasets

Synthetic datasets were generated in order to enable more control in the experimental design. Stemming from the fact that analysis of differential expression in proteomic profiles produced through mass spectrometry is one of the primary fields of application for feature selection, 3 of the datasets were generated according to work in [ZLS$^+$06]. Another two of the datasets were generated according to work in [WMC$^+$00]. For each of the methods some simplifying conventions where used, namely the size of the dataset was fixed to 300 samples, each composed of 200 variables. From these 200 variables, 10 were relevant to the problem and 190 where noise. While using these conventions we generate the following datasets:

1. *Fake Class* , the relevant features where gaussians defined as $N(0.25, 1)$ for class 1 and $N(-0.25, 1)$ for class -1. The remaining,"junk", features where also gaussian $N(0, 1)$. The class labels are then randomly permuted. This dataset can act as a sanity-check for supervised feature selection algorithms.

2. *Linear Weston*, for the total of 10 relevant features the first five $\{x_1, ..., x_5\}$ where drawn as $x_i = yN(i, 1)$ and the second ten features $\{x_6, ..., x_{10}\}$ with a probability of 0.7. Otherwise the first five where drawn as $x_i = N(0, 1)$ and the second ten as $x_i = yN(i - 10, 1)$. The remaining features where gaussian noise $x_i = n(0, 20)$, $i = 21, ..., 500$

3. *Linear Zhang - Sample* , the relevant features where gaussians defined as $N(0.25, 1)$ for class 1 and $N(-0.25, 1)$ for class -1. The remaining,"junk", features where also gaussian $N(0, 1)$. The data is constructed to contain 5% "outlier samples," which were made by randomly picking 5% of the samples and increasing the standard deviation of every variable in these samples by a factor of 10.

4. *Linear Zhang - Feat* , the relevant features where gaussians defined as $N(0.25, 1)$ for class 1 and $N(-0.25, 1)$ for class -1. The remaining,"junk", features where also gaussian $N(0, 1)$. For the informative genes 5% of values in all samples were randomly used as outliers by making them to follow $N(0.25, 100)$ for positive samples and $N(-0.25, 100)$ for negative samples.

5. *Nonlinear Weston*, For $y = -1$, $\{x_1, ..., x_{10}$ where drawn with equal probability from $N(\mu_1, \Sigma)$, or $N(\mu_2, \Sigma)$, where $\mu_1 = \{-0.7500, -1.0, ..., -2.75 - 3.0000\}$ and $\mu_2 = \{-3, -2.4, ..., 2.4, 3\}$ and $\Sigma = I$. If $y = 1$, then $\{x_1, ..., x_{10}\}$ are drawn again from two normal distributions with equal probability and means $\mu_3 = \{3, 2.4, ..., -2.4, 3\}$ and $\mu_4 = \{-3, -2.4, ..., 2.4, 3\}$ and same covariance Σ as before. The rest of the features are just noise, $x_i = N(0, 20)$, $i = 21, ..., 500$.

A.4 Real Data

We conducted experiments in real datasets arising in the computational profiling of tuberculosis (TB), an application where feature selection plays a pivotal role both in terms of improving accuracy but also providing insight into the underlying mechanisms. We conducted experiments on two different

datasets. The first TB dataset consists of 523-dimensional mass-spectrometry proteomic profiles of blood plasma [SBE+12], and consists of 100 active TB samples, 40 symptomatic controls, and 49 samples of patients with TB-Like symptoms with a co-existing latent TB infection (LTBI). We performed pairwise comparisons between active TB and Unhealthy Controls *(Task 1)*, Active TB and symptomatic LTBI *(Task 2)*, and Active TB with symptomatic patients without LTBI *(Task 3)*,which correspond to scenarios in real clinical applications. The second dataset comprises of the transcriptomic profiles of 69 healthy individuals with LTBI and 133 healthy controls from [BGM+10]. Preprocessing removed probes with low acquisition precision as well as factors with missing values, resulting in a set of 6247 variables. Table 2 summarises the experiments.

A.5 Software

The majority of the code used for the experiments presented in the thesis is hosted on github. The thesis would not have been possible without some excellent open source packages, which provide the foundation for our code. The majority of the experiments rely on LIBSVM [CL11] for a SVM solver. Ryota Tomioka's DAL package [TS09] was used for solving Lasso and L_1-logistic regression problems. Tom Minka's lightspeed toolbox [Min] has boosted the performance of many routines. The linear programming boosting implementation relies on CVX [GBY] for solving the underlying linear problem.

Bibliography

[AFRP+06] Dan Agranoff, Delmiro Fernandez-Reyes, Marios C Papadopoulos, Sergio A Rojas, Mark Herbster, Alison Loosemore, Edward Tarelli, Jo Sheldon, Achim Schwenk, Richard Pollok, et al. Identification of diagnostic markers for tuberculosis by proteomic fingerprinting of serum. *The Lancet*, 368(9540):1012–1021, 2006.

[Bac08] Francis R Bach. Bolasso: model consistent lasso estimation through the bootstrap. In *Proceedings of the 25th international conference on Machine learning*, pages 33–40. ACM, 2008.

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. 2013.

[BGM+10] Matthew PR Berry, Christine M Graham, Finlay W McNab, Zhaohui Xu, Susannah AA Bloch, Tolu Oni, Katalin A Wilkinson, Romain Banchereau, Jason Skinner, Robert J Wilkinson, et al. An interferon-inducible neutrophil-driven blood transcriptional signature in human tuberculosis. *Nature*, 466(7309):973–977, 2010.

[BLJ04] Francis R Bach, Gert RG Lanckriet, and Michael I Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6. ACM, 2004.

[BPZL12] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The Journal of Machine Learning Research*, 13:27–66, 2012.

[Bro09] Gavin Brown. A new perspective for information theoretic feature selection. In *International Conference on Artificial Intelligence and Statistics*, pages 49–56, 2009.

[BY06] Peter Bühlmann and Bin Yu. Sparse boosting. *The Journal of Machine Learning Research*, 7:1001–1024, 2006.

[Cha13] Challenges in representation learning: The black box learning challenge (https://www.kaggle.com/c/challenges-in-representation-learning-the-black-box-learning-challenge), 2013.

[CL11] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[CMR12] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13:795–828, 2012.

[CSTEK01] Nello Cristianini, John Shawe-Taylor, Andre Elisseeff, and Jaz Kandola. On kernel target alignment. In *NIPS*, volume 2, page 4, 2001.

[CT07] Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, pages 2313–2351, 2007.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[DBST02] Ayhan Demiriz, Kristin P Bennett, and John Shawe-Taylor. Linear programming boosting via column generation. *Machine Learning*, 46(1-3):225–254, 2002.

[DET06] David L Donoho, Michael Elad, and Vladimir N Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *Information Theory, IEEE Transactions on*, 52(1):6–18, 2006.

[DLBN+12] Pietro Di Lena, Pierre Baldi, Ken Nagata, P Bartlett, FCN Pereira, CJC Burges, L Bottou, and KQ Weinberger. Deep spatio-temporal architectures and learning for protein structure prediction. In *NIPS*, pages 521–529, 2012.

[EC12] Jesse Eickholt and Jianlin Cheng. Predicting protein residue–residue contacts using deep networks and boosting. *Bioinformatics*, 28(23):3066–3072, 2012.

[EHJ+04] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[FHST10] Tristan Fletcher, Zakria Hussain, and John Shawe-Taylor. Multiple kernel learning on the limit order book. *Journal of Machine Learning Research-Proceedings Track*, 11:167–174, 2010.

[FS97] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[FSA99] Yoav Freund, Robert Schapire, and N Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[GBSS05] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic learning theory*, pages 63–77. Springer, 2005.

[GBY] Michael Grant, Stephen Boyd, and Y Ye. Cvx: Matlab software for disciplined convex programming, version 2.0 beta. *Recent Advances in Learning and Control*.

[GGNZ06] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and L Zadeh. Feature extraction. *Foundations and applications*, 2006.

[GWBV02] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

[HGS+04] Karsten Hiller, Andreas Grote, Maurice Scheer, Richard Münch, and Dieter Jahn. Predisi: prediction of signal peptides and their cleavage positions. *Nucleic acids research*, 32(suppl 2):W375–W379, 2004.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[HOT06] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[HRTZ04] Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. *Ann Arbor*, 1001:48109–1092, 2004.

[KKB07] Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. An interior-point method for large-scale 1-regularized logistic regression. *Journal of Machine learning research*, 8(7), 2007.

[KR92] Kenji Kira and Larry A Rendell. A practical approach to feature selection. In *Proceedings of the ninth international workshop on Machine learning*, pages 249–256. Morgan Kaufmann Publishers Inc., 1992.

[KS96] Daphne Koller and Mehran Sahami. Toward optimal feature selection. 1996.

[KTS+09] Mitsunori Kayano, Ichigaku Takigawa, Motoki Shiga, Koji Tsuda, and Hiroshi Mamitsuka. Efficiently finding genome-wide three-way gene interactions from transcript-and genotype-data. *Bioinformatics*, 25(21):2735–2743, 2009.

[LBRN07] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19:801, 2007.

[LCB+04] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, 2004.

[LLAN06] Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y Ng. Efficient l˜ 1 regularized logistic regression. In *Proceedings of the National Conference on Artificial Intelligence*,

volume 21, page 401. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[LST02] John Langford and John Shawe-Taylor. Pac-bayes & margins. In *NIPS*, volume 15, page 423, 2002.

[LXLF13] Michael K. K. Leung, Hui Yuan Xiong, Leo J. Lee, and Brendan J. Frey. Tissue-dependent alternative splicing prediction using deep neural networks. In *NIPS Workshop on Machine Learning in Computational Biology*, volume 2013, 2013.

[MB10] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[McA03] David A McAllester. Pac-bayesian stochastic model selection. *Machine Learning*, 51(1):5–21, 2003.

[Min] Tom Minka. The lightspeed matlab toolbox.

[MRBL07] Y MarcAurelio Ranzato, Lan Boureau, and Yann LeCun. Sparse feature learning for deep belief networks. *Advances in neural information processing systems*, 20:1185–1192, 2007.

[NKC+11] Jiquan Ngiam, Pang Wei Koh, Zhenghao Chen, Sonia A Bhaskar, and Andrew Y Ng. Sparse filtering. In *NIPS*, volume 11, pages 1125–1133, 2011.

[NWC+11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, volume 2011, 2011.

[OPT00] Michael R Osborne, Brett Presnell, and Berwin A Turlach. A new approach to variable selection in least squares problems. *IMA journal of numerical analysis*, 20(3):389–403, 2000.

[PBvHN11] Thomas Nordahl Petersen, Søren Brunak, Gunnar von Heijne, and Henrik Nielsen. Signalp 4.0: discriminating signal peptides from transmembrane regions. *Nature methods*, 8(10):785–786, 2011.

[QSST10] Novi Quadrianto, Alexander J Smola, Le Song, and Tinne Tuytelaars. Kernelized sorting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(10):1809–1821, 2010.

[RAS+13] Juho Rousu, Daniel D Agranoff, Olugbemiro Sodeinde, John Shawe-Taylor, and Delmiro Fernandez-Reyes. Biomarker discovery by sparse canonical correlation analysis of complex clinical phenotypes of tuberculosis and malaria. *PLoS computational biology*, 9(4):e1003018, 2013.

[RBCG08] Alain Rakotomamonjy, Francis R Bach, Stéphane Canu, and Yves Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 9(11), 2008.

[SBB+07] Le Song, Justin Bedo, Karsten M Borgwardt, Arthur Gretton, and Alex Smola. Gene selection via the bahsic family of algorithms. *Bioinformatics*, 23(13):i490–i498, 2007.

[SBE+12] Gurjinder Sandhu, Francesca Battaglia, Barry K Ely, Dimitrios Athanasakis, Rosario Montoya, Teresa Valencia, Robert H Gilman, Carlton A Evans, Jon S Friedland, Delmiro Fernandez-Reyes, et al. Discriminating active from latent tuberculosis in patients presenting to community clinics. *PloS one*, 7(5):e38080, 2012.

[SFG+09] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Gert RG Lanckriet, and Bernhard Schölkopf. Kernel choice and classifiability for rkhs embeddings of probability distributions. In *NIPS*, pages 1750–1758, 2009.

[SGB13] Dino Sejdinovic, Arthur Gretton, and Wicher Bergsma. A kernel test for three-variable interactions. In *Advances in Neural Information Processing Systems*, pages 1124–1132, 2013.

[SH07] Ruslan Salakhutdinov and Geoffrey E Hinton. Using deep belief nets to learn covariance kernels for gaussian processes. In *NIPS*, 2007.

[SRSS06] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *The Journal of Machine Learning Research*, 7:1531–1565, 2006.

[SSG+12] Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *The Journal of Machine Learning Research*, 98888(1):1393–1434, 2012.

[STBWA98] John Shawe-Taylor, Peter L Bartlett, Robert C Williamson, and Martin Anthony. Structural risk minimization over data-dependent hierarchies. *Information Theory, IEEE Transactions on*, 44(5):1926–1940, 1998.

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[TS09] Ryota Tomioka and Masashi Sugiyama. Dual-augmented lagrangian method for efficient sparse reconstruction. *Signal Processing Letters, IEEE*, 16(12):1067–1070, 2009.

[vVDVDV+02] Laura J van't Veer, Hongyue Dai, Marc J Van De Vijver, Yudong D He, Augustinus AM Hart, Mao Mao, Hans L Peterse, Karin van der Kooy, Matthew J Marton, Anke T Witteveen, et al. Gene expression profiling predicts clinical outcome of breast cancer. *nature*, 415(6871):530–536, 2002.

[WMC+00] Jason Weston, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio, and Vladimir Vapnik. Feature selection for svms. In *NIPS*, volume 12, pages 668–674, 2000.

[WRMC12] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012.

[ZH02] Marco Zaffalon and Marcus Hutter. Robust feature selection using distributions of mutual information. In *Proceedings of the 18th International Conference on Uncertainty in Artificial Intelligence (UAI-2002)*, pages 577–584, 2002.

[ZLS+06] Xuegong Zhang, Xin Lu, Qian Shi, Xiu-qin Xu, E Leung Hon-chiu, Lyndsay N Harris, James D Iglehart, Alexander Miron, Jun S Liu, and Wing H Wong. Recursive svm feature selection and sample classification for mass-spectrometry and microarray data. *BMC bioinformatics*, 7(1):197, 2006.

[ZY06] Peng Zhao and Bin Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.