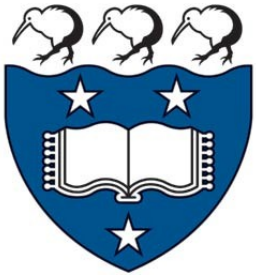
On the application of the stability methods to time series data

Vicky Deng

Supervisor: Dr. Ciprian Doru Giurcăneanu



Bachelor of Science (Honours)
Department of Statistics
The University of Auckland
New Zealand

Abstract

Many researchers have been working on the problem of high-dimensional statistical modelling, where the sample size (n) is much smaller than the number of covariates (p) (see Bühlmann and Geer (2011)). This problem is difficult because the large value of p makes the total number of candidate variables impractically large. A possible approach for selecting variables considers the 'stability' of the selection under subsampling. In Meinshausen and Bühlmann (2010), instead of applying a particular algorithm to the whole data set of size n, the algorithm is applied repeatedly to subsets of size $n/2$, and a variable chosen frequently when running the experiments is deemed to be relevant. A more advanced variant of this procedure for variable selection was introduced in Shah and Samworth (2013). The methods mentioned above were designed for independent and identically distributed (i.i.d) data. However, the method from Shah and Samworth (2013) was recently applied to time series in Bijral (2019) by sampling from the data blocks that are 'almost' independent. In this project, we evaluate the performance of the method from Bijral (2019) on simulated and real-life data, and we also propose some modifications.

Acknowledgements

I would like to express my sincere gratitude to Dr. Ciprian Doru Giurcăneanu for being an extremely supportive supervisor. I am thankful for the trust and encouragement he gives me continuously. His guidance and assistance have made this research project very enjoyable and an invaluable learning experience, which has sparked my interest in pursuing further studies and research. I feel fortunate to work under Dr. Giurcăneanu's supervision, and I am inspired by his dedicated and responsible attitude towards his work.

I would also like to say thank you to my family and friends whose love and care have supported me throughout my honours degree.

Contents

1 Introduction to predictor selection **1**

1.1 Stable predictor selection on i.i.d data 1

1.2 Block Pair Average (BPA) on time series data 3

1.3 Other predictor selection methods 8

1.4 Dissertation organisation 8

2 Main algorithm **9**

2.1 Preliminaries 9

2.2 Lasso 9

2.3 BPA algorithm 13

3 Experiments with simulated data **17**

3.1 Preliminaries 17

3.2 Artificial data 17

3.2.1 Mathematical model for the simulated data 17

3.2.2 Scenario I 20

3.2.3 Scenario II 20

3.3 Parameter settings 21

3.4 Experimental settings 23

3.4.1 A variant of the BPA . . . 23
3.5 Simulated data results for Scenario I . . . 25
3.5.1 BPA under different SNR level . . . 25
3.5.2 BPA under different parameters . . . 29
3.5.3 Comparison between BPA and BPA-m . . . 34
3.6 Simulated data results for Scenario II . . . 36

4 Experiments with real data **38**
4.1 Preliminaries . . . 38
4.2 Air pollution data . . . 38
4.3 BPA experimental settings . . . 39
4.3.1 Mathematical model . . . 39
4.3.2 BPA parameter settings . . . 42
4.3.3 Performance evaluation . . . 43
4.4 Experimental results for air pollution data . . . 43

5 Final remarks **51**

A Stopping rules for Matching Pursuit Algorithm (MPA) **53**

B Supplementary experiment results **55**
B.1 For BPA in Scenario I . . . 56
B.2 For BPA-m in Scenario I . . . 57
B.3 For BPA in Scenario II . . . 60
B.4 For BPA-m in Scenario II . . . 66

List of Figures

1.1 Example of CPSS random selection of independent pairs of subsets A_{2j-1} and A_{2j} in $j = 1, ..., B$ iterations. 3

1.2 Example of BPA O' and O'' blocks in B iterations. 6

1.3 A pair of blockwise sequences constructed from a dependent sequence $\{\mathbf{Z}_t\}$. 7

3.1 $\mathbf{M}$ and $\{\mathbf{A}_i\}_{1\leq i\leq q_u}$ example, where $q_u = 3$. 19

3.2 TPR/FPR for BPA under different SNR levels in Scenario I. 25

3.3 Boxpot for λ_q values, for BPA under different SNR levels in Scenario I (outlier removed). BPA parameters are fixed at $a_T = 100$, $q = 0.4p$, $\phi = 0.8$, $B = 50$. 26

3.4 Illustrations for the range of λ values generated by MATLAB `lasso` function on a data set for each SNR level. 27

3.5 Number of predictors (p^*) before applying and after applying $\phi = 0.8$ for BPA under different SNR levels in Scenario I. 28

3.6 TPR/FPR for BPA under different a_T values in Scenario I. Other parameters are fixed at $q = 0.4$, $\phi = 0.8$, $B = 50$. 29

3.7 TPR/FPR for BPA under different q values in Scenario I. Other parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 50$. 30

3.8 $\lambda_{\mathbf{q}}$ selection and number of final predictors $\mathbf{p}^*$ for BPA under different SNR levels in Scenario I. Parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 50$. . . 31

3.9 TPR/FPR for BPA under different ϕ values in Scenario I. Other parameters are fixed at $a_T = 100$, $q = 0.4$, $B = 50$. 32

3.10 TPR/FPR for BPA under different B values in Scenario I. Other parameters are fixed at $a_T = 100$, $q = 0.4$, $\phi = 0.8$. 33

3.11 TPR/FPR for BPA-m under different SNR levels for Scenario I. 34

3.12 Comparison of TPR/FPR between BPA and BPA-m under different SNR values for Scenario I. Other parameters are fixed at $a_T = 100$, $q = 0.4p$, $\phi = 0.8$, $B = 50$. 35

3.13 TPR/FPR for BPA and BPA-m under different σ^2 levels in Scenario II. . . 37

4.1 Scenario (a) - Full set of predictors (FullSet). 40

4.2 Scenario (b) - Constrained set of predictors (ConSet). 41

4.3 Boxplots for the values of λ_q selected in $N_{TR} = 100$ runs for each scenario of each site. 44

4.4 Range of λ values generated by MATLAB Lasso function 46

4.5 Number of predictors (p^*) before applying and after applying $\phi = 0.8$ for air pollution data. 47

4.6 Boxplots for the number of final predictors selected in $N_{TR} = 100$ runs for each scenario of each site. 49

B.1 λ_q boxplots for different SNR level (outlier included) in Scenario I. Parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 50$. 56

B.2 λ_q boxplots for different q (outlier included) in Scenario I. Parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 50$. 56

B.3 TPR/FPR for BPA under different a_T values. Other parameters are fixed at $q = 0.4$, $\phi = 0.8$, $B = 1$. 57

B.4 TPR/FPR for BPA under different q values. Other parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 1$. 58

B.5 TPR/FPR for BPA under different ϕ values. Other parameters are fixed at $a_T = 100$, $q = 0.4$, $B = 1$. 59

B.6 λ_q selection and number of final predictors for BPA under different noise levels in Scenario II. Parameters are fixed at $a_T = 100$, $q = 0.8$, $\phi = 0.8$. . . 60

B.7 Error bar TPR/FPR for BPA and BPA-m under different σ^2 values in Scenario II. Other parameters are fixed at $a_T = 100$, $q = 0.4p$, $\phi = 0.8$. . . . 61

B.8 TPR/FPR for BPA under different a_T values. Other parameters are fixed at $q = 0.4$, $\phi = 0.8$, $B = 50$. 62

B.9 TPR/FPR for BPA under different q values. Other parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 50$. 63

B.10 TPR/FPR for BPA under different ϕ values. Other parameters are fixed at $a_T = 100$, $q = 0.4$, $B = 50$. 64

B.11 TPR/FPR for BPA under different B values. Other parameters are fixed at $a_T = 100$, $q = 0.4$, $\phi = 0.8$. 65

B.12 TPR/FPR for BPA under different a_T values. Other parameters are fixed at $q = 0.4$, $\phi = 0.8$, $B = 1$. 66

B.13 TPR/FPR for BPA under different q values. Other parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 1$. 67

B.14 TPR/FPR for BPA under different ϕ values. Other parameters are fixed at $a_T = 100$, $q = 0.4$, $B = 1$. 68

List of Tables

2.1 Lasso selection example. 12

2.2 Toy example: Selection of O' and O'' in $B = 10$ iterations. 14

2.3 Toy example: Block average selection estimator. 15

2.4 Toy example: Block average selection estimator continued. 16

3.1 Conversion from ψ to the absolute noise variance σ_n^2 and SNR_{dB}. 20

3.2 Values of a_T and corresponding μ_T and l_T. 21

3.3 Number of possible combinations for each a_T values tested. 23

4.1 Response vector and the matrix of predictors for the air pollution data. . . 40

4.2 NMSE of predictive models for air pollution data 44

4.3 Final predictors ranked by the number of times they get selected in the 100 runs. 50

Chapter 1

Introduction to predictor selection

In this chapter, we explore the feature selection methodologies for dependent data. A well-known sub-sampling based selection framework was developed by Shah and Samworth (2013) and is typically used for independent and identically distributed (i.i.d) data. In this work, our analysis aims to closely examine a time series predictor selection scheme proposed by Bijral (2019) under the sub-sampling based selection framework.

1.1 Stable predictor selection on i.i.d data

High-dimensional statistics refers to the situation when the number of predictors, p, is much larger than the sample size n, i.e., $p >> n$. This situation is called the curse of dimensionality, which means data becomes increasingly sparse when dimensionality increases. Data sparsity makes the density and distance between points, which is critical to supervised regression and classification models, unsupervised clustering, graphical modelling, and multiple testing difficult (Meinshausen and Bühlmann, 2010), become less meaningful. The curse of dimensionality also increases the risk of overfitting and makes it very difficult to identify patterns in the data without having plenty of training data. Overall, high dimensionality makes training extremely slow, and it is harder to find a good solution.

In the case of ordinary least square regression, if $p > n$, then there is no longer a unique coefficient estimate. Also, the variance is infinite, so the method cannot be used at all.

Therefore, dimension reduction is needed to avoid the curse of dimensionality, help eliminate irrelevant features and reduce noise, reduce required time and space, and allow easier visualization.

It is crucial to implement dimension reduction and regularization to address these issues with high-dimensional data. Stability selection is proposed by Meinshausen and Bühlmann (2010) to do regularization under subsampling, and Lasso is used as a base procedure for variable selection. "Stability" means the uncertainties can be quantified for this sub-

Algorithm 1 Complementary Pairs Stability Selection (CPSS)

Let: $(A_{2j-1}, A_{2j}) : j = 1, ..., B$ be randomly chosen independent pairs of subsets of $\{1, ..., n\}$ of size $\lfloor n/2 \rfloor$ such that $A_{2j-1} \cap A_{2j} = \emptyset$.
Define:

$$\hat{\Pi}_B(k) := \frac{1}{2B} \sum_{j=1}^{2B} \mathbb{1}_{k \in \hat{S}(A_j)} \tag{1.1}$$

Complementary pairs selector estimator:

$$\hat{S}^{CPSS}_{n,\tau} := \{k : \hat{\Pi}_B(k) \geq \tau\} \tag{1.2}$$

for some $\tau \in (0, 1]$.

sampling selection algorithm on high-dimensional data. A desirable feature of stability selection is the error control that is provided by an upper bound on the expected number of falsely selected variables (Meinshausen and Bühlmann (2010), Theorem 1). This feature provides a transparent principle for choosing a proper amount of regularization. Under the stability selection framework by Meinshausen and Bühlmann (2010), the more advanced Complementary Pairs Stability Selection (CPSS) is introduced by Shah and Samworth (2013) with improved error control.

Algorithm 1 shows the CPSS algorithm: B is the number of times for random selection of independent pairs of subsets. The operator $\lfloor \cdot \rfloor$ gives the greatest integer less than or equal to the real number in the argument. k is an arbitrary predictor (from the pre-defined set of predictors). $\hat{S}(A_j)$ is the base selection procedure (Lasso in our case), which is executed on the subset A_j for base-level variable selection. $\mathbb{1}$ is the indicator function that maps elements of the subset to one and all other elements of the set to zero, so the predictor k is counted to the total if it is selected for the subset A_j. The symbol τ denotes the top-level variable selection threshold. By setting a high variable selection threshold, the variable selection procedure in Equation (1.2) selects variables that have high selection probability under the base procedure and avoids selecting those variables with low selection probability.

The colour blocks in Figure 1.1 demonstrate how the random selection of independent subsets works in CPSS. The key idea of stability selection is to improve on this simple estimator of $\mathbb{E}(\mathbb{1}_{k \in \hat{S}(A_j)})$ through sub-sampling. By means of the averaging that is involved in Equation (1.1), $\hat{\Pi}_B(k)$ will have reduced variance compared with $\mathbb{1}_{k \in \hat{S}(A_j)}$, and this increased stability will exceed the compensation for the bias incurred (Shah and Samworth, 2013). This averaging technique has been successfully applied in bagging (Breiman,1996 1999) and subagging (Bühlmann and Yu, 2002), such as classification trees (Breiman et al., 1984) and nearest neighbour classifiers (Hall and Samworth, 2005; Biau et al., 2010; Samworth, 2012).

Thus, we can see that this methodological concept shares some common aspects with nonparametric statistics, e.g., bootstrapping and machine learning.

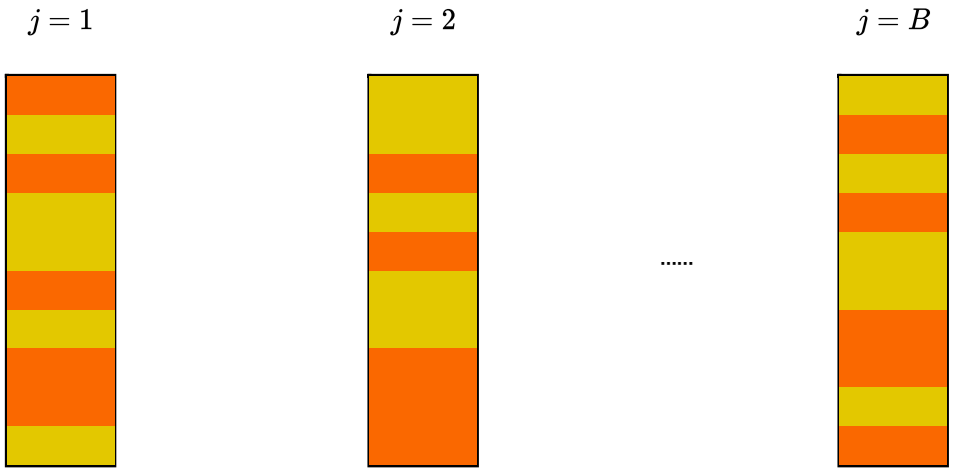


Figure 1.1: Example of CPSS random selection of independent pairs of subsets A_{2j-1} and A_{2j} in $j = 1, ..., B$ iterations. Orange blocks represent randomly selected A_{2j-1} blocks, and yellow blocks represent the residual A_{2j} blocks. This colour blocks plot is inspired by the CPSS plot in Shah and Samworth (2013) presentation.

1.2 Block Pair Average (BPA) on time series data

As mentioned earlier, regularization is crucial to overcoming the problem of $p >> n$. Time series data is often high-dimensional; therefore, performing regression on time series data is also subject to the problem of $p >> n$. However, the stability selection method described in the previous section holds only for i.i.d data. In time series applications, the error control yielded by the stability procedures does not hold as the sub-sampling does not account for the underlying dependence (Bijral, 2019).

Most existing variable selection methods in time series simply use a plain variable selection procedure on the entire data. Three types of variable selection procedures are distinguished in Ng (2013):

- Criterion-based methods, e.g., Akaike Information Criterion (AIC) (Akaike, 1969, 1970), Bayesian Information Criterion (BIC) (Schwarz, 1978).
- Regularization, e.g., Lasso (Tibshirani, 1996), Ridge (Hoerl and Kennard, 1970).
- Dimension reduction procedures, e.g., Principal Components Analysis (PCA) (Pearson, 1901), Canonical Correlation Analysis (CCA) (Hotelling, 1992), Partial Least Squares (PLS)(Wold, 1968), Factor models (Connor and Korajczyk, 1993).

Statisticians are actively studying the variable selection problem in time series for many applications. Some areas are listed below:

- Economics: forecasting with many predictors (Mol et al., 2008) or understanding structural relationships (Christiano et al., 1999).
- Finance: building large scale systemic risk models.
- Functional genomics: reconstructing gene regulatory networks based on limited experimental data.
- Neuroscience: building detailed connectivity maps on temporal data exhibiting multiple structural changes.
- Network control: for large sparse systems (Liu et al., 2011).

These examples suggest the need for a good variable selection method for time series data.

A concept of block pair average (BPA) is introduced by Bijral (2019) which provides quantifiable error control over the base variable selection method Lasso. Bijral's experimental results show that the predictors selected by BPA have superior predictive performance on several data sets over plain Lasso, Adaptive Lasso (Zou, 2006), Elastic Net (Zou and Hastie, 2005), and AIC. A mathematical justification is also provided in Bijral (2019).

BPA is a modified Complementary Pairs Stability Selection (CPSS), see Algorithm 2. To explain how the algorithm works, we assume that the measurements $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T$ are available for a d-dimensional time series. Note that we use bold font for matrices and vectors. The following model is adopted for the time series:

$$\mathbf{y}_t = \mathbf{B}\mathbf{Z}_t + \mathbf{u}_t, \text{ for } t \in \{1, 2, \ldots, T\},$$

where $\mathbf{Z}_t$ is obtained by stacking all the predictors that are deemed to be relevant for $\mathbf{y}_t$ and $\mathbf{u}_t$ denotes the zero-mean noise. In general, it is assumed that the distribution of the noise is Gaussian, and the covariance matrix is unknown. We emphasize that all entries of $\mathbf{Z}_t$ are observed, and the sequence $\{\mathbf{Z}_t\}$ is supposed to be stationary. The main task is to estimate the entries of the matrix $\mathbf{B}$. At the same time, it is also necessary to select the predictors, or equivalently, to find the zeros of the $\mathbf{B}$-matrix.

By applying BPA, we construct B times the side-by-side blockwise sequence via sampling from the dependent sequence $\{\mathbf{Z}_t\}$ to aggregate the results of Lasso regularization on those subsamples of the data (Shah and Samworth, 2013), see Figure 1.2. The procedure is described below. First, divide the sequence $\{\mathbf{Z}_t\}$ (see Figure 1.3) into $4l_t$ blocks of length a_T and assume that $T = 2\mu_T a_T$ and $l_t = \mu_t/2$. There are therefore $\mu_T = 2l_T$ odd blocks denoted by $O = \{\mathbf{O}_1, ..., \mathbf{O}_{2l_T}\}$ and $\mu_T = 2l_T$ even blocks denoted by $E = \{\mathbf{E}_1, ..., \mathbf{E}_{2l_T}\}$. Then, by randomly sampling half of the O blocks so that there are l_T odd blocks in a set O', we construct a dependent matrix $\mathbf{Z}'$ stacked by O' blocks (see again Figure 1.3). For easier understanding, we use O' and O'' in this work to denote the pairs of the blockwise sequence for (O_{2j-1}, O_{2j}) in each of $j = 1, ..., B$ (see Algorithm 2). After that, the side-by-side blockwise matrix $\mathbf{Z}''$ is constructed by stacking the complementary l_T number of O'' blocks. The points in a block are dependent in $\mathbf{Z}''$, but the blocks are independent. The odd blocks in $\mathbf{Z}'$ and $\mathbf{Z}''$ have the same distribution (Bijral, 2019). The intuition behind this approach is that, for an appropriately mixing sequence via random re-sampling, the odd blocks are roughly independent, provided the block length a_T is large enough to create

Algorithm 2 Block Pair Average (BPA) Stability Selection

Let: $(O_{2j-1}, O_{2j}) : j = 1, ..., B$ be the set of randomly selected pairs of sequence of blocks such that $O_{2j-1} \cap O_{2j} = \emptyset$.
Define:

$$\Pi_B^{av}(k) := \frac{1}{2B} \sum_{j=1}^{2B} \mathbb{1}_{k \in \hat{S}_{|O_j|}}$$

Select block average selector estimator:

$$\hat{S}^{av} := \{k : \Pi_B^{av}(k) \geq \phi\}$$

for some $\phi \in (0, 1]$.

sufficient gaps between adjacent odd blocks. Suppose we create "independent" blocks in $\mathbf{Z}''$ with each new block having the same distribution as its dependent counterpart in $\mathbf{Z}'$. In that case, we can use this duality to work with the original dependent time series $\{\mathbf{Z}_t\}$ (Bijral, 2019).

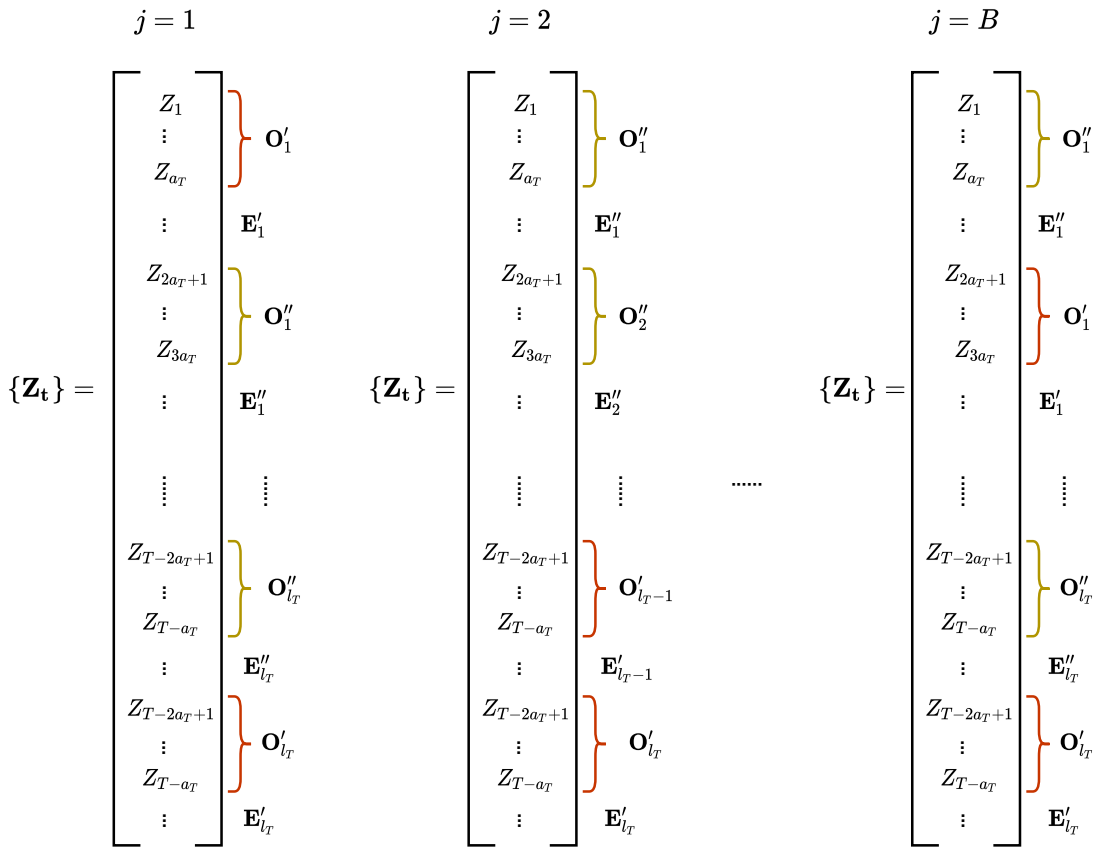


Figure 1.2: Example of BPA O' and O'' blocks in B iterations. The colouring of the blocks corresponds to the CPSS random pairs selection in Figure 1.1. In this plot, orange blocks represent the randomly selected O' blocks, $O' = \cup_{j=1}^{l_T}\{\mathbf{O}'_j\}$, and yellow blocks represent the complement O'' blocks, $O'' = \cup_{j=1}^{l_T}\{\mathbf{O}''_j\}$.

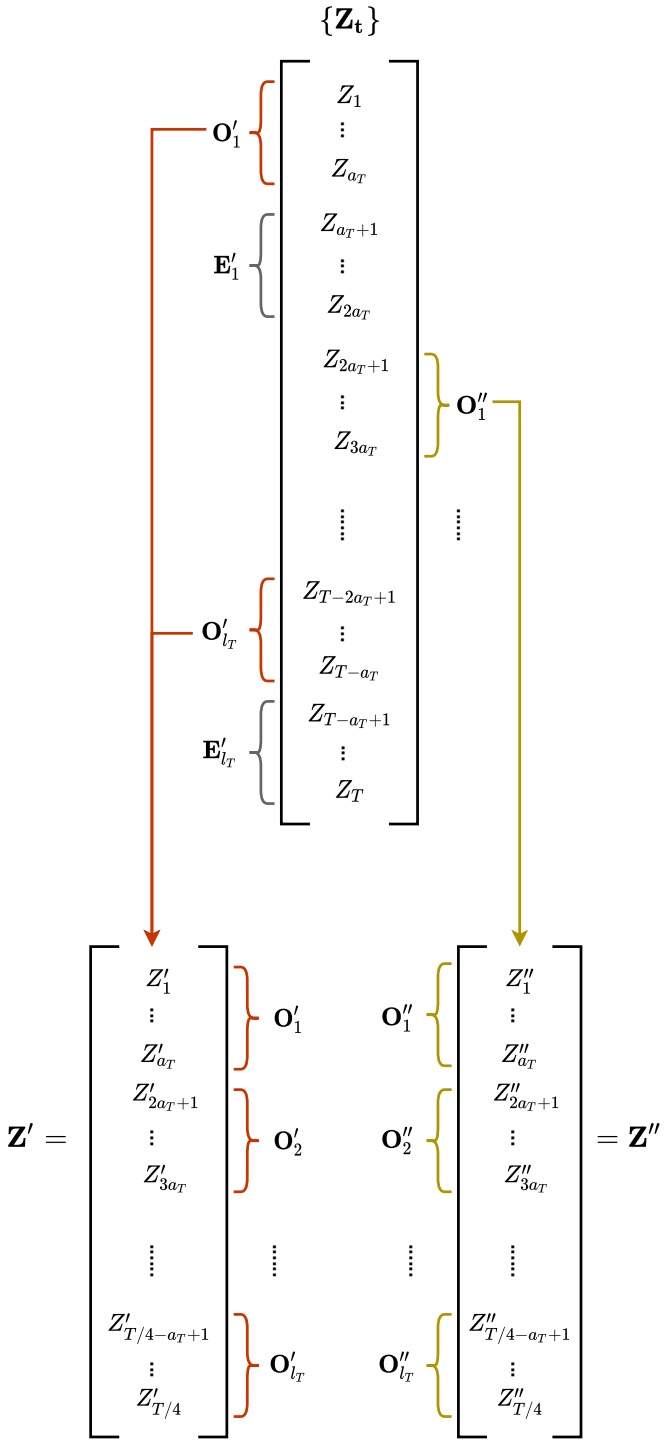


Figure 1.3: A pair of blockwise sequences constructed from a dependent sequence $\{\mathbf{Z}_t\}$. The random selection of blocks corresponds to the block selection in $j = 1$ in Figure 1.2. Orange blocks represent the randomly selected O' blocks, $O' = \cup_{j=1}^{l_T}\{\mathbf{O}'_j\}$, and yellow blocks represent the complement O'' blocks, $O'' = \cup_{j=1}^{l_T}\{\mathbf{O}''_j\}$. The grey blocks are even blocks which are discarded in the construction of $\mathbf{Z}'$ and $\mathbf{Z}''$. This figure is inspired by Figure 1 in Bijral (2019).

1.3 Other predictor selection methods

In this section, we list some other methods that have been used for selecting the predictors in time series data. We do not discuss the technical details of these methods.

- Bayesian penalized regression

  A tutorial on such methods as well as their MATLAB and R implementations can be found in Makalic and Schmidt (2016). Other methods include: Bayesian variant by Park and Casella (2008) of the well-known lasso algorithm (Tibshirani, 1996) and Horseshoe by Carvalho et al. (2010).

- Greedy algorithms

  The Matching Pursuit Algorithm (MPA) is used in many applications for selecting the best predictors for high-dimensional data; it has lower computational complexity than Bayesian penalized regression. The various stopping rules for the matching pursuit algorithm are examined in Li et al. (2019).

- CART algorithm

  An early and important book about statistics for complex data is Breiman et al. (1984) with a strong emphasis placed on the CART algorithm.

- Machine learning techniques

  The influential book by Hastie et al. (2009) covers an extensive range of methods and techniques at the interface between statistics and machine learning, also called "statistical learning" and "data mining". Other methods include Kernel methods from machine learning by Hofmann et al. (2008).

Overall, there is no single method that can substantially outperform the others. There are still some difficulties in finding the predictors for time series data.

1.4 Dissertation organisation

This study on the application of BPA to time series data is divided into four parts. A brief review of the previous research was already discussed in this chapter. The BPA algorithm is analysed in detail in Chapter 2. To accomplish the goal of this work, we analyse the applicability of BPA to simulated and real time series data sets and arrive at a conclusion about whether the scheme can accommodate the statistical dependence of time series data. The investigation of the impact of different noise levels and parameter values on BPA's performance and a variant of BPA can be found in Chapter 3. Finally, BPA's performance in real data experiments compared to greedy algorithms is in Chapter 4. The experimental settings and empirical results are included in Chapters 3 and 4. Chapter 5 presents the conclusions and implications drawn from our results which can be explored further in the future work. The author's contribution is summarised at the end of this work.

Chapter 2

Main algorithm

2.1 Preliminaries

As we already pointed out in Chapter 1, the problem that we want to solve is: for lots of predictors in $\mathbf{Z_1}\ \mathbf{Z}_2\ \cdots\ \mathbf{Z}_T$, which only a few are relevant, we want to find a solution that only selects those few useful predictors. Our interest is to understand the BPA stability method proposed by Bijral (2019) that is designed to solve such problems. Below we present the components of the main algorithm.

There are T observations in a data set and p predictors.

Data set is divided into blocks of length a_T (see Figures 1.2 and 1.3).

Therefore, there are $T/2a_T = \mu_T$ odd blocks and even blocks respectively, and $l_t = \mu_t/2$ O' and O'' blocks respectively.

There are q active entries in each column of the matrix of regularized predictor coefficients $\mathbf{B}$.

The variable selection threshold is ϕ.

The number of iterations for random selection of pairwise blocks sequence is B.

2.2 Lasso

Lasso is the base procedure which we use in BPA stability selection. It is well-known that Lasso (Tibshirani, 1996) is a regularization method used in penalized regression to solve high dimensional problems. The penalty term is called Lasso-type penalty, or ℓ_1-form penalty. The goal of the algorithm is to minimize the sum of squares with constraint $\lambda\|\mathbf{B}\|_1$, i.e.,

minimise Residual Sum of Squares +
$\lambda *$ (sum of the absolute value of the magnitude of coefficients),

presented as below:

$$\min_{\mathbf{B}} \|\mathbf{Y} - \mathbf{BZ}\| + \lambda \|\mathbf{B}\|_1 \tag{2.1}$$

where bias increases as λ increases; variance increases as λ decreases. The variance-bias trade-off here means that we will need to tune the value for λ to find a relatively low bias and low variance model. The value for λ is controlled by the parameter q in BPA, and the λ selected by q is represented as λ_q. The value of q is also explored by empirical experiments later in this work.

Other base procedures that can be used are the AIC-type penalty (ℓ_0-form penalty), Ridge-type penalty (ℓ_2-form penalty), Elastic Net and Adaptive Lasso.

Ridge regression introduced by Hoerl and Kennard (1970) is robust against multicollinearity (Duzan and Shariff, 2015). It does proportional shrinkage, which means more shrinkage on low information. It is slower than Lasso. The geometric interpretation of Ridge regression gives a circular disk region, and the Residual Sum of Squares (RSS) is unlikely to meet the axis directly. Therefore, Ridge regression is unlikely to produce a zero coefficient. Ridge is a good default regularization method, but if only a few features are suspected to be useful, Lasso or Elastic Net is preferred.

Lasso smooths the penalty term by using soft thresholding, which means as the smoothing parameter is varied, the sample path of the estimates moves continuously to zero (Tibshirani, 1996). Its constraint is a convex set, so it is easier to fit with Lasso than with AIC-type penalties. Note that AIC is the acronym for the well-known Akaike Information Criterion. Unlike Ridge regression, the geometric interpretation of Lasso is that the constraints give "corners" on the x-axis and y-axis; more predictors will give more corners. And if the RSS meets one of the corners, then it will give us a zero coefficient (since it's on the x-axis or y-axis), i.e., shrinkage, and thus achieve the goal of variable selection. This type of regularization can result in sparse models with few coefficients, which makes the Lasso far more straightforward to interpret than the Ridge regression (Muthukrishnan and Rohini, 2016).

However, Lasso behaves erratically when the number of features exceeds the number of training instances or when several features are strongly correlated. So Lasso is not a very satisfactory variable selection method in the $p >> n$ case. Elastic Net was proposed by Zou and Hastie (2005), which is a particularly useful regularization and variable selection method when the number of predictors (p) is much larger than the number of observations (n).

Another version of Lasso is Adaptive Lasso (Zou, 2006), where adaptive weights are used for penalizing different coefficients in the ℓ_1 penalty. By such, it avoids overfitting penalizing the large coefficients. Besides, it has the same advantage as Lasso: it can shrink some of the coefficients to exactly zero, performing variable selection.

In our study, we choose to use Lasso to identify and remove the redundant predictors,

following the procedure in Bijral (2019). We use the following notations for Lasso regularization. $\mathbf{B}_\Lambda$ is a p-by-L matrix, where p is the number of predictors, and L is the number of Λ values. Each column of $\mathbf{B}_\Lambda$ corresponds to a particular regularization coefficient in Λ. Λ parameters are in ascending order from left to right column. As the Λ parameter increases, the Lasso-type penalty increases, i.e., the regularization is harsher, which will leave fewer active variables.

We use the built-in `lasso` function in MATLAB to select the range of Λ values. We set all the optional inputs as default so that the maximum value of Λ is the largest λ value that gives a nonnull model, and the minimum value of Λ is set by the ratio of the smallest to the largest λ values, which is $1e^{-4}$. The length of the Λ vector is by default 100. However, the `lasso` function can return fewer than 100 values if the residual error of the fits drops below a threshold fraction of the variance of the response data $\mathbf{Y}$.

We want to find the minimum λ from the Λ vector generated by the MATLAB `lasso` function that gives the desired q (number of active entries of $\mathbf{B}$). See the example in Table 2.1, which is a 20-by-100 matrix, meaning there are 20 predictors in $\mathbf{B}$ and 100 λ values. The columns with the same q active entries are folded, and only the column with the smallest λ value out of those columns is shown, i.e., the table only shows the smallest λ value column for each distinct q active entries. As λ value increases from left to right columns, q decreases. As we set our $q = 0.4p = 0.4 * 20 = 8$, we found $\lambda_{62} = 0.00559$ gives the desired q active entries, see bold column in Table 2.1.

Note that the Lasso regression model and its sparse estimation only serve as an instrument for predictor selection in BPA, as described in Bijral (2019). This means the Lasso regression model is not our final model for predictor selection. We apply Lasso in each block of data and aggregate the results from those blocks for many iterations. As the base procedure, Lasso is only the first layer of predictor selection. The predictors need to be selected a number of times above a particular threshold ϕ to be chosen as our final predictors. We refer to this control by ϕ as the second layer of predictor selection. We will discuss more the BPA algorithm in the next section.

	λ index																
	1	27	30	39	47	54	55	59	60	**62**	72	74	75	97	98	99	100
	λ value																
Predictor No.	1.92E-05	2.15E-04	2.85E-04	6.58E-04	0.00138	0.00265	0.00291	0.00423	0.00464	**0.00559**	0.01417	0.01707	0.01873	0.14501	0.15915	0.17467	0.1917
1	0.038	0.035	0.035	0.031	0.026	0.015	0.013	0.003	0.000	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.016	0.014	0.014	0.011	0.007	0.001	0.000	0.000	0.000	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.020	0.017	0.016	0.011	0.000	0.000	0.000	0.000	0.000	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	-0.221	-0.219	-0.219	-0.218	-0.216	-0.213	-0.212	-0.210	-0.209	**-0.207**	-0.195	-0.188	-0.186	-0.019	0.000	0.000	0.000
5	0.081	0.081	0.080	0.079	0.076	0.075	0.075	0.075	0.075	**0.076**	0.083	0.084	0.084	0.054	0.049	0.020	0.000
6	0.017	0.016	0.015	0.013	0.011	0.006	0.005	0.000	0.000	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7	-0.015	-0.014	-0.014	-0.013	-0.010	0.000	0.000	0.000	0.000	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8	0.012	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
9	-0.085	-0.084	-0.083	-0.080	-0.074	-0.063	-0.062	-0.054	-0.051	**-0.047**	-0.010	0.000	0.000	0.000	0.000	0.000	0.000
10	0.014	0.013	0.013	0.010	0.006	0.000	0.000	0.000	0.000	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
11	0.005	0.004	0.003	0.000	0.000	0.000	0.000	0.000	0.000	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12	-0.005	-0.002	-0.002	-0.002	-0.004	-0.007	-0.009	-0.011	-0.011	**-0.010**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
13	0.017	0.014	0.014	0.010	0.006	0.004	0.003	0.002	0.001	**0.001**	0.002	0.001	0.000	0.000	0.000	0.000	0.000
14	-0.044	-0.045	-0.044	-0.041	-0.035	-0.023	-0.021	-0.008	-0.005	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
15	-0.016	-0.016	-0.016	-0.015	-0.013	-0.011	-0.010	-0.002	0.000	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
16	-0.006	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
17	0.975	0.973	0.972	0.968	0.960	0.947	0.944	0.931	0.927	**0.924**	0.904	0.896	0.888	0.222	0.145	0.081	0.000
18	1.000	0.999	0.998	0.996	0.991	0.980	0.978	0.966	0.962	**0.956**	0.902	0.883	0.872	0.051	0.000	0.000	0.000
19	0.939	0.937	0.936	0.933	0.927	0.917	0.915	0.905	0.902	**0.896**	0.840	0.820	0.809	0.000	0.000	0.000	0.000
20	-0.044	-0.042	-0.041	-0.038	-0.031	-0.020	-0.018	-0.007	-0.003	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Active entries *q*	20	19	18	17	16	15	13	12	10	**8**	7	6	5	4	3	2	0

Table 2.1: Lasso selection example. This is a toy example with 20 predictors, from simulated data for which $T = 10^3$. Green represents active entries (variable selected), red represents inactive entries (variable not selected). In this example where $q = 0.4p = 8$, by running the experiment, at λ with index 62 the number of active entries q is 8. Therefore, we found the smallest λ that returns at most $q = 8$ active entries is $\lambda_{62} = 0.00559$.

Algorithm 3 BPA (with the base procedure Lasso)

Input: $\mathbf{Y}$, $\mathbf{Z}$ (T measurements, p predictors),
$a_T \in \{5, 50, 100, 250, 500\}, q \in \{0.2p, 0.4p, 0.6p\}, \phi \in \{0.5, 0.6, 0.7, 0.8, 0.9\}, B = 50,$
$\Lambda = \{\lambda_1, ..., \lambda_{100}\}$ a sequence of regularizers
Initialize: $\Pi_B^{av}(k) = 0, \forall k \in \{1, ..., p\}$
q-Estimates: Solve

$$\min_{\mathbf{B}} \|\mathbf{Y} - \mathbf{BZ}\| + \lambda \|\mathbf{B}\|_1$$

and set $\lambda_q \in \Lambda$ to be the smallest λ that returns q active entries of $\mathbf{B}$.
for $n = 1$ to B **do**
Sample: Sequence of blocks $O' = \{\mathbf{O}'_1, ..., \mathbf{O}'_{\mu_{T/2}}\}$ from the set of $\mu_T = \dfrac{T}{2a_T}$ odd blocks (O) without replacement and set $O'' = O \backslash O'$. At each run, a different pair of sets $\{O', O''\}$ is generated.
Set: For $\{O', O''\}$,

$$\hat{S}_{|O'|} = \{i : \hat{\mathbf{B}}_i \neq 0, \hat{\mathbf{B}} = \underset{\mathbf{B}}{\operatorname{argmin}} \|\mathbf{Y}(O') - \mathbf{BZ}(O')\|^2 + \lambda_q \|\mathbf{B}\|_1\}$$

$$\hat{S}_{|O''|} = \{i : \hat{\mathbf{B}}_i \neq 0, \hat{\mathbf{B}} = \underset{\mathbf{B}}{\operatorname{argmin}} \|\mathbf{Y}(O'') - \mathbf{BZ}(O'')\|^2 + \lambda_q \|\mathbf{B}\|_1\}$$

.
BPA: $\Pi_B^{av}(k) = \Pi_B^{av}(k) + \mathbb{1}_{k \in \hat{S}_{|O'|}}/(2B) + \mathbb{1}_{k \in \hat{S}_{|O''|}}/(2B), \forall k \in \{1, ..., p\}.$
end for
Output: $\hat{S}^{av} = \{k : \Pi_B^{av}(k) \geq \phi\}$

2.3 BPA algorithm

Algorithm 3 shows the use of the BPA in the context of the Lasso base predictor.

There are two layers of predictor selection. The first bound is on the number of variables included by the Lasso base procedure (controlled by q), which may allow variables with low selection probability to pass through. The second bound is on the proportion of subsamples for which a predictor must be selected (controlled by ϕ) for it to be declared significant. The two bounds can be further tightened to yield improved error control, therefore increasing the applicability of the methodology (Shah and Samworth, 2013). The tightening of the bounds is explored in the later chapter via changing the BPA parameter values.

In our toy example (see again Table 2.1), we randomly divide O blocks into complementary O' and O'' for 10 iterations ($B = 10$). We want to ensure that each selection round is different from the other. There are $\binom{10}{5} = 252$ different possible selections for O' blocks, so we can definitely be able to get ten different possible selections of O' blocks. Table 2.2 is an example of a random selection of O' and O'' blocks for ten iterations. Recall a graphical representation of the random block selection was in Figure 1.2. Note the numbering and selection of the blocks are based on the relative position in O.

	O' blocks no.					O'' blocks no.				
n=1	1	2	3	7	9	4	5	6	8	10
n=2	1	2	6	9	10	3	4	5	7	8
n=3	2	3	4	5	8	1	6	7	9	10
n=4	2	6	8	9	10	1	3	4	5	7
n=5	2	5	6	7	10	1	3	4	8	9
n=6	1	2	3	5	10	4	6	7	8	9
n=7	1	4	5	7	9	2	3	6	8	10
n=8	2	4	5	7	8	1	3	6	9	10
n=9	2	4	5	7	10	1	3	6	8	9
n=10	1	2	3	4	8	5	6	7	9	10

Table 2.2: Toy example: Selection of O' and O'' in $B = 10$ iterations. We continue to use the toy example from Table 2.1, assume $T = 10^3$ and $a_T = 50$. Therefore, there are $\mu_T = T/2a_T = 10$ O blocks with $l_T = \mu_T/2 = 5$ O' blocks and O'' blocks each.

The toy example is continued in Table 2.3, where we demonstrate how the BPA selection estimator works. There are two 20-by-5 matrices, one for the O' block average selection, $\hat{S}_{|O'|}$, and the other for the O'' block average selection, $\hat{S}_{|O''|}$. Each block's regularized coefficients by λ_q is a single column in the $\hat{S}_{|O'|}$ matrix. We use λ_q=0.00559, which is the smallest λ that returns $q = 0.4p = 8$ active entries for the entire data set (see again Table 2.1), to select predictors in subsamples O' and O''.

The number of times for each predictor being selected by O' and O'' are counted separately. The block average estimator is the probability of each predictor being selected in O' and O'' for B iterations. Since we run $B = 10$ times, we have $2B = 20$ of O' and O'' combined. Therefore, the count of times for each predictor being selected by one of O' and O'' is divided by 20.

In summary, to calculate the average for B runs, sum all the selections for each predictor divided by $2B$ for O', and do the same for O'', see the last column in Table 2.3. Then sum the two averages together for each predictor. This gives the block average estimate $\Pi_B^{av}(k)$. This average cumulative probability is calculated for each predictor in the second column in Table 2.4. If we set the value of ϕ to 0.8, then the predictors selected in the toy example are given in the third column of Table 2.4. Lastly, the predictors selected for this toy example are compared to the true predictors to calculate the true positive rate (TPR) and false positive rate (FPR).

$\hat{S}_{|O'|}$

Predictor No.	Number of iterations										No. of times predictor selected	Count. / 20
	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10		
1	-0.012	0.030	0.000	0.043	0.014	0.000	0.000	0.000	0.000	0.000	4	0.2
2	0.000	0.010	0.000	0.000	0.004	0.063	-0.013	-0.059	0.000	0.000	5	0.25
3	0.000	0.000	-0.007	0.000	-0.004	-0.021	0.000	-0.018	-0.020	0.000	5	0.25
4	-0.094	-0.115	-0.161	-0.089	-0.119	-0.182	-0.179	-0.163	-0.179	-0.120	10	0.5
5	0.023	0.065	0.058	0.081	0.054	0.037	0.040	0.033	0.038	0.033	10	0.5
6	0.000	0.000	0.000	0.000	0.000	0.000	0.029	0.035	0.026	0.000	3	0.15
7	-0.009	-0.069	0.000	-0.020	-0.002	0.000	0.000	0.000	0.000	0.000	4	0.2
8	-0.027	-0.029	-0.019	0.000	0.000	-0.013	0.000	0.000	-0.010	-0.114	6	0.3
9	-0.035	-0.038	-0.009	-0.014	-0.072	-0.030	-0.053	-0.043	-0.074	0.000	9	0.45
10	-0.013	-0.075	0.005	-0.035	0.000	-0.050	0.059	0.060	0.039	0.000	8	0.4
11	0.000	0.000	-0.012	0.000	0.000	-0.001	-0.008	0.000	-0.012	0.000	4	0.2
12	-0.101	-0.056	-0.025	-0.098	-0.076	-0.023	-0.012	-0.038	0.000	0.000	8	0.4
13	0.000	0.000	0.000	0.005	0.000	0.000	0.000	0.000	0.000	0.018	2	0.1
14	-0.024	-0.044	-0.030	0.000	0.000	-0.063	0.000	0.000	0.000	-0.060	5	0.25
15	-0.040	-0.021	0.000	0.000	-0.018	0.000	-0.125	-0.008	-0.090	0.000	6	0.3
16	-0.028	0.000	-0.018	-0.004	-0.015	-0.016	-0.013	-0.003	-0.003	0.000	9	0.45
17	1.077	1.115	0.908	1.085	0.876	0.938	0.974	0.899	0.866	1.069	10	0.5
18	0.850	0.864	0.967	0.969	1.040	0.903	0.910	1.009	0.978	0.841	10	0.5
19	0.783	0.800	0.987	0.867	0.864	0.871	0.882	0.973	0.915	0.935	10	0.5
20	-0.058	-0.050	0.050	-0.031	0.014	0.000	0.000	0.050	0.057	0.017	8	0.4

$\hat{S}_{|O''|}$

Predictor No.	Number of iterations										No. of times predictor selected	Count. / 20
	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10		
1	0.039	0.000	0.000	-0.003	0.000	0.000	0.039	0.018	0.011	0.000	5	0.25
2	0.000	0.000	0.000	0.000	0.000	-0.055	0.011	0.048	0.006	0.000	4	0.2
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0	0
4	-0.149	-0.140	-0.106	-0.183	-0.152	-0.080	-0.040	-0.104	-0.066	-0.124	10	0.5
5	0.099	0.081	0.067	0.043	0.060	0.082	0.095	0.091	0.067	0.079	10	0.5
6	0.010	0.005	0.000	0.006	0.000	0.000	0.000	0.000	0.000	0.000	3	0.15
7	0.000	0.000	-0.032	0.000	0.000	-0.021	0.000	-0.035	-0.013	-0.009	5	0.25
8	-0.002	-0.003	-0.020	-0.033	-0.069	0.000	-0.061	-0.051	-0.010	0.000	8	0.4
9	-0.057	-0.050	-0.059	-0.056	0.000	-0.047	-0.016	-0.029	0.000	-0.071	8	0.4
10	0.000	0.055	-0.056	0.000	0.000	0.041	-0.077	-0.083	-0.062	0.000	7	0.35
11	0.000	-0.005	0.000	-0.029	0.000	0.000	0.000	0.000	0.000	0.000	2	0.1
12	-0.014	-0.043	-0.074	-0.007	-0.011	-0.077	-0.084	-0.050	-0.116	-0.076	10	0.5
13	0.000	0.004	0.000	0.000	0.005	0.002	0.022	0.000	0.000	0.000	4	0.2
14	0.000	0.000	0.000	-0.022	-0.073	0.000	0.000	-0.051	-0.019	0.000	4	0.2
15	0.000	0.000	-0.063	-0.074	-0.027	-0.059	0.000	-0.029	0.000	-0.063	6	0.3
16	-0.010	-0.048	-0.011	-0.033	0.000	-0.027	-0.025	-0.022	-0.045	-0.022	10	0.5
17	0.842	0.841	1.033	0.882	1.099	0.973	0.979	1.067	1.084	0.895	10	0.5
18	1.017	0.986	0.890	0.880	0.780	0.948	0.997	0.813	0.858	1.009	10	0.5
19	1.011	1.009	0.802	0.917	0.887	0.933	0.921	0.793	0.852	0.846	10	0.5
20	0.047	0.027	-0.047	0.026	-0.004	0.000	0.000	-0.054	-0.044	0.000	7	0.35

Table 2.3: Toy example: Block average selection estimator. Green represents active entries, red represents inactive entries. Each column in the "Number of iterations" section is one iteration, and each column represents $\hat{S}_{|O'|}$ (top table) and $\hat{S}_{|O''|}$ (bottom table) for that particular iteration.

Predictor No.	**Sum Count. /20**	**Predictors selected by BPA estimator (>=0.8)**	**True Predictors**
1	0.45		✓
2	0.45		
3	0.25		
4	1	✓	✓
5	1	✓	✓
6	0.3		
7	0.45		
8	0.7		✓
9	0.85	✓	✓
10	0.75		
11	0.3		
12	0.9	✓	✓
13	0.3		
14	0.45		
15	0.6		
16	0.95	✓	
17	1	✓	✓
18	1	✓	✓
19	1	✓	✓
20	0.75		
		TPR=	**0.7778**
		FPR=	**0.0909**

Table 2.4: Toy example: Block average selection estimator continued. Ticks represent the predictors selected.

Chapter 3

Experiments with simulated data

3.1 Preliminaries

In this chapter, we analyse the applicability of the BPA stability method on simulated time series data sets. We also want to explore the effect of changing noise levels and parameters on the performance of the BPA stability method. The relationships between the different noise levels and parameters, and their corresponding true positive rate (TPR) and false positive rate (FPR) are examined based on the experimental results.

3.2 Artificial data

3.2.1 Mathematical model for the simulated data

We simulated the data following the definition of VAR model (vector autoregressive model) in Lütkepohl (2005). For $t \in \mathbb{Z}$, we consider the VAR model with exogenous variables:

$$\mathbf{u}(t) = \sum_{i=1}^{q_u} \mathbf{A}_i \mathbf{u}(t-i) + \mathbf{C}\mathbf{v}(t) + \mathbf{w}(t), \tag{3.1}$$

where $\mathbf{u}(t) \in \mathbb{R}^{K_u \times 1}$, $\mathbf{A}_i \in \mathbb{R}^{K_u \times K_u}$ for $i \in \{1, ..., q_u\}$ are coefficient matrices, $\mathbf{C} \in \mathbb{R}^{K_u \times p_v}$ with $\mathbf{v}(t) \in \mathbb{R}^{p_v \times 1}$ produces a random vector allowing for the possibility of exogenous variables at different lags, $\mathbf{w}(t) \in \mathbb{R}^{K_u \times 1}$ is a K_u-dimensional Gaussian white noise or innovation process.

In our settings, the number of time series is $K_u = 4$ and the AR order is $q_u = 3$. All entries in $\{\mathbf{A}_i\}_{1 \leq i \leq q_u}$ are independent outcomes from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, where $\sigma = 0.25$. In other words, σ is the standard deviation for the entries of the matrix

coefficients. Then we define a $K_u \times K_u$ mask matrix $\mathbf{M}$ with all entries equal to one. Some entries of $\mathbf{M}$ on the off-diagonal locations are randomly forced to be zero, and the entries on the main diagonal are always 1: $(\mathbf{A}_i)_{aa} = 1$ for $i \in \{1, ..., q_u\}$ and $a \in \{1, ..., K_u\}$. The number of zero entries is restricted to be below 60% of the K_u^2 entries. The element-wise product of the entries in $\{\mathbf{A}_i\}_{1 \leq i \leq q_u}$ and $\mathbf{M}$ will give the final $(\mathbf{A}_i)_{ab}$, where for any $a, b \in \{1, ..., K_u\}$, we have that the entry indexed by (a, b) is either zero or non-zero for all the matrix coefficients. All the models that we generate are **stable**. This refers to the stability condition from Lütkepohl (2005): A VAR(q_u) process (3.1) is stable if its reverse characteristic polynomial has no roots in and on the complex unit circle. The condition is equivalent to

$$\det(\mathbf{I}_{K_u} - \mathbf{A}_1 z - \cdots - \mathbf{A}_{q_u} z^{q_u}) \neq 0, \quad \text{for } |z| \leq 1, \tag{3.2}$$

where $\det(\cdot)$ denotes the determinant and $\mathbf{I}_{K_u}$ is the identity matrix of size $K_u \times K_u$.

Figure 3.1 is an example of $\mathbf{M}$ and $\{\mathbf{A}_i\}_{1 \leq i \leq q_u}$ for which the stability condition is satisfied.

The matrix $\mathbf{C}$ contains the coefficients for exogenous variables. It has only 3 non-zero entries: $c_{11} = c_{12} = c_{13} = 1$. All the entries of $\mathbf{v}(t)$ are independent and they have zero-mean.

The first entry of $\mathbf{v}(t)$ is simulated from an univariate Gaussian AR(1)-process $\{X_t\}$ defined as

$$X_t = -0.9X_{t-1} + \epsilon_t, \tag{3.3}$$

where the autocorrelation function equals $\rho(\tau) = (-0.9)^{|\tau|}$ for all lags $\tau \in \mathbb{Z}$.

Similarly, the second entry is also produced by an univariate Gaussian AR(1)-process $\{Y_t\}$ defined as

$$Y_t = -0.5Y_{t-1} + \epsilon_t, \tag{3.4}$$

where the autocorrelation function equals $\rho(\tau) = (-0.5)^{|\tau|}$ for all lags $\tau \in \mathbb{Z}$.

The first two entries of $\mathbf{v}(t)$ are normalized such that their variance is σ^2. All other entries of $\mathbf{v}(t)$ are drawn from $\mathcal{N}(0, \sigma^2)$, where $\sigma = 0.25$. In other words, σ here is the standard deviation of exogenous variables and is the same as the standard deviation for the entries of the matrix coefficients.

The vectors $\{\mathbf{w}(t)\}$ are independent and identically distributed, and they are drawn from a K_u-variate Gaussian distribution with zero mean vector and covariance matrix $(\psi\sigma)^2\mathbf{I}$, where $\psi \in \{0.1, 1, 10\}$ and $\sigma = 0.25$. So the variance of the noise is set to $\sigma_n^2 = (\psi\sigma)^2$. Therefore, $1/\psi^2 = \sigma^2/\sigma_n^2$ acts as the signal-to-noise ratio (SNR). In general, SNR is defined as the ratio of signal power to noise power, often expressed in decibels. The decibel (symbol: dB) is a relative unit of measurement equal to one-tenth of a bel (B). The standard definition for SNR is

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log_{10} \frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}, \tag{3.5}$$

where P is average power. In our case,

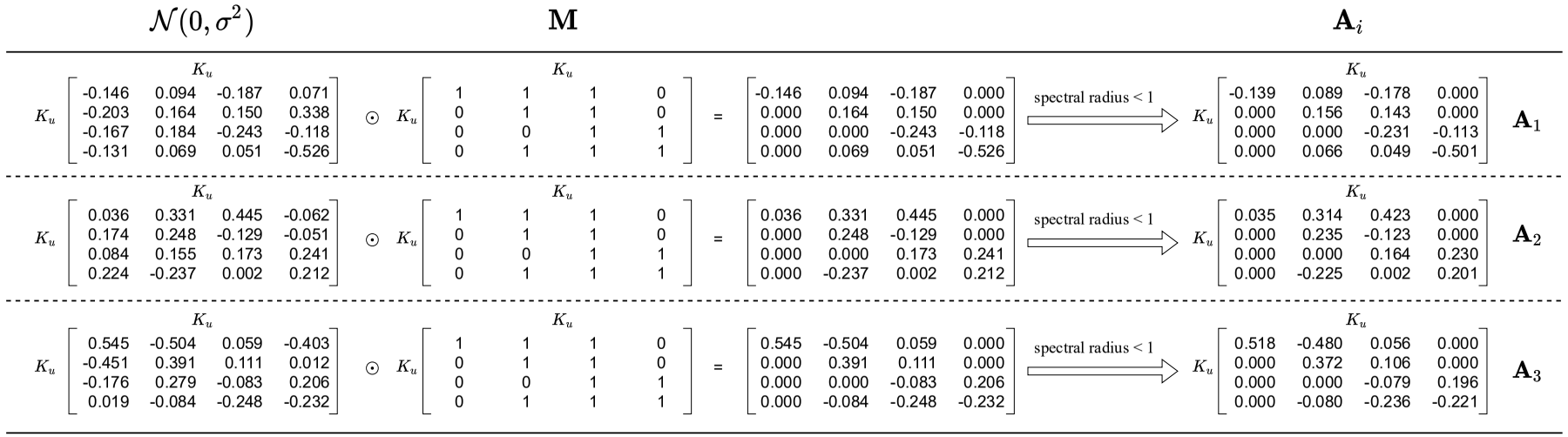


Figure 3.1: $\mathbf{M}$ and $\{\mathbf{A}_i\}_{1 \leq i \leq q_u}$ example, where $q_u = 3$. The operator $\odot$ is for element-wise product of two matrices of the same dimensions and produces another matrix of the same dimension as the operands. The spectral radius is related to be stability condition. The final $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$ are the matrix coefficients of a stable VAR(3) process.

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \frac{\sigma^2}{\sigma_n^2}. \tag{3.6}$$

A ratio higher than 1:1 (greater than 0 dB) indicates more signal than noise, which makes signal easier to be detected. See Table 3.1 for the conversion between ψ, σ_n^2, and SNR_{dB}.

ψ	σ_n^2	SNR	SNR_{dB}
0.1	0.000625	100	20dB
1	0.0625	1	0dB
10	6.25	0.01	-20dB

Table 3.1: Conversion from ψ to the absolute noise variance σ_n^2 and SNR_{dB}.

Next, we introduce how we simulate data for two sets of noise experiments.

3.2.2 Scenario I

In **Scenario I**, we're interested in how the BPA method performs when SNR changes and whether it is robust against added noise ratio.

By using the model described in the section above, we simulated 100 data sets for each SNR_{dB}. For each of the data set, there are $T = T_0 + 10^4$ observations, where $T_0 = 10^2$. The T_0 samples are for initialization, and they are discarded before doing the estimation. There are total $p = q_{max} * K_u + p_v$ predictors. The max VAR order is $q_{max} = 4$, so the number of endogenous predictors is $q_{max} * K_u = 4 * 4 = 16$. The number of exogenous predictors p_v is set to 84 to make the total number of predictors 100.

3.2.3 Scenario II

Our **Scenario II** is a modification of **Scenario I** on σ and σ_n. Here the variance of the noise σ_n^2 is such that $\sigma_n^2 \in \{0.025, 0.05, 0.075, 0.1, 1\}$. This noise setting matches with the values of noise variance used in Bijral (2019).

The standard deviation of matrix coefficients and exogenous variables that we simulated, σ, is equal to σ_n. The SNR is exactly 0dB or 1:1 variance of the signal to variance of the noise, i.e., $\psi = 1$ in Table 3.1. Therefore, increasing σ_n^2 does not effectively increase the noise level in response to the signal. It magnifies the magnitude of both the signal and the noise. We are interested in how the magnitude of signal and noise will affect the performance of BPA when SNR stays at 0dB. Note that when SNR_{dB}=0dB in **Scenario I**, $\sigma_n^2 = (1 * 0.25)^2 = 0.0625$ (see Table 3.1), which is pretty close to the value of variance $\sigma_n^2 = 0.05$ in **Scenario II**. Therefore, we expect to observe similar results in these two cases.

3.3 Parameter settings

Parameter a_T a_T is the number of observations in a block, referred to in the discussion in Chapter 2. It is usually set to some multiple of seasonality in the data according to Bijral (2019), and set to $\sqrt{T}$ or $\log T$ in the absence of seasonality. Reference Bijral (2019) suggested there is no significant difference between these different choices of a_T. Therefore for our simulated data, we set the default a_T to $\sqrt{T} = \sqrt{10^4} = 10^2$. However, it is also interesting to explore $a_T = \log T$ to see if it does not have a noticeable difference to $a_T = \sqrt{T}$ in terms of TPR/FPR since $\log(10^4) = 4$ is much smaller than the value for $\sqrt{10^4} = 10^2$.

The relationship of $T/2a_T = \mu_T$ means a_T determines the number of odd blocks. When a_T is larger, we have longer blocks and, therefore, fewer odd blocks. The distance between two odd blocks is also larger when a_T increases, making the blocks more independent of each other. However, we are interested in whether increasing the length of blocks will impact the TPR/FPR. We conduct experiments to find out the optimal length of blocks that gives the highest TPR while a relatively low FPR.

Since $T/4 = a_T * l_T$ and $10^4/4 = 2500$. Therefore, to make l_T an integer, 2500 should be a multiple of a_T. If l_T is not an integer, we will have an unequal number of O' and O'' blocks which we will have to discard some data to make the block numbers equal. To avoid unnecessary waste of data, we find all the possible values of a_T that give integers l_T. See Table 3.2 for a_T values and their corresponding μ_T and l_T. Note that for our experiment we only tested on 6 a_T values, $a_T \in \{5, 10, 50, 100, 250, 500\}$ (bold rows in Table 3.2). These a_T values should be sufficient for the purpose of our analysis.

Possible a_T value	μ_T (number of O blocks)	l_T (number of O' blocks)
1	5000	2500
2	2500	1250
4	1250	625
5	**1000**	**500**
10	**500**	**250**
20	250	125
25	200	100
50	**100**	**50**
100	**50**	**25**
125	40	20
250	**20**	**10**
500	**10**	**5**
625	8	4
1250	4	2
2500	2	1

Table 3.2: Values of a_T and corresponding μ_T and l_T. Bold rows are tested in later parameter experiment on a_T.

Parameter q q determines the first variable selection criterion λ_q in BPA, so we are interested to find out how much it will affect the performance of the BPA method. The lower the q, the harsher the Lasso penalty (see Table 2.1). The value of $q = 0.4p$ was set as default in Bijral (2019), with $q = 0.2p$ and $q = 0.6p$ tested in their real data experiment. They found that BPA with $q = 0.2p$ has a lower RMSE/MAPE than $q = 0.6p$. It was noted in their work that the choice of q does make a difference to the result, as a small q would exclude signal variables, and a large q would add noisy variables to the selection of variables. When $q = 0.2p$, it might be too conservative, while $q = 0.6p$ is less conservative and easier to let predictors in for the second screening controlled by ϕ. In the absence of prior knowledge on selecting q, selecting a middle ground value for q, i.e., $0.4p$, may be a safe choice. Hence, we want to explore the different parameter values of $q \in \{0.2p, 0.4p, 0.6p\}$ on the TPR and FPR, while the default is set at $q = 0.4p$.

Parameter ϕ ϕ is the second variable selection criterion in BPA. Therefore, we will pay close attention to the relationship between q and ϕ. For a lower ϕ threshold, it is easier for variables to be selected, while it will get harder when the ϕ threshold is higher. For example, only the most prominent and useful variables will get selected for a threshold of 0.9. Therefore, it is expected that the final number of variables selected will be fewer when we raise the ϕ threshold. The value for ϕ is set to 0.8 in Bijral (2019). We will explore if the harsher selection criteria will result in a better variable selection result in terms of TPR/FPR. The parameter value tested are $\phi \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, while the default is set at $\phi = 0.8$.

Parameter B B is the number of iterations, and it is set to 50 in Bijral (2019). Since more iterations make the BPA method more computationally intensive, we want to find out whether lowering B can retain the same BPA performance. B is tested for values $\{5, 25, 50\}$, with a default value at 50.

When experimenting on B, we keep in mind that each iteration should give a different selection of O' and O'' blocks. Therefore we must ensure a sufficient number of possible combinations to make each iteration's sampling with replacement different from the others. We check that using Table 3.3, and all number of possible combinations are greater than 50. This means we can always get at least 50 different possible selections for O' and O'' blocks. If we selected the same set of O' or O'' block as in any of the previous iterations, we can just re-sample again until we get something different.

a_T	μ_T	l_T	Number of possible combinations $\binom{\mu_T}{l_T}$
5	1000	500	$\binom{1000}{500} = 2.70e299$
10	500	250	$\binom{500}{250} = 1.17e149$
50	100	50	$\binom{100}{50} = 1.01e29$
100	50	25	$\binom{50}{25} = 1.26e14$
250	20	10	$\binom{20}{10} = 184756$
500	10	5	$\binom{10}{5} = 252$

Table 3.3: Number of possible combinations for each a_T values tested.

3.4 Experimental settings

Noise experiment We first do the noise experiment to test how the BPA method performs under different noise levels for Scenarios I and II. The TPR and FPR are calculated and presented later in the report. While noise level changes, the parameters are fixed to the following default values: $a_T = 100$, $q = 0.4p$, $\phi = 0.8p$, $B = 50$.

Experiment to investigate the influence of various parameters of BPA Then we try to find out the parameter values for optimal BPA performance by conducting parameter experiments. The default is $a_T = 100$, $q = 40$, $\phi = 0.8p$, $B = 50$ as suggested in Bijral (2019) and we change the parameter values to $a_T \in \{5, 10, 50, 100, 250, 500\}$, $q \in \{0.2p, 0.4p, 0.6p\}$, $\phi \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, $B \in \{5, 25, 50\}$ one at a time under different noise levels and calculate their respective TPR and FPR.

Performance evaluation We apply the same set of parameter values over the 100 simulated data sets for each noise level. We calculate the average TPR and FPR over B iterations for each data set, and plot the distribution of TPR and FPR for the 100 simulated data sets. This gives the final TPR and FPR distribution for the particular set of parameter values for that noise level. The noise levels are different in Scenarios I and II, so we perform parameter tests on each scenario.

3.4.1 A variant of the BPA

We are interested in the performance of BPA if blocks are not aggregated. Previously, for the BPA algorithm introduced in Chapter 2, the blocks $\{O'_1, ..., O'_{l_T}\}$ are aggregated to O' and $\{O''_1, ..., O''_{l_T}\}$ are aggregated to O''. Now we want to consider the case when $\{O'_1, ..., O'_{l_T}\}$ and $\{O''_1, ..., O''_{l_T}\}$ are not aggregated to the bigger blocks. That is, we are now only dividing the data into small blocks $\{O'_1, ..., O'_{\mu_T}\}$ for one run, and there is no

Algorithm 4 BPA-m

Input: $\mathbf{Y}$, $\mathbf{Z}$ (T measurements, p predictors),
$a_T \in \{5, 50, 100, 250, 500\}, q \in \{0.2p, 0.4p, 0.6p\}, \phi \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$,
$\Lambda = \{\lambda_1, ..., \lambda_{100}\}$ a sequence of regularizers
Initialize: $\Pi_B^{av}(k) = 0, \forall k \in \{1, ..., p\}, \mathbf{Y}_1 = \mathbf{Y}(:, 1 : a_T)$, $\mathbf{Z}_1 = \mathbf{Z}(:, 1 : a_T)$
q-Estimates: Solve

$$\min_{\mathbf{B}} \|\mathbf{Y}_1 - \mathbf{B}\mathbf{Z}_1\| + \lambda\|\mathbf{B}\|_1$$

and set $\lambda_q \in \Lambda$ to be the smallest λ that returns q active entries of $\mathbf{B}_1$.
Sample: Sequence of odd blocks $O = \{\mathbf{O}_1, ..., \mathbf{O}_{\mu_T}\}$, $\mu_T = \dfrac{T}{2a_T}$
for $l = 1$ to μ_T **do**
Set:

$$\hat{S}_{|\mathbf{O}_l|} = \{i : \hat{\mathbf{B}}_i \neq 0, \hat{\mathbf{B}} = \underset{\mathbf{B}}{\operatorname{argmin}} \|\mathbf{Y}(\mathbf{O}_l) - \mathbf{B}\mathbf{Z}(\mathbf{O}_l)\|^2 + \lambda_q\|\mathbf{B}\|_1\}.$$

BPA-m: $\Pi_l^{av}(k) = \Pi_l^{av}(k) + \mathbb{1}_{k \in \hat{S}_{|O_l|}}/\mu_T, \forall k \in \{1, ..., p\}$.
end for
Output: $\hat{S}^{av} = \{k : \Pi_l^{av}(k) \geq \phi\}$

random selection of blocks. By such, we do not aggregate the small blocks, and we perform Lasso on each small block.

The q-Estimate is also modified to select λ_q only based on the first a_T data. The reason for this change is because using the complete data for λ_q selection would give a very small λ_q that is not sufficient to perform effective regularization on data blocks of length a_T, where $a_T \in \{5, 50, 100, 250, 500\}$. When the λ value is small, the penalty is less, and more final predictors are selected. By doing some preliminary experiments, we found that using complete data for λ_q selection has resulted in the FPR being as high as TPR at around 0.5 even for the lowest noise level. The λ_q selected from all available data T used on the a_T length small blocks is too generous on the predictors, which means the penalty is not strong enough to discard the weak predictors. Therefore, when selecting λ_q, we should use the amount of data similar to the block size. BPA does not have this issue because the blocks are aggregated to O' and O'' with 2500 observations each, while there is a total of $T = 10^4$ observations. The size difference for BPA is not too significant, which using all T data will not select an inadequate λ_q for the regularization in O' and O''.

For simplicity, we call this modified variant of the BPA algorithm **BPA-m**, see Algorithm 4. We conduct experiments using Scenario I and II simulated data on the two algorithms to compare their performance empirically.

Since we have 100 predictors, when $a_T \in \{5, 10, 50\}$ BPA-m is in the case $p > n$; when $a_T = 100$ we have the case $p = n$; when $a_T \in \{250, 500\}$ we are in the case $p < n$. It will be interesting to see how this BPA variant method performs under situations when $p > n$, $p = n$ and $p < n$. We expect to see BPA-m performs poorly for $a_T \in \{5, 10, 50\}$ due to the high-dimensional data problem of $p > n$ for the base procedure Lasso and also the lack of independence between blocks. BPA does not have the high-dimensional problem because

the blocks are aggregated. Each big block O' and O'' has 2500 observations; hence the number of observations is far greater than the 100 predictors.

3.5 Simulated data results for Scenario I

3.5.1 BPA under different SNR level

The BPA experiment result under different SNR levels is presented in Figure 3.2. The TPR does not change a lot, and it stays at a pretty high level as SNR changes. However, there are more outliers and slightly lower TPR for higher SNR. The FPR drops as SNR increases, which is expected because the model is less likely to detect false predictors when the data is less noisy.

Our results are generally aligned with the results in Bijral (2019). The results suggest that BPA is quite robust against noise.

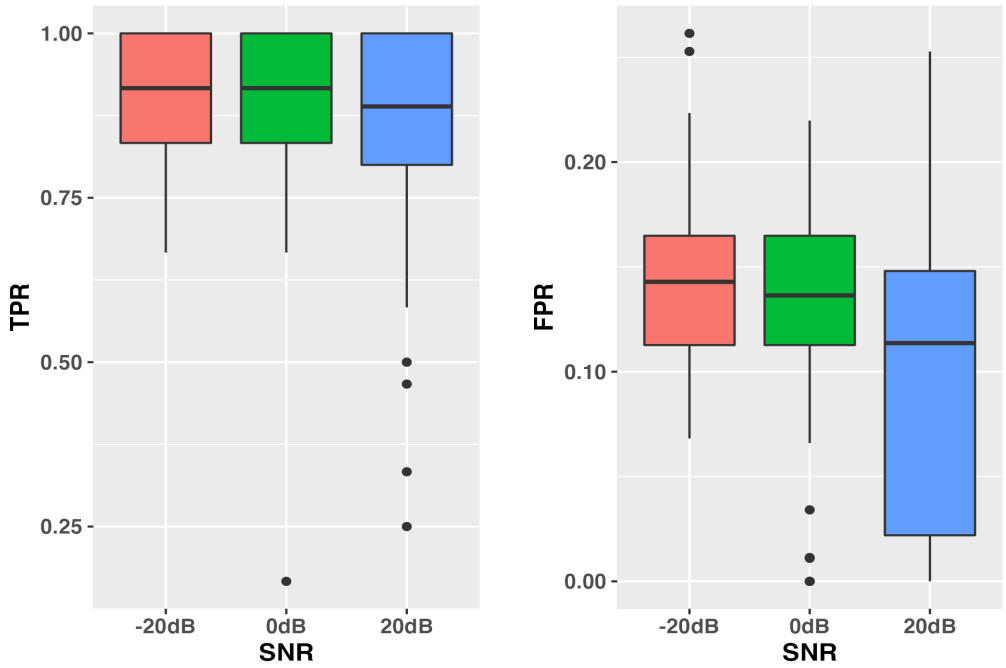


Figure 3.2: TPR/FPR for BPA under different SNR levels in Scenario I.

To explore further the cause of slightly lower TPR for higher SNR, we investigate the selection of λ_q, and the number of predictors selected p^* for each SNR level. To improve analysis, we removed the outliers from the λ_q plot (see the Appendix (Figure B.1) for the λ_q plot before removing the outlier). In Figure 3.3, we notice that λ_q are generally very similar; however, SNR at 20dB has more outliers and a broader range of λ_q values than the SNR at -20dB.

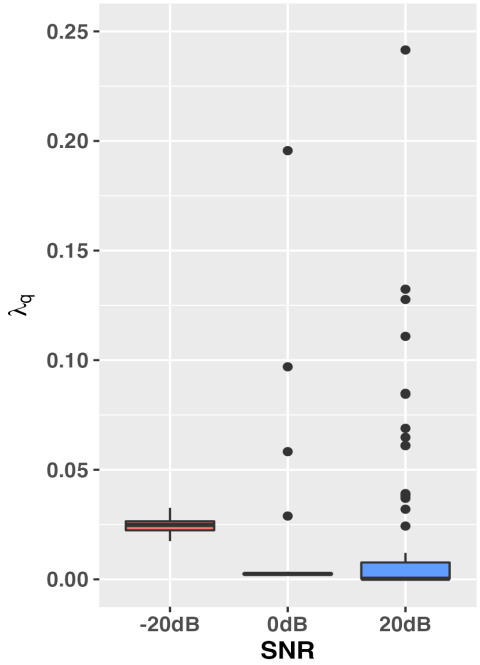
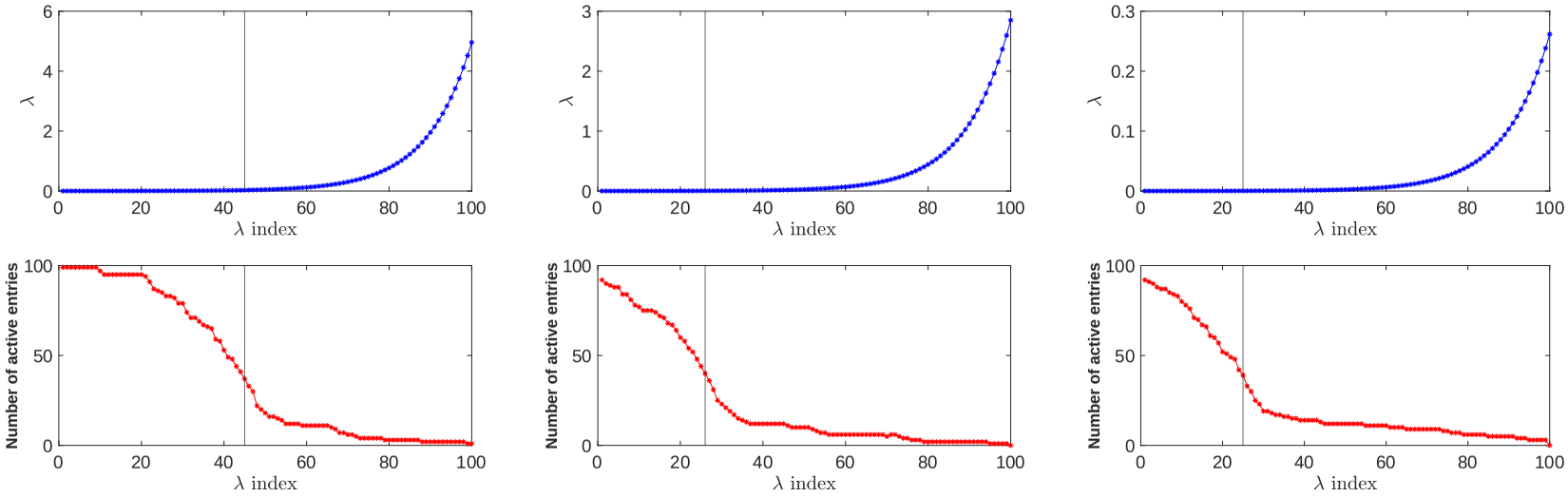


Figure 3.3: Boxpot for λ_q values, for BPA under different SNR levels in Scenario I (outlier removed). BPA parameters are fixed at $a_T = 100$, $q = 0.4p$, $\phi = 0.8$, $B = 50$.

We want to explore further the MATLAB `lasso` function that is used to select the Λ values, see Figure 3.4. The illustrations are not representative of the 100 simulated data sets for each SNR level. We observed the range of Λ values varies from data set to data set for the same SNR level.



(a) Range of Λ values for SNR=$-$20dB (b) Range of Λ values for SNR=0dB (c) Range of Λ values for SNR=20dB

Figure 3.4: Illustrations for the range of λ values generated by MATLAB `lasso` function on a data set for each SNR level. The black vertical line represents the index for λ_q. The range of Λ values varies from data set to data set, so the illustrations are not representative.

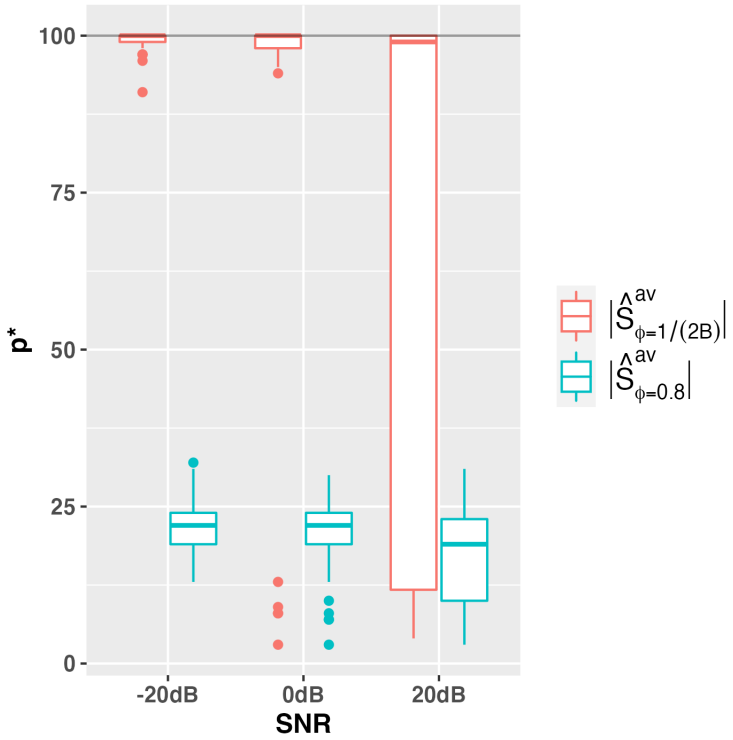


Figure 3.5: Number of predictors (p^*) before applying and after applying $\phi = 0.8$ for BPA under different SNR levels in Scenario I. BPA parameters are fixed at $a_T = 100$, $q = 0.4p$, $\phi = 0.8$, $B = 50$. The red boxes $|\hat{S}^{av}_{\phi=1/(2B)}|$, where $|\cdot|$ denotes cardinality, are the number of predictors that are selected at least once in any block. The blue boxes $|\hat{S}^{av}_{\phi=0.8}|$ are the number of predictors that are selected at least 80% of the two aggregated blocks for B iterations, i.e., at least selected $0.8 * 2 * 50 = 80$ times by O' and O'' blocks combined in 50 iterations. The solid black line represents the total number of predictors ($p = 100$). The first layer of predictor selection by λ_q happens when the number of predictors selected falls from the black line to the red boxes. The second layer of predictor selection by $\phi = 0.8$ happens when the number of predictors selected falls from the red boxes to the blue boxes.

Next, we want to investigate the effect of ϕ on the final number of predictors selected. Parameter ϕ is the second predictor selection criterion in BPA. Therefore, we pay close attention to the relationship between q and ϕ. The results, as shown in Figure 3.5, indicate that most predictor selection happens after applying $\phi = 0.8$ at the last step of BPA. This confirms the fact that Lasso regularization controlled by q only serves as an instrument for predictor selection as stated in Bijral (2019) and ϕ is the critical criterion for filtering high selection probability predictors selected by Lasso in each block to get the final model. In fact, it was shown in Bijral (2019) that the BPA selects only at most 28% of the low selection probability predictors selected by the base procedure. We also notice that there are fewer final predictors selected for SNR=20dB, which resulted in both its TPR and FPR being lower in Figure 3.2.

3.5.2 BPA under different parameters

Parameter a_T In Figure 3.6, we observe that as a_T increases, the TPR and FPR stay relatively constant. In fact, no matter the value of a_T, the size of the O' and O'' blocks remains the same at 2500 observations ($a_T \times l_T$). There are 100 predictors; therefore, there is no high-dimensional problem for any value of a_T in the experiment. Thus, we see no real difference between different a_T values for TPR and FPR. Our results prove the claim in Bijral (2019) suggesting there is no significant difference between the different choices of a_T.

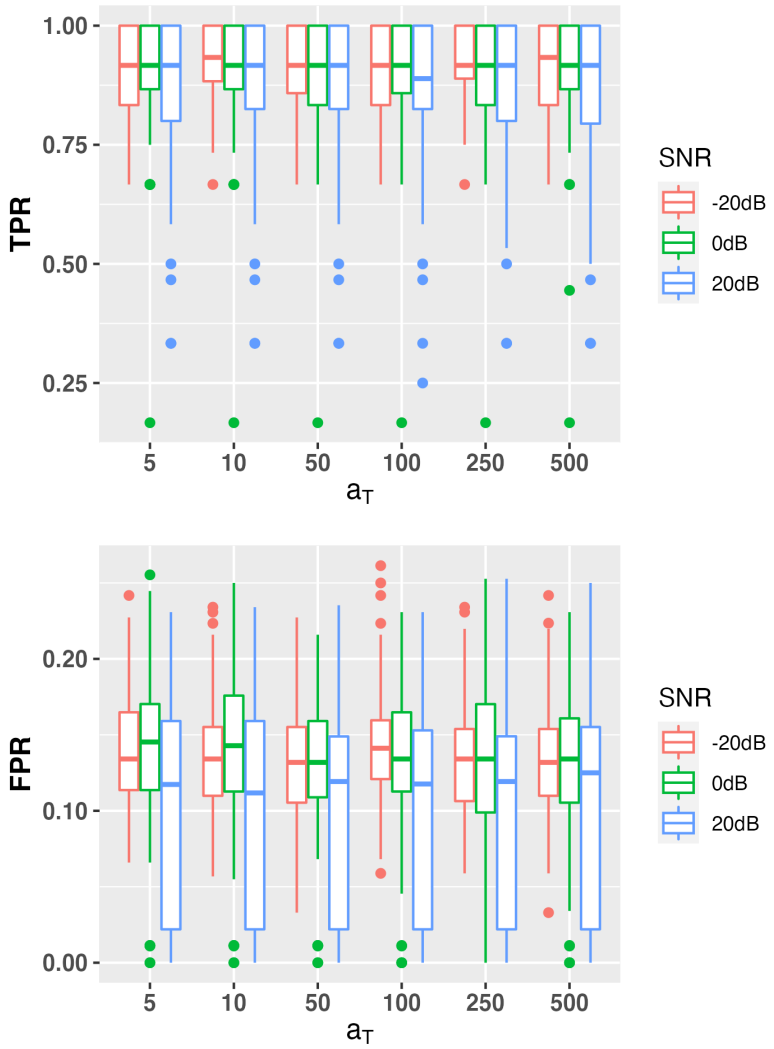


Figure 3.6: TPR/FPR for BPA under different a_T values in Scenario I. Other parameters are fixed at $q = 0.4$, $\phi = 0.8$, $B = 50$.

Parameter q In Figure 3.7, we observe that as q increases, the Lasso penalty gets weaker and more variables can pass through the criteria. Note that the second selection criterion

ϕ is fixed at 0.8 for the different q, and the two criteria together impact the final TPR and FPR. Our result suggests that while keeping the second selection criterion constant, as we loosen the first criterion by increasing q, we get higher TPR and higher FPR, so there's an apparent trade-off between TPR and FPR. Interestingly, the data in this result graph shows a much more significant increase in FPR than the increase in TPR. The much higher FPR for $q = 0.6p$ suggests that the loose Lasso regularization performed at this step to select the variables has a considerable impact on the correct final selections. Therefore, there is a need to be strict at this step. When $q = 0.2p$, the TPR is lower, but we also get a much lower FPR. We'll need to determine whether a strict criterion is needed that sacrifices TPR for the lower FPR. The influence of $\phi = 0.8$ (the second condition being quite harsh) can only be effective at filtering out the less good variables and keeping the FPR relatively low when the variable cannot pass the first criteria easily.

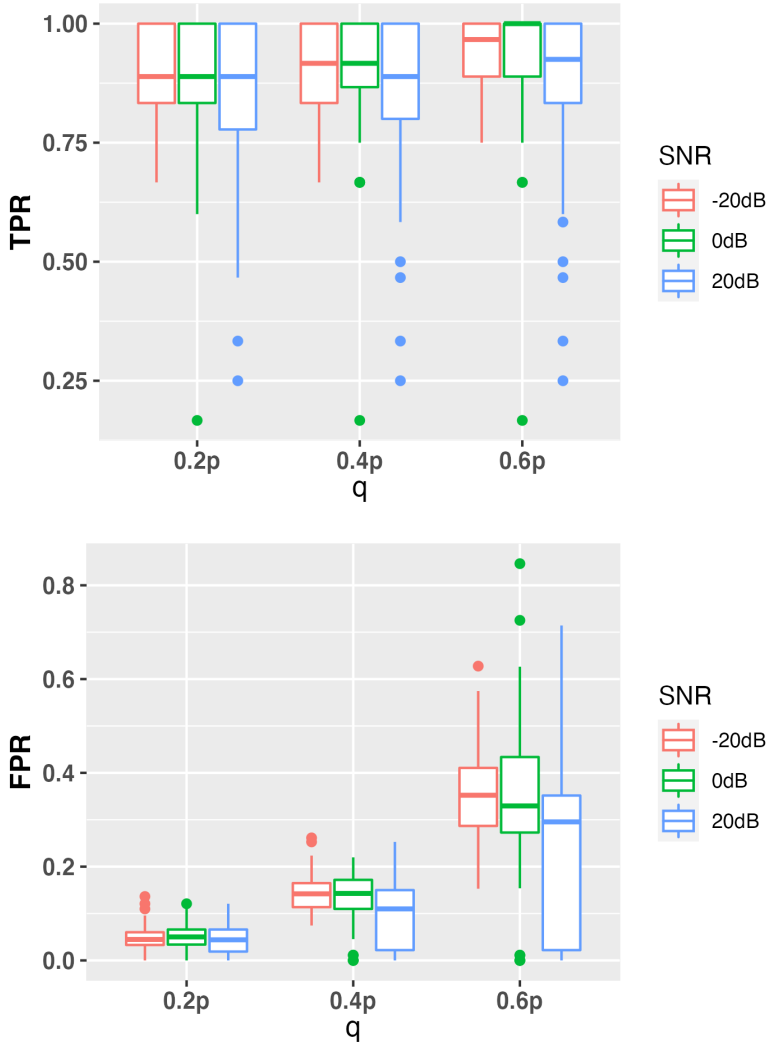
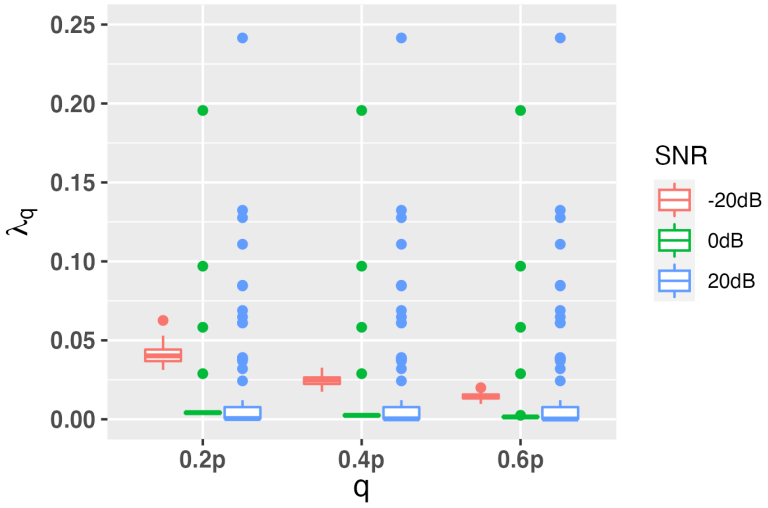


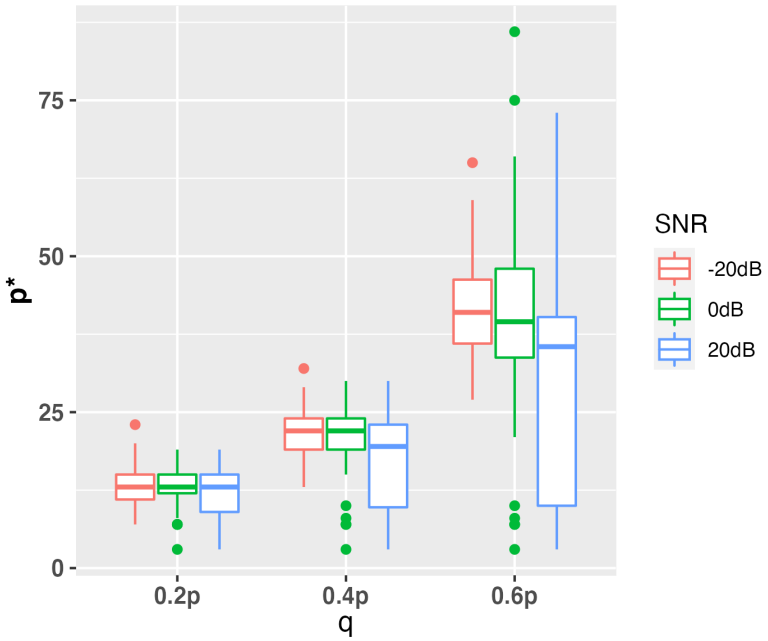
Figure 3.7: TPR/FPR for BPA under different q values in Scenario I. Other parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 50$.

To see the impact of q more clearly, we investigate its effect on λ_q and the number of final

predictors p^* selected. To improve analysis, we removed the outliers in the TPR plot for λ_q (see the Appendix (Figure B.2) for the TPR plot before removing the outliers). From the results in Figure 3.8, there's not much difference in λ_q except for SNR=−20dB where λ_q decreases. The number of final predictors increases significantly when q increases. Since there are more final predictors for higher q, the TPR and FPR are higher for higher q. Overall, we should use a medium q value and allow adequate but not excessive variables to pass through and then leave the second selection criterion to select the variables.



(a) λ_q plot for different q (outliers removed)



(b) p^* plot for different q

Figure 3.8: $\lambda_\mathbf{q}$ selection and number of final predictors $\mathbf{p}^*$ for BPA under different SNR levels in Scenario I. Parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 50$.

Parameter ϕ In Figure 3.9, we observe that as ϕ increases, the FPR drops together with a slight drop in TPR. There is no apparent trade-off between TPR and FPR. $\phi \in \{0.5, 0.6, 0.7\}$ is completely not acceptable for FPR. Therefore, ϕ should not be less than 0.8. $\phi = 0.9$ is acceptable but may be too harsh because of the slightly lower TPR. Therefore it is good to set the default at 0.8.

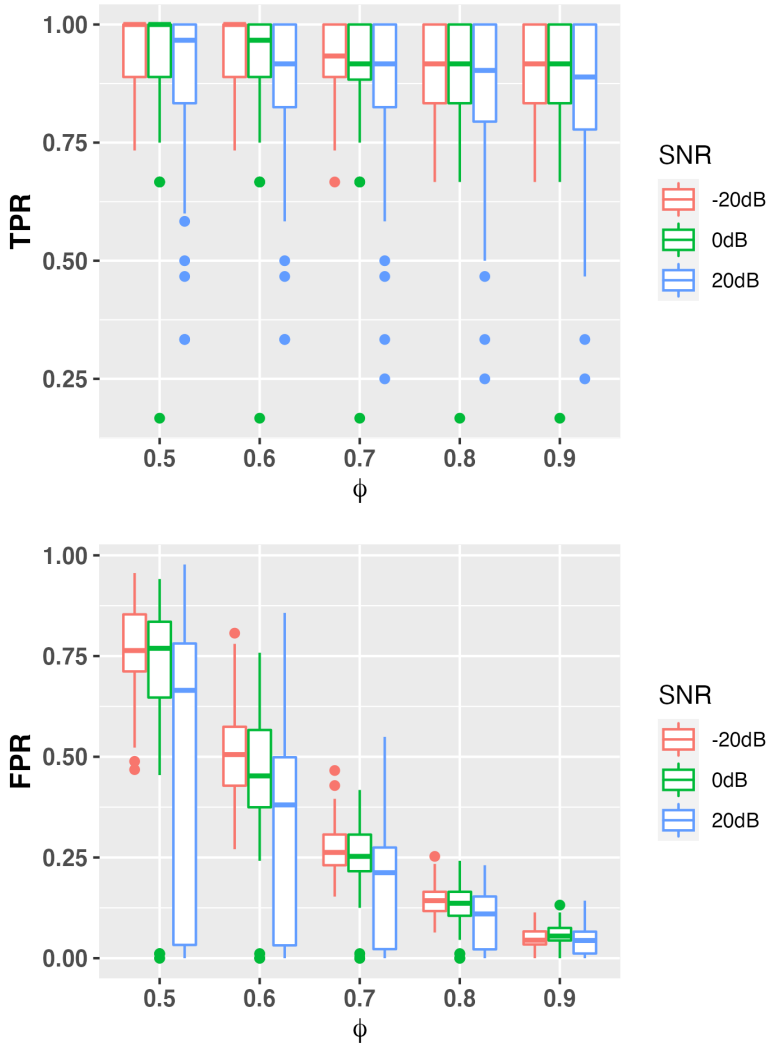


Figure 3.9: TPR/FPR for BPA under different ϕ values in Scenario I. Other parameters are fixed at $a_T = 100$, $q = 0.4$, $B = 50$.

Parameter B In Figure 3.10, we observe that the result for B suggests no significant difference in TPR/FPR between 25 and 50 iterations. This means that 25 iterations are sufficient to arrive at a good selection of the final predictors. Still, if we have the computing capacity to do 50 iterations, it will help improve the FPR slightly. However, when we reduce iterations to 5, the FPR is significantly higher than in the cases of 25 and 50 iterations. Therefore, we should probably not use too few iterations.

Our result aligns with the value recommended in Shah and Samworth (2013) to set $B = 50$ as the default value. They stated that choosing B larger than 50 increases the computational burden and may lead to the r-concavity assumptions being violated.

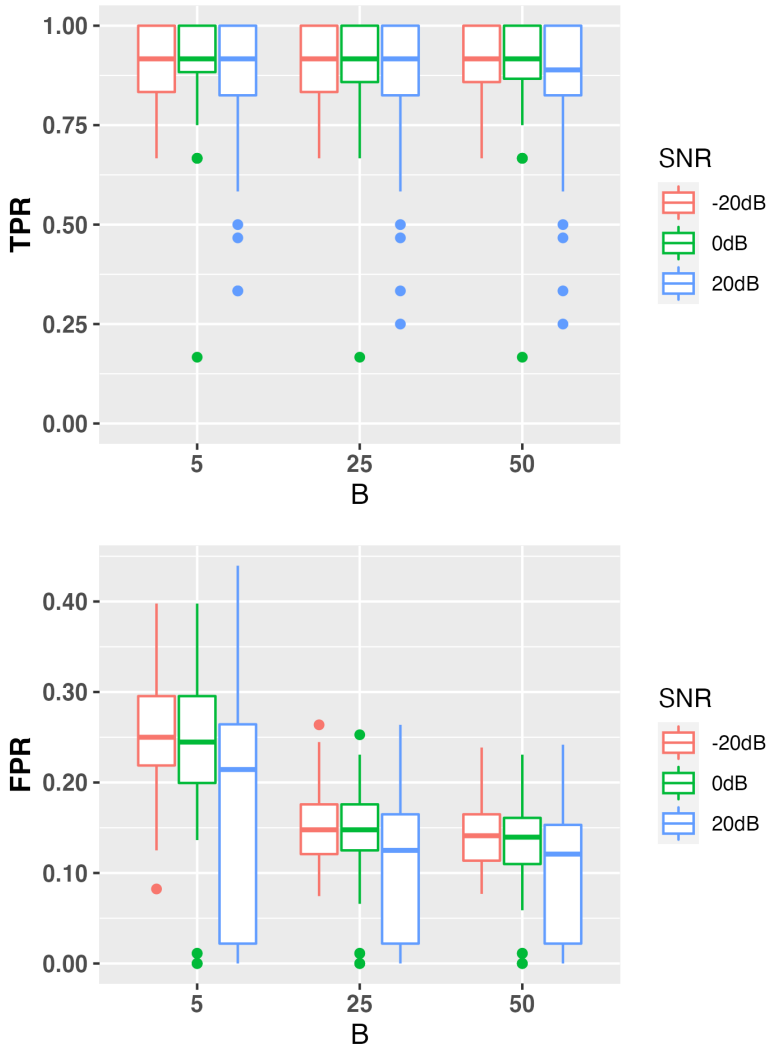


Figure 3.10: TPR/FPR for BPA under different B values in Scenario I. Other parameters are fixed at $a_T = 100$, $q = 0.4$, $\phi = 0.8$.

Overall observation TPR is usually similar for different SNRs when the parameters change, and FPR is generally lower for higher SNRs. Therefore, the pattern discussed for Figure 3.2 remains the same for different BPA parameter values.

3.5.3 Comparison between BPA and BPA-m

The result from Figure 3.11 indicates that BPA-m TPR drops significantly when noise increases. The FPR stays lower than 0.02 as noise increases. Overall, this reveals that the BPA-m method is robust to the noise in terms of FPR but not TPR.

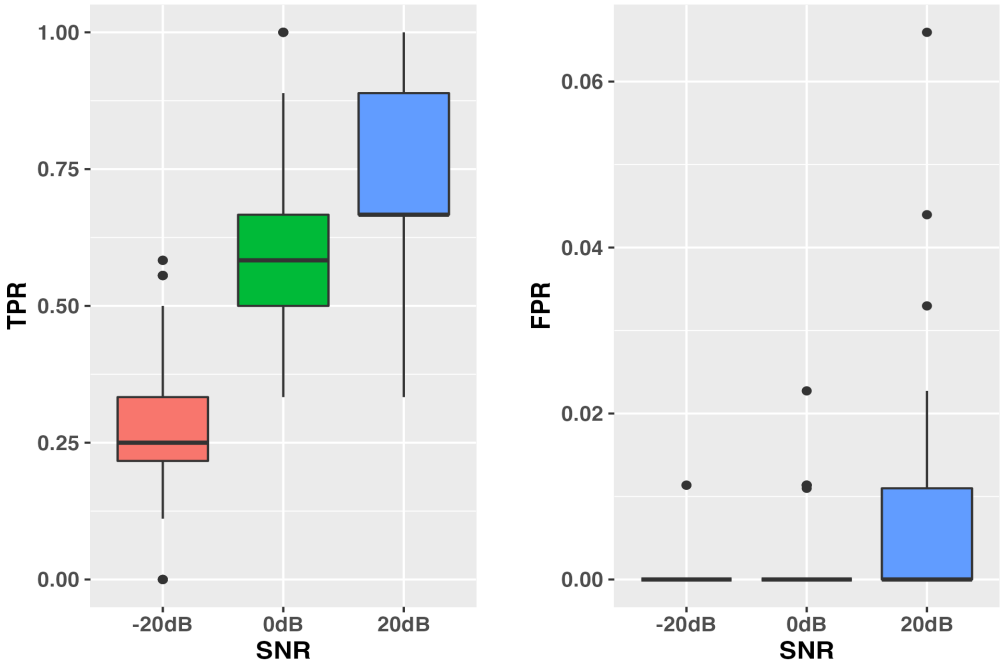


Figure 3.11: TPR/FPR for BPA-m under different SNR levels for Scenario I.

The result from Figure 3.12 indicates that BPA performs a lot better than BPA-m in terms of TPR but performs worse in terms of FPR. Therefore, BPA-m is too conservative compared to BPA, and BPA-m cannot achieve the performance of BPA.

We also noticed the TPR improvement from increasing SNR is clear for BPA-m but not for BPA; however, the FPR improvement from increasing SNR is clear for BPA but not for BPA-m. This is probably due to the number of final predictors selected by the two different models.

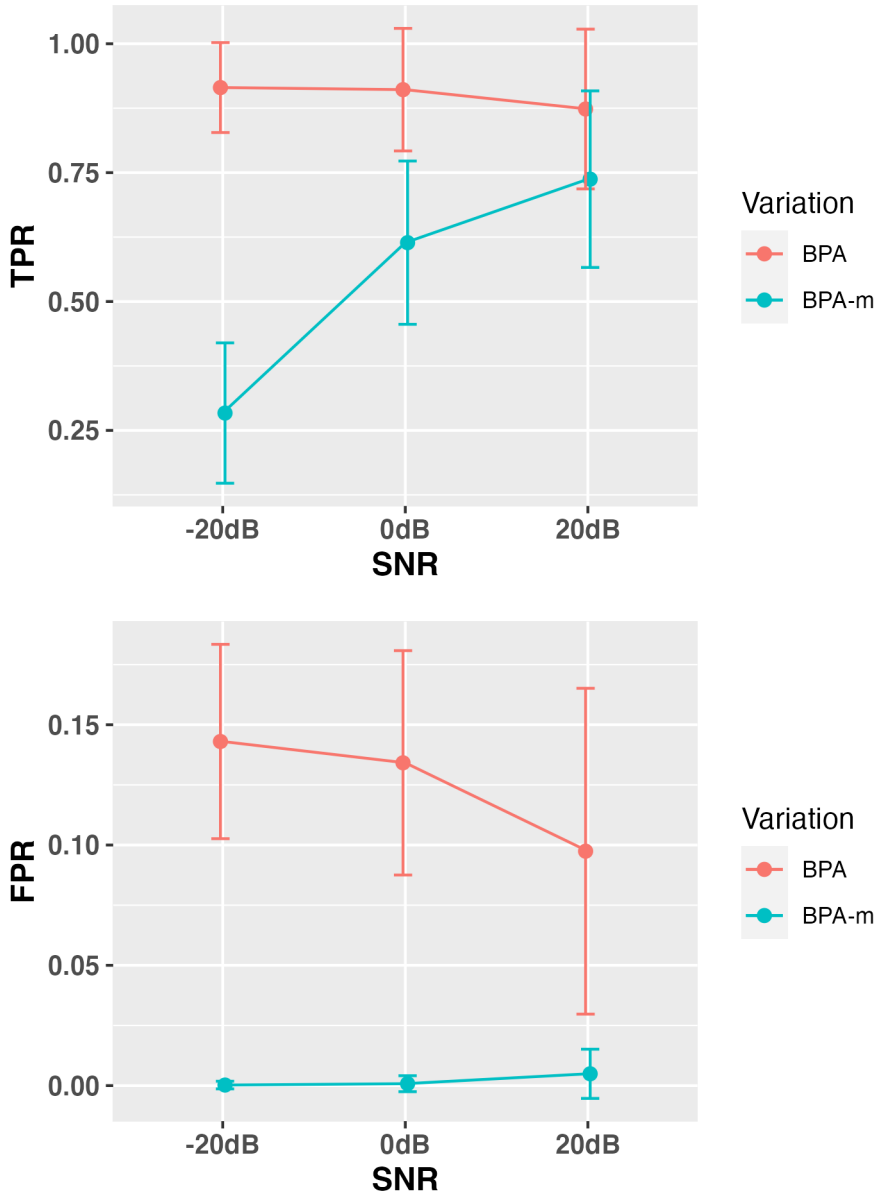


Figure 3.12: Comparison of TPR/FPR between BPA and BPA-m under different SNR values for Scenario I. Other parameters are fixed at $a_T = 100$, $q = 0.4p$, $\phi = 0.8$, $B = 50$.

Further parameter experiments on BPA-m for Scenario I are included in the Appendix B.2 for the sake of conciseness. The results from those experiments give similar optimal parameter values as for BPA.

From the results in Figure B.3, we notice the effect of high-dimensional problem in BPA-m. As a_T increases, the TPR increases. The TPR is around 0 when $a_T = 5$ and $a_T = 10$. This is expected before our experiment because there are only five observations but 100 predictors. Doing Lasso on these small blocks will greatly suffer from the $p > n$ high-

dimensional problem. Recall that these small a_T values also suffer from the independence problem between blocks. Therefore, we see that $a_T \in \{5, 10, 50\}$ of BPA-m have very low TPR. Then, when we come to the $a_T = 100$, which is when $p = n$, the TPR is around 0.6, significantly higher than the smaller values of a_T. After that, we come to the case when $p < n$, where the blocks no longer suffer from the high-dimensional problem. The TPR continues to increase; however, the increase seems to be slightly levelling off. The TPR does not improve that much from 250 to 500. The FPR stays pretty constant and below 0.1 as a_T increases, but there are more outliers when a_T increases. Therefore there are not many trade-offs between TPR and FPR. Overall, this suggests we should use a higher a_T value.

3.6 Simulated data results for Scenario II

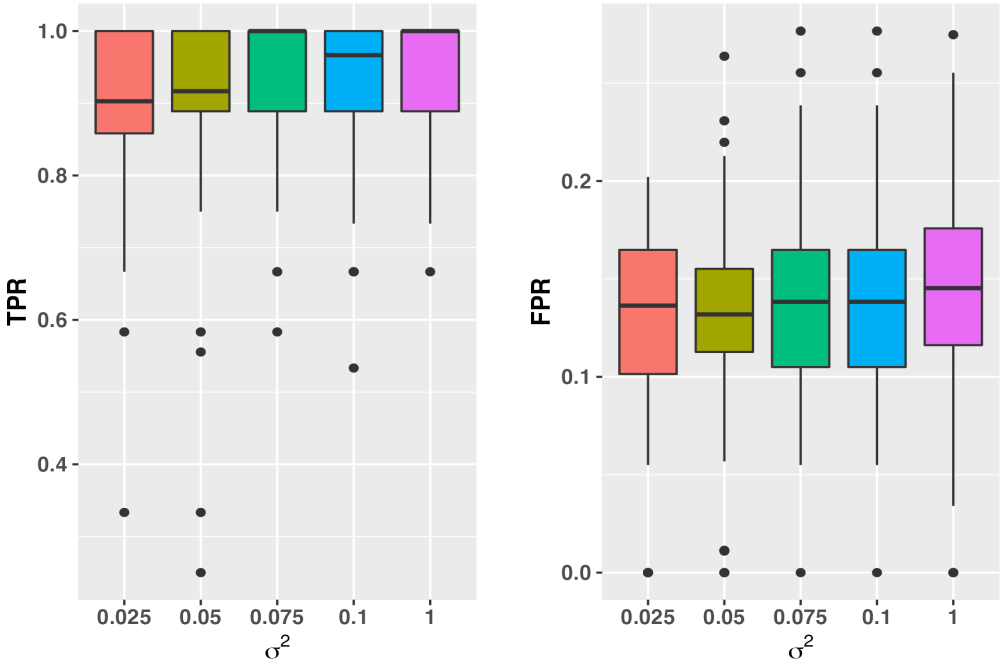
We are interested in the performance of BPA and BPA-m under Scenario II. The result from Figure 3.13 suggests that the TPR and FPR do not vary a lot when σ^2 changes. This reveals that changing σ^2 (magnitude of signal and noise) in Scenario II data does not affect much. The slight difference is ambiguous to draw a conclusion on the effect because the TPR and FPR are not monotonic functions of σ^2. However, when the magnitude is at 1, we do see a slightly higher TPR and FPR. The higher TPR may be because the signal is easier to detect when the signal magnitude is much larger. At the same time, the noise magnitude is also more prominent, so the model becomes more susceptible to noise which results in a higher FPR. Overall, this analysis could not identify a clear difference for different σ^2 in TPR/FPR.

Note that when SNR_{dB}=0dB in Scenario I, $\sigma_n^2 = (1 * 0.25)^2 = 0.0625$ (see Table 3.1), which is pretty close to the value of variance $\sigma_n^2 = 0.05$ in Scenario II. Therefore, we indeed observe similar results for these two cases.

To examine closely how the magnitude of noise when SNR=0dB affects the performance of BPA, we conduct experiments under the different σ^2-values. In Figure B.6 (see Appendix), we do not see many changes in λ_q and the number of final predictors selected. This suggests that the changing magnitude of σ^2 does not affect the first and second layers of predictor selection.

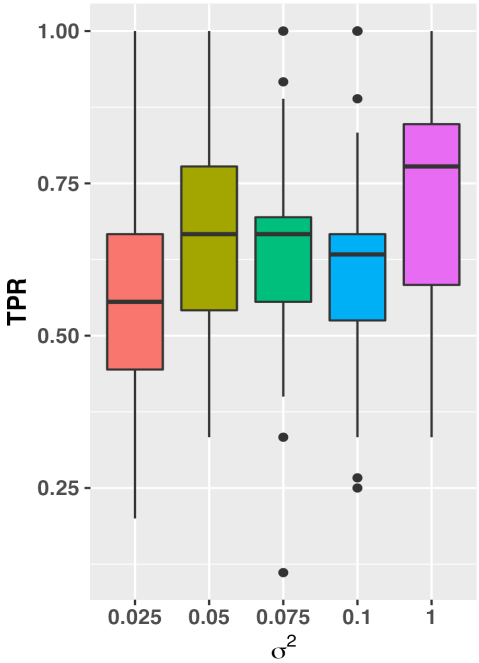
The result from Figure B.7 indicates that BPA performs better than BPA-m in terms of TPR but performs worse in terms of FPR in Scenario II. Since TPR are usually more critical than FPR, we believe BPA is the superior method.

Further parameter experiments on BPA and BPA-m for Scenario II are included in the Appendix B.3 and B.4 for the sake of conciseness. Those experiments show similar optimal parameter values for BPA and BPA-m as in Scenario I.

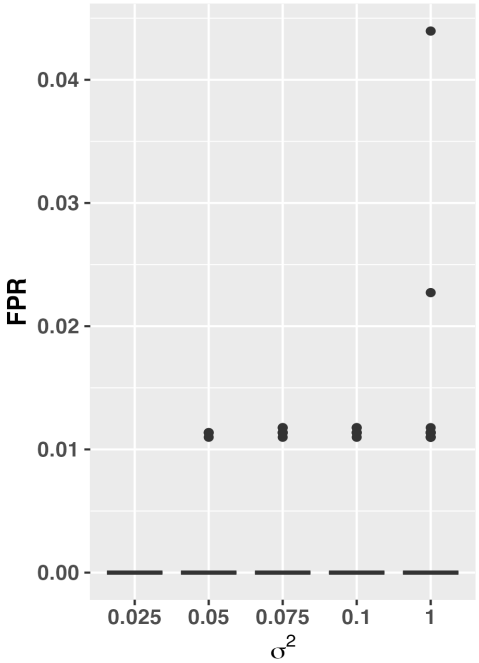


(a) BPA

(b) BPA



(c) BPA-m



(d) BPA-m

Figure 3.13: TPR/FPR for BPA and BPA-m under different σ^2 levels in Scenario II.

Chapter 4

Experiments with real data

4.1 Preliminaries

In this chapter, we analyse the applicability of the BPA stability method on the air pollution data set used in Li et al. (2019). We want to explore how the BPA stability method performs on a real-life data set where the true predictors are unknown. The goal is to use BPA selected predictors to forecast future measurements. The NMSE (normalised mean square error) is computed for the BPA predictions and is compared against the other predictive models in Li et al. (2019). Specifically, those models have different stopping rules for the matching pursuit algorithm (MPA). MPA is a greedy algorithm; therefore, a stopping rule is used for its model selection.

4.2 Air pollution data

The Auckland air pollution data set is a daily measurement of the concentrations of particulate matter (PM), specifically $PM_{2.5}$ and PM_{10} (in $\mu g/m^3$) at four locations in Auckland. $PM_{2.5}$ includes particles less than 2.5 μm in diameter, PM_{10} those less than 10 μm in diameter. Therefore, $PM_{2.5}$ is a subset of PM_{10}. The sites where the data were collected are Patumahoe, Penrose, Takapuna, and Whangaparaoa, and we will use PA, PE, TA, WH to represent them respectively in this work. The measurements are from 30/04/2008 to 30/06/2014.

For a measurement site, let $\theta_{\text{site}}(1), \theta_{\text{site}}(2), ...$ be the time series of log-transformed daily concentrations of $PM_{2.5}$. For example,$\theta_{\text{PA}}(1)$ is the $PM_{2.5}$ measurement on the first day for Patumahoe. Similarly, $\xi_{\text{site}}(1), \xi_{\text{site}}(2), ...$ are the log-transformed values for the concentrations of PM_{10} measured at the specific site, during day $1, 2, ...$

4.3 BPA experimental settings

4.3.1 Mathematical model

We want to find a linear model which describes the relationship between the log-transformed concentration of $PM_{2.5}$ on the current day at a specific site and the following variables: (i) past and present log-transformed concentrations of PM_{10} for the specific site and (ii) past and present log-transformed concentrations of $PM_{2.5}$ for all other three sites.

Let $n = 365$ since it is the number of measurements for an entire year. We take the response vector to be

$$\mathbf{y}_{\text{site}}(t) = [\theta_{\text{site}}(t)\ \theta_{\text{site}}(t-1)\ ...\ \theta_{\text{site}}(t-n+1)]^T, \tag{4.1}$$

where $(\cdot)^T$ denotes transposition.

For a certain site, we use the notation $\mathbf{X}_{\text{site}}(t)$ for the matrix of predictors at time moment t. The columns of the matrix are arranged into four blocks. The representations for the response vector and the matrix of predictors for each site are shown in Figure 4.1. The response vector and the columns of the predictors' matrix are centred and standardised.

We consider the following two scenarios for constructing the four blocks in the matrix of predictors. The two scenarios have been used in Li et al. (2019).

Scenario (a) - Full set of predictors (FullSet) The first block contains the measurements from present to past one year log-transformed concentrations of PM_{10} for the specific site that we want to predict. The next three blocks contain the present to past one-year log-transformed concentrations of $PM_{2.5}$ for all other three sites. The predictors are shown in Figure 4.1. The total number of predictors $p = 4(n+1) = 1464$ is much larger than n, thus $p > n$ and there is high-dimensional problem.

Scenario (b) - Constrained set of predictors (ConSet) In this scenario, we reduce the total number of predictors by using empirical knowledge from air pollution scientists. The first block contains the present, the recent past, six months ago and one year ago log-transformed concentrations of PM_{10} for the specific site that we want to predict. The next three blocks contain the present, the recent past, six months ago and one year ago log-transformed concentrations of $PM_{2.5}$ for all other three sites. The predictors are shown in Figure 4.2. The total number of predictors is $p = 4 \times 17 = 68$, thus $p < n$.

Response vector	Matrix of predictors
$\mathbf{y}_{\text{PA}}(t) = [\boldsymbol{\Theta}_{\text{PA}}(t)]_{:1}$	$\mathbf{X}_{\text{PA}}(t) = [\boldsymbol{\Xi}_{\text{PA}}(t)\ \boldsymbol{\Theta}_{\text{PE}}(t)\ \boldsymbol{\Theta}_{\text{TA}}(t)\ \boldsymbol{\Theta}_{\text{WH}}(t)]$
$\mathbf{y}_{\text{PE}}(t) = [\boldsymbol{\Theta}_{\text{PE}}(t)]_{:1}$	$\mathbf{X}_{\text{PE}}(t) = [\boldsymbol{\Xi}_{\text{PE}}(t)\ \boldsymbol{\Theta}_{\text{PA}}(t)\ \boldsymbol{\Theta}_{\text{TA}}(t)\ \boldsymbol{\Theta}_{\text{WH}}(t)]$
$\mathbf{y}_{\text{TA}}(t) = [\boldsymbol{\Theta}_{\text{TA}}(t)]_{:1}$	$\mathbf{X}_{\text{TA}}(t) = [\boldsymbol{\Xi}_{\text{TA}}(t)\ \boldsymbol{\Theta}_{\text{PA}}(t)\ \boldsymbol{\Theta}_{\text{PE}}(t)\ \boldsymbol{\Theta}_{\text{WH}}(t)]$
$\mathbf{y}_{\text{WH}}(t) = [\boldsymbol{\Theta}_{\text{WH}}(t)]_{:1}$	$\mathbf{X}_{\text{WH}}(t) = [\boldsymbol{\Xi}_{\text{WH}}(t)\ \boldsymbol{\Theta}_{\text{PA}}(t)\ \boldsymbol{\Theta}_{\text{PE}}(t)\ \boldsymbol{\Theta}_{\text{TA}}(t)]$

Table 4.1: Response vector and the matrix of predictors for the air pollution data. The blue and purple boxes represent the blocks of predictors. The subscript in $[\boldsymbol{\Theta}_{\text{site}}(t)]_{:1}$ means the 1st column in the matrix $\boldsymbol{\Theta}_{\text{site}}(t)$. The exact predictors are defined in Figure 4.1 for Scenario (a) FullSet and in Figure 4.2 for Scenario (b) ConSet.

Measurements for $\text{PM}_{2.5}$ when site $\in \{\text{PA}, \text{PE}, \text{TA}, \text{WH}\}$

$$\boldsymbol{\Theta}_{\text{site}}(t) = \begin{bmatrix} \theta_{\text{site}}(t) & \theta_{\text{site}}(t-1) & \cdots & \theta_{\text{site}}(t-n) \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{\text{site}}(t-n+1) & \theta_{\text{site}}(t-n) & \ldots & \theta_{\text{site}}(t-2n+1) \end{bmatrix}$$

Measurements for PM_{10} when site $\in \{\text{PA}, \text{PE}, \text{TA}, \text{WH}\}$

$$\boldsymbol{\Xi}_{\text{site}}(t) = \begin{bmatrix} \xi_{\text{site}}(t) & \xi_{\text{site}}(t-1) & \cdots & \xi_{\text{site}}(t-n) \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{\text{site}}(t-n+1) & \xi_{\text{site}}(t-n) & \ldots & \xi_{\text{site}}(t-2n+1) \end{bmatrix}$$

Figure 4.1: Scenario (a) - Full set of predictors (FullSet). The entries on the columns of the matrix in purple $\boldsymbol{\Theta}_{\text{site}}(\mathbf{t})$ are the $\text{PM}_{2.5}$ measurements for the past 365 days at each time point in t to $t-n$. The entries on the columns of the matrix in blue $\boldsymbol{\Xi}_{\text{site}}(\mathbf{t})$ are the PM_{10} measurements for the past 365 days at each time point in t to $t-n$.

Measurements for $\mathrm{PM}_{2.5}$ when site $\in \{\mathrm{PA}, \mathrm{PE}, \mathrm{TA}, \mathrm{WH}\}$

$$\boldsymbol{\Theta}_{\text{site}} = \begin{bmatrix} \theta_{\text{site}}(t) & \theta_{\text{site}}(t-1) & \cdots & \theta_{\text{site}}(t-10) & \theta_{\text{site}}(t-182) & \theta_{\text{site}}(t-183) & \theta_{\text{site}}(t-184) & \theta_{\text{site}}(t-n+2) & \theta_{\text{site}}(t-n+1) & \theta_{\text{site}}(t-n) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \theta_{\text{site}}(t-n+1) & \theta_{\text{site}}(t-n) & \cdots & \theta_{\text{site}}(t-n-9) & \theta_{\text{site}}(t-n-181) & \theta_{\text{site}}(t-n-182) & \theta_{\text{site}}(t-n-183) & \theta_{\text{site}}(t-2n+3) & \theta_{\text{site}}(t-2n+2) & \theta_{\text{site}}(t-2n+1) \end{bmatrix}$$

Measurements for PM_{10} when site $\in \{\mathrm{PA}, \mathrm{PE}, \mathrm{TA}, \mathrm{WH}\}$

$$\boldsymbol{\Xi}_{\text{site}} = \begin{bmatrix} \xi_{\text{site}}(t) & \xi_{\text{site}}(t-1) & \cdots & \xi_{\text{site}}(t-10) & \xi_{\text{site}}(t-182) & \xi_{\text{site}}(t-183) & \xi_{\text{site}}(t-184) & \xi_{\text{site}}(t-n+2) & \xi_{\text{site}}(t-n+1) & \xi_{\text{site}}(t-n) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \xi_{\text{site}}(t-n+1) & \xi_{\text{site}}(t-n) & \cdots & \xi_{\text{site}}(t-n-9) & \xi_{\text{site}}(t-n-181) & \xi_{\text{site}}(t-n-182) & \xi_{\text{site}}(t-n-183) & \xi_{\text{site}}(t-2n+3) & \xi_{\text{site}}(t-2n+2) & \xi_{\text{site}}(t-2n+1) \end{bmatrix}$$

Figure 4.2: Scenario (b) - Constrained set of predictors (ConSet). The entries on the columns of the matrix in purple $\boldsymbol{\Theta}_{\text{site}}(\mathbf{t})$ are the $\mathrm{PM}_{2.5}$ measurements for the past 365 days at each time point in t to $t-10$, and $t-182$ to $t-184$, and $t-n+2$ to $t-n$. The entries on the columns of the matrix in blue $\boldsymbol{\Xi}_{\text{site}}(\mathbf{t})$ are the PM_{10} measurements for the past 365 days at each time point in t to $t-10$, and $t-182$ to $t-184$, and $t-n+2$ to $t-n$.

4.3.2 BPA parameter settings

The BPA parameters that need to be set for the real data experiment are listed at the start of Algorithm 3.

In our experiment, the data specific parameters settings are:

- $T = 365$ since $n = 365$
- $a_T = 3 \times 7 = 21$

  Note: a_T is set to the multiple of seasonality as suggested in Bijral (2019). We take a_T to be three times the weekly seasonality because this gives an equal number of O' blocks and O'' blocks later. This multiple of seasonality is chosen only for convenience since it was proven in our simulated data experiment that the value of a_T does not matter much to the final TPR and FPR.

- $q_{max} = 0$

  Note: the max endogenous lags to select from is 0 because all predictors are exogenous, i.e., no component of the response vector is included in the matrix of predictors.

- $\mu_T = \lfloor T/2a_T \rfloor = \lfloor 365/(2 \times 21) \rfloor = 8$. Therefore there are 8 odd blocks in total, comprising of 4 O' blocks and 4 O'' blocks of length $a_T = 21$. The length of the aggregated O' block is $4 \times 21 = 84$ measurements.

  Note: The floor operator is used for μ_T because using the multiple of seasonality to set a_T will not get an integer for μ_T. As we divide the data set into pairs of odd and even blocks, the last pair of blocks will not have enough observations to be equal in length to the other pairs of blocks. Therefore, we discard the last pair of odd and even blocks using the floor operator.

The other BPA parameters settings are:

- $q = \lfloor 0.4 \times p \rfloor$
- $\phi = 0.8$
- $B = 50$

We chose the above parameter values because it was suggested as the default in Bijral (2019), and those values are proven to yield good TPR and FPR in our simulated data experiment.

4.3.3 Performance evaluation

The performance evaluation is done by using the same methodology as in Li et al. (2019).

We use a frame of length $3n$ from the data in each run, corresponding to three consecutive years of measurements. The first two years are used to train the BPA predictor and the last year to evaluate it. To initialise the experiment, we take t_0 to be 30/04/2008 and let t_1 be the last day of the second year so $t_1 = t_0 + 2n - 1$. Therefore the training response vector is $\mathbf{y}_{\text{site}}(t_1)$ and the training predictors matrix is $\mathbf{X}^{\text{SC}}_{\text{site}}(t_1)$, where $(\cdot)^{\text{sc}}$ is used to distinguish scenario (a) and (b). The resulting vector of linear parameters is produced by applying BPA for predictor selection to the training predictors matrix and then using least squares to find the coefficients for each predictor. For the r-th run, the vector of coefficients is further used together with the testing predictors matrix in the third year $\mathbf{X}^{\text{SC}}_{\text{site}}(t_1 + 8r + n)$ in order to produce the estimate $\hat{\mathbf{y}}_{\text{site}}(t_1 + 8r + n)$. The procedure is applied for $\text{N}_{TR} = 100$ runs, so $r \in \{1, 2, ..., 100\}$. This means that the last day of the second year is forwarded by 8 days by each N_{TR} run.

The normalized mean square error (NMSE) is computed by applying the following formula for each site:

$$\text{NMSE}^{\text{SC}}_{\text{site}} = \frac{\sum_{r=1}^{\text{N}_{TR}} ||\exp[\mathbf{y}_{\text{site}}(t_r + n)] - \exp[\hat{\mathbf{y}}^{\text{SC}}_{\text{site}}(t_r + n)]||^2}{\sum_{r=1}^{N_{TR}} ||\exp[\mathbf{y}_{\text{site}}(t_r + n)]||^2},$$

where applying exp() to the vector means we exponentiate each entry of that vector. Exponentiation is used to back-transform the log data to the normal scale for interpretation.

4.4 Experimental results for air pollution data

The values of NMSE computed by applying BPA are outlined in Table 4.2. The result shows that most NMSEs for BPA are smaller than the smallest NMSEs by the other predictive models with different stopping rules for matching pursuit algorithm (MPA) in Li et al. (2019), which suggests BPA outperforms those models.

Notably, BPA outperforms all other algorithms in the FullSet scenario. This supports the argument that BPA is an exceptionally robust algorithm in the high-dimensional FullSet scenario. This observation is perfectly in line with the theoretical grounds on which BPA is a suitable methodology to use when the data has a high-dimensional problem Bijral (2019).

It is also observed that the ConSet scenario always has a smaller NMSE. This is expected because in ConSet, we use prior knowledge to pre-select the useful predictors before feeding the measurements to the predictive models. Therefore, it is reasonable that all models perform better in ConSet than in FullSet.

	Patumahoe(PA)		Penrose(PE)		Takapuna(TA)		Whangaparaoa(WH)	
	FullSet	ConSet	FullSet	ConSet	FullSet	ConSet	FullSet	ConSet
BPA NMSE	**5.61%**	**5.59%**	**3.75%**	**3.60%**	**6.05%**	**5.73%**	**3.17%**	**3.15%**
Smallest NMSE	6.14%	5.42%	4.03%	3.63%	6.19%	5.08%	3.35%	3.13%
Stopping Rule	EgMDL_1^*	CV	ESC_1^*	MMLG_1	EgMDL_1^*	AIC_c	EBIC^*	KIC

Table 4.2: NMSE of predictive models for air pollution data. The values of NMSE are shown in the columns. BPA is compared against the smallest NMSE by the other predictive models in Li et al. (2019). Green highlights the NMSE in BPA that are less than the smallest NMSE by the other predictive models. The stopping rules for matching pursuit algorithm (MPA) are briefly explained in Appendix A.

To investigate further the difference in performance for BPA between FullSet and ConSet, we consider the effect of q and ϕ on the number of predictors selected.

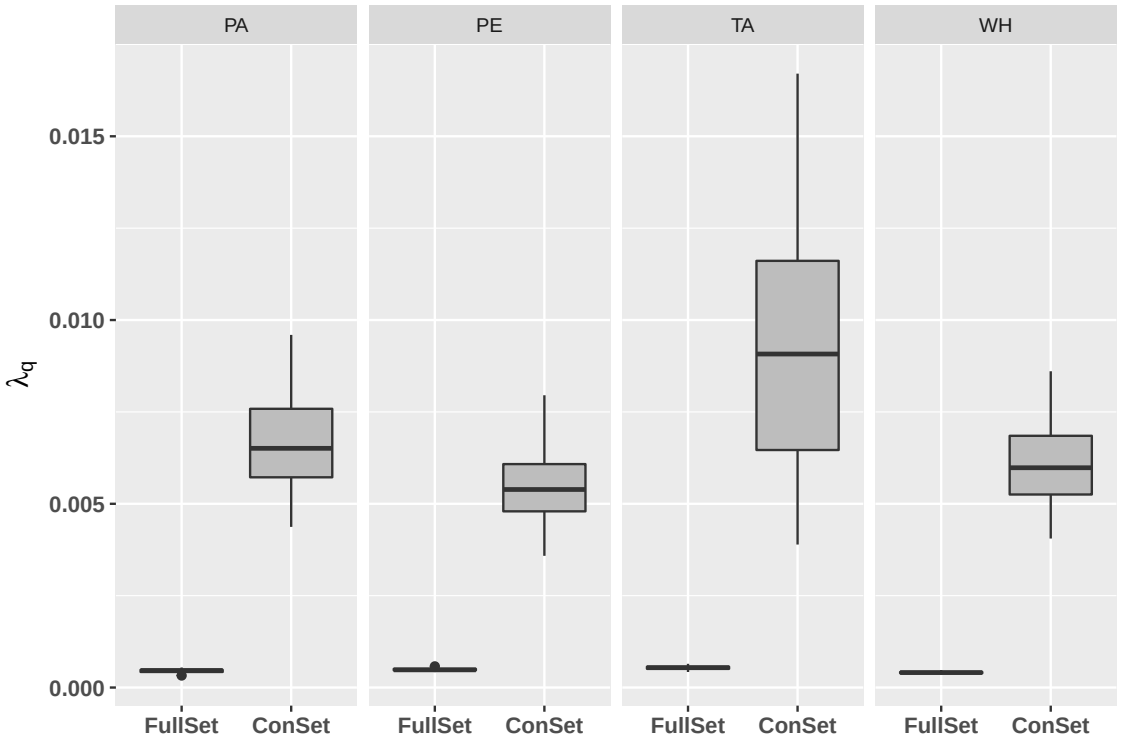


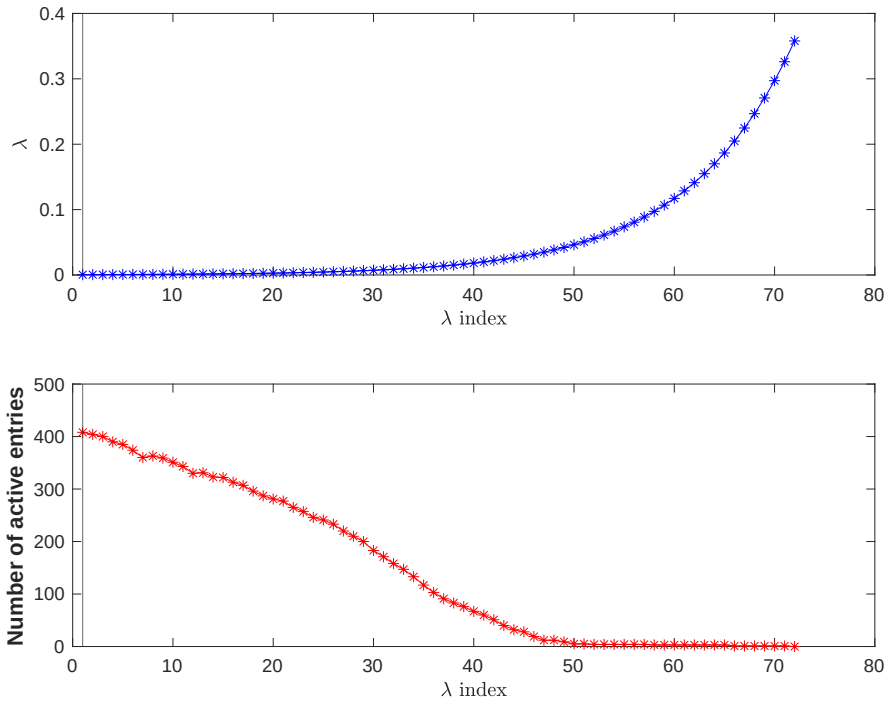
Figure 4.3: Boxplots for the values of λ_q selected in $N_{TR} = 100$ runs for each scenario of each site.

Selection of λ_q in different settings In Figure 4.3, λ_q is the minimum λ selected for at most q active entries, see Algorithm 3. We observed that FullSet has generally smaller λ_q than ConSet. This is expected because FullSet has many useless predictors, so even a small λ_q penalty can effectively eliminate a large number of weak predictors. Whereas most of the predictors in ConSet are useful, λ_q needs to be larger to reduce the number of predictors to only 40% of the total number of predictors.

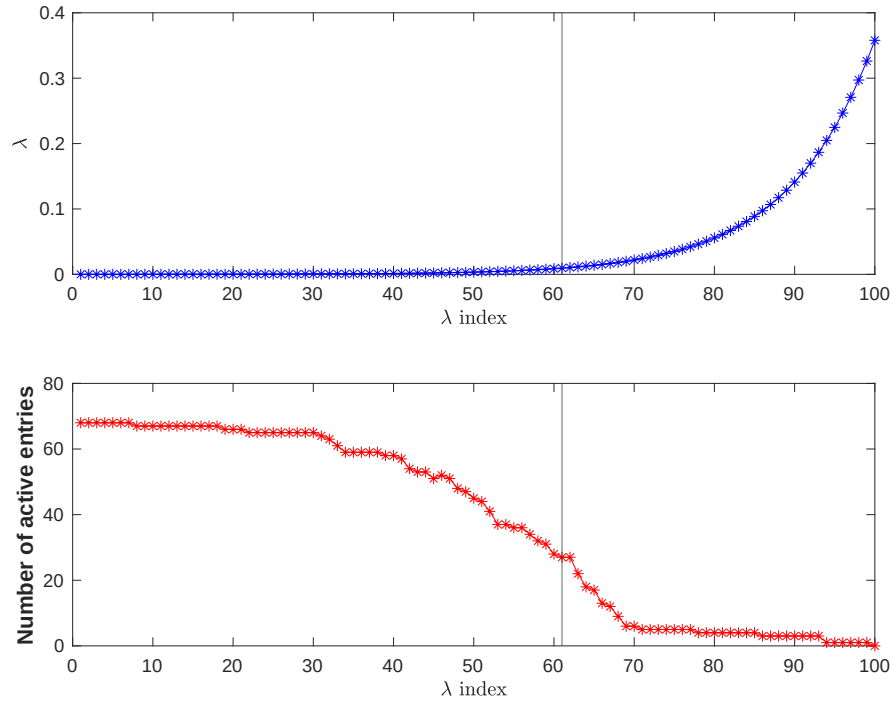
In Figure 4.4, we observed that with the FullSet scenario, it selected the smallest λ value of using the ratio of the smallest to the largest λ values $1e^{-4}$ (default minimum ratio in the MATLAB `lasso` function). This indicates that even a minimal λ value can provide enough regularization for variable selection to return only $q = 1464 * 0.4 = 585$ active entries. This is not surprising because most predictors in FullSet are not useful, and they do not reduce enough RSS for the Lasso regression to be able to afford the cost of keeping those predictors, as referred to in the Lasso regression in Equation (2.1).

Notably, the ConSet scenario has a more extensive range of values for λ_q, and it maps with what we observed for the high signal simulated data set SNR=20dB.

We also observed the cardinality of the Λ-set for FullSet is not the default 100. It is the case when the residual error of the fits drops below a threshold fraction of the variance of the response data $\mathbf{Y}$ so that the `lasso` function can return fewer than 100 candidates.



(a) Possible λ values for $N_{TR} = 1$ on FullSet



(b) Possible λ values for $N_{TR} = 1$ on ConSet

Figure 4.4: Range of λ values generated by MATLAB Lasso function

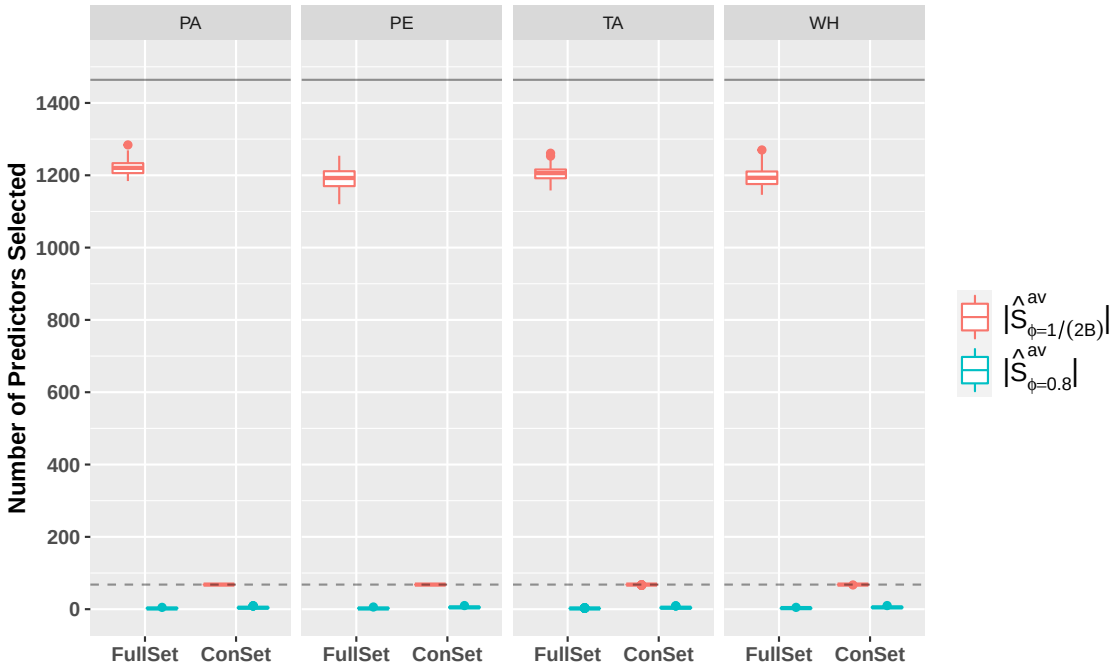


Figure 4.5: Number of predictors (p^*) before applying and after applying $\phi = 0.8$ for air pollution data. The red boxes $|\hat{S}^{av}_{\phi=1/(2B)}|$, where $|\cdot|$ denotes cardinality, are the number of predictors that are selected at least once in any block. The blue boxes $|\hat{S}^{av}_{\phi=0.8}|$ are the number of predictors that are selected at least 80% of the two aggregated blocks for B iterations, i.e., at least selected $0.8 * 2 * 50 = 80$ times by O' and O'' blocks combined in 50 iterations. The solid black line represents the original amount of predictors for FullSet ($p = 1464$), and the dash line represents the original amount of predictors for ConSet ($p = 68$). The first layer of predictor selection by λ_q happens when the number of predictors selected falls from the black line to the red boxes. The second layer of predictor selection by $\phi = 0.8$ happens when the number of predictors selected falls from the red boxes to the blue boxes. The number of final predictors selected for each scenario of each site, i.e., the blue boxes, are shown on a smaller scale in Figure 4.6.

Effect of ϕ in different settings In Figure 4.5, the number of predictors selected before and after applying the first layer of predictor selection λ_q and the second layer of predictor selection $\hat{S}^{av} = \{k : \prod_B^{av}(k) \geq \phi\}$ are displayed on a side-by-side boxplot for each scenario of each site, see Algorithm 3 for the definition of λ_q and ϕ. We observed large drops for FullSet after both λ_q and $\phi = 0.8$. This is expected because FullSet have many useless predictors, and Lasso (with parameter λ_q) can easily shrink the coefficient of those useless predictors to zero. A few blocks select some leftover weak predictors that pass the first layer of selection, but they do not have strong enough predictive power to be selected in most of the blocks to be in the final model. The second layer of predictor selection criterion, i.e., $\phi = 0.8$, effectively excludes the weak predictors from FullSet. Based on prior knowledge, we suspect that those predictors that are in the FullSet but not in the ConSet are weak predictors and should not be selected. In ConSet, it is less likely to select random predictors since the predictors are deemed to be useful before feeding them to the model. The predictive power for ConSet predictors is strong, which will decrease the RSS error enough to compensate for the penalty cost; see again Equation (2.1). Thus, λ_q is higher for ConSet. The regularization by λ_q for ConSet does not have much effect on reducing the predictors. This is because the Lasso can only shrink the coefficient of predictors to zero when the predictor is completely useless. However, all predictors in ConSet are useful to some extent, so Lasso cannot effectively regularize predictors with some predictive power. Therefore, it is not surprising that most predictors are selected at least once.

Therefore in the ConSet scenario, the second layer criterion by $\phi = 0.8$ reduces a large proportion of predictors. For both FullSet and ConSet scenarios, the predictor selection happens mainly at the second layer. The first layer of regularization using λ_q does not seem to discard the weaker predictors.

Significance of final number of predictors selected in different settings In Figure 4.6, which emphasizes the final predictors of each scenario for the different sites, we observed that the ConSet have more final predictors than FullSet. This suggests that BPA is good at keeping the most useful predictors from a set of all useful predictors in ConSet which eventually got a small NMSE shown in Table 4.2. However, BPA is not that good at selecting predictors out of lots of predictors, and of which many are not useful. Since the NMSE for FullSet is always higher than ConSet, and the ConSet predictors are a subset of the FullSet predictors, the prediction for FullSet will be improved if BPA could have selected those extra predictors selected in the ConSet scenario. That is, if the same set of predictors is chosen in the FullSet scenario as in the ConSet scenario, the NMSE for FullSet will improve. Nevertheless, BPA still outperforms the various stopping rules for MPA in the FullSet scenario.

Predictors selected in different settings Table 4.3 shows the top 3 predictors selected out of the 100 N_{TR} runs in each scenario for the different sites. We observed that PM_{10} from the same site on the present day is always the top voted predictor and is always selected in each of the N_{TR} runs. These predictors are denoted $[\boldsymbol{\Xi}_{\text{site}}(t)]_{:1}$ in Figure 4.3. The predictor selection makes sense because $PM_{2.5}$ is a subset of PM_{10}, so $PM_{2.5}$ inevitably correlates with the PM_{10} measurement on the same day at the same site. Then the $PM_{2.5}$

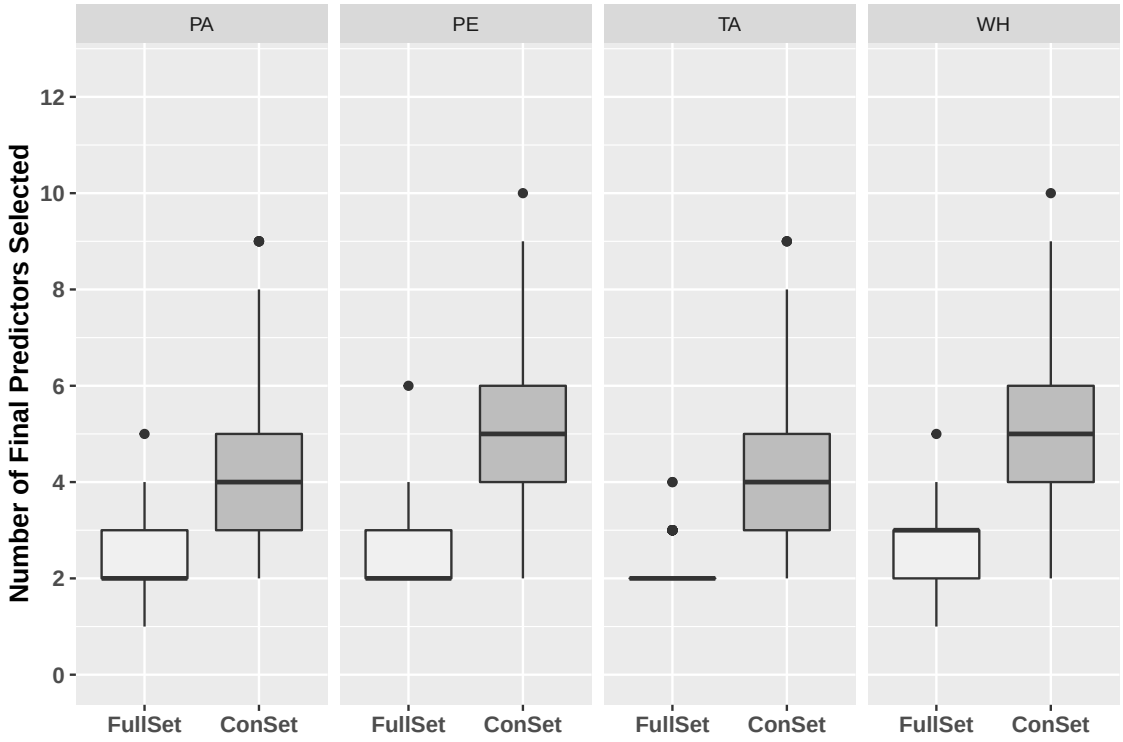


Figure 4.6: Boxplots for the number of final predictors selected in $N_{TR} = 100$ runs for each scenario of each site.

measurements on the same day for the other sites are usually the second and third most voted predictors. These predictors are denoted $[\boldsymbol{\Theta}_{\text{site}}(t)]_{:1}$ in Figure 4.3. This selection also makes sense because the $PM_{2.5}$ measurements in the site that we want to predict will most likely correlate with the $PM_{2.5}$ measurement on the same day in the other sites in the Auckland region. The only exception is for the ConSet of PE, where the third most voted predictor is the $PM_{2.5}$ measurement of TA one day before the present day. This predictor is denoted $[\boldsymbol{\Theta}_{\text{TA}}(t)]_{:2}$ in Figure 4.3. This selection is reasonable because PE and TA are close to each other and may have similar $PM_{2.5}$ measurements. This analysis is only descriptive and does not suggest any causal relationship between the predictors and the $PM_{2.5}$ measurement.

		I	**II**	**III**
$\mathbf{y}_{\mathrm{PA}}(t) = [\boldsymbol{\Theta}_{\mathrm{PA}}(t)]_{:1}$	FullSet	$[\boldsymbol{\Xi}_{\mathrm{PA}}(t)]_{:1}$ (100)	$[\boldsymbol{\Theta}_{\mathrm{PE}}(t)]_{:1}$ (82)	$[\boldsymbol{\Theta}_{\mathrm{TA}}(t)]_{:1}$ (50)
	ConSet	$[\boldsymbol{\Xi}_{\mathrm{PA}}(t)]_{:1}$ (100)	$[\boldsymbol{\Theta}_{\mathrm{TA}}(t)]_{:1}$ (59)	$[\boldsymbol{\Theta}_{\mathrm{PE}}(t)]_{:1}$ (55)
$\mathbf{y}_{\mathrm{PE}}(t) = [\boldsymbol{\Theta}_{\mathrm{PE}}(t)]_{:1}$	FullSet	$[\boldsymbol{\Xi}_{\mathrm{PE}}(t)]_{:1}$ (100)	$[\boldsymbol{\Theta}_{\mathrm{TA}}(t)]_{:1}$ (100)	$[\boldsymbol{\Theta}_{\mathrm{PA}}(t)]_{:1}$ (22)
	ConSet	$[\boldsymbol{\Xi}_{\mathrm{PE}}(t)]_{:1}$ (100)	$[\boldsymbol{\Theta}_{\mathrm{TA}}(t)]_{:1}$ (100)	$[\boldsymbol{\Theta}_{\mathrm{TA}}(t)]_{:2}$ (29)
$\mathbf{y}_{\mathrm{TA}}(t) = [\boldsymbol{\Theta}_{\mathrm{TA}}(t)]_{:1}$	FullSet	$[\boldsymbol{\Xi}_{\mathrm{TA}}(t)]_{:1}$ (100)	$[\boldsymbol{\Theta}_{\mathrm{PE}}(t)]_{:1}$ (100)	$[\boldsymbol{\Theta}_{\mathrm{PA}}(t)]_{:1}$ (13)
	ConSet	$[\boldsymbol{\Xi}_{\mathrm{TA}}(t)]_{:1}$ (100)	$[\boldsymbol{\Theta}_{\mathrm{PE}}(t)]_{:1}$ (100)	$[\boldsymbol{\Theta}_{\mathrm{WH}}(t)]_{:1}$ (32)
$\mathbf{y}_{\mathrm{WH}}(t) = [\boldsymbol{\Theta}_{\mathrm{WH}}(t)]_{:1}$	FullSet	$[\boldsymbol{\Xi}_{\mathrm{WH}}(t)]_{:1}$ (100)	$[\boldsymbol{\Theta}_{\mathrm{PA}}(t)]_{:1}$ (88)	$[\boldsymbol{\Theta}_{\mathrm{PE}}(t)]_{:1}$ (55)
	ConSet	$[\boldsymbol{\Xi}_{\mathrm{WH}}(t)]_{:1}$ (100)	$[\boldsymbol{\Theta}_{\mathrm{PA}}(t)]_{:1}$ (88)	$[\boldsymbol{\Theta}_{\mathrm{PE}}(t)]_{:1}$ (66)

Table 4.3: Final predictors ranked by the number of times they get selected in the 100 runs. The numbers in brackets coloured by blue are the number of times the predictor has been selected in the $\mathrm{N}_{TR} = 100$ runs. The columns **I, II, III** are the top three most selected predictors ranked from high to low.

Chapter 5

Final remarks

Summary and conclusions In this work, we have investigated the time series predictor selection scheme BPA proposed by Bijral (2019) under the sub-sampling based selection framework. We analysed the applicability of BPA to simulated and real time series data sets and test on various noise levels and parameter values. We found BPA is quite robust against noise, which is aligned with the findings from previous work by Bijral (2019). From the experiments on the simulated data set, we have investigated the different parameter settings for BPA and discovered that parameters q and ϕ are the most influential parameters for BPA's performance. Parameter q should be set at a medium level to be loose and allow predictors to pass through the first layer, whereas ϕ should be set at a relatively high level for effective predictor selection. We used our findings from the simulated data experiment on the optimal parameter values and tested the BPA model on the real air pollution data. BPA is performing very well in the experiment on real data, where it beats all other greedy algorithms studied in Li et al. (2019) for the FullSet scenario.

In the future, it might also be interesting to compare BPA with other existing time series data predictor selection techniques. An area where BPA may be further optimised is to use different base methods, like Ridge Regression, Elastic Net or Adaptive Lasso.

The results are pretty promising for BPA in this work. It might be interesting to evaluate BPA's performance on data sets in the other discipline, e.g., financial, economics, biology, ecology, etc. The applicability of this relatively new BPA method to the various types of problems can also be explored further.

Author's contribution A previous version of the simulated data code (for Chapter 3) was provided by the supervisor. The author carefully checked the calculations and modified the code to create Scenario I and Scenario II simulated data. The author proposed the variant BPA-m for BPA. The MATLAB implementation of the BPA and BPA-m algorithms for the experiments in Chapter 3 and 4 were written by the author and corrected by the supervisor. The air pollution data set and the MATLAB code for the experiment settings in Chapter 4 were provided by the supervisor. The author integrated the BPA predictor selection function into those real data experiment functions. The author ran

all the experiments locally and using high-performance computing provided by NeSI and produced all the figures and tables, which are included in the report using R, MATLAB, LaTeX, draw.io. The author also wrote the original version of this report, which later on was modified following the supervisor's comments.

Appendix A

Stopping rules for Matching Pursuit Algorithm (MPA)

Most of the the information theoretic criteria have been introduced in the previous literature for the case of the linear regression when the number of measurements is larger than the number of predictors. For the evaluation of the criteria a least-squares problem should be solved. As MPA is designed to be a low-complexity algorithm, it does not solve a least-squares problem. Hence, the formulas of the criteria should be altered to be compatible with MPA.

- EgMDL_1^*

  Modified Extended Generalized Minimum Description (alteration 1), originally introduced in Li et al. (2017), modified in Li et al. (2019).

- CV

  Leave-one-out cross-validation, see more in Sancetta (2016).

- ESC_1^*

  Modified Extended Stochastic Complexity (alteration 1), originally introduced in Li et al. (2017), modified in Li et al. (2019).

- MMLG_1

  Minimum Message Length (alteration 1), see more in Schmidt and Makalic (2009).

- AIC_c

  Akaike Information Criterion (corrected), see more in Bühlmann (2006).

- EBIC^*

  Modified Extended Bayesian Information Criterion, originally introduced in Bühlmann and Van De Geer (2011), modified in Li et al. (2019).

- KIC

  Kullback Information Criterion, see more in Cavanaugh (1999).

Appendix B

Supplementary experiment results

B.1 For BPA in Scenario I

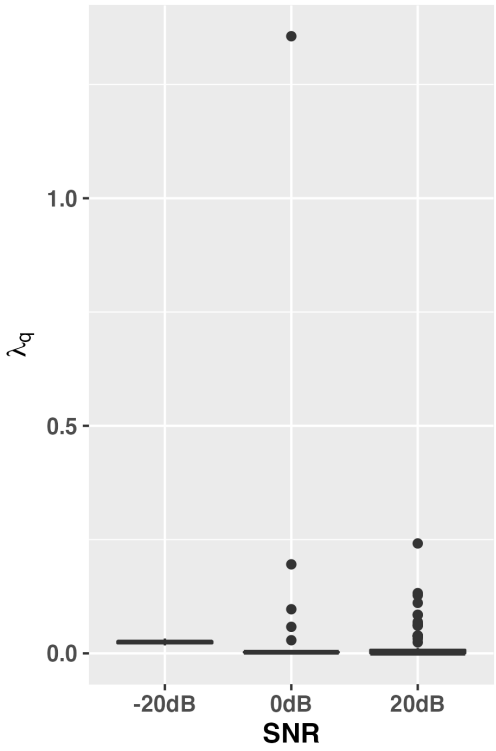


Figure B.1: λ_q boxplots for different SNR level (outlier included) in Scenario I. Parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 50$.

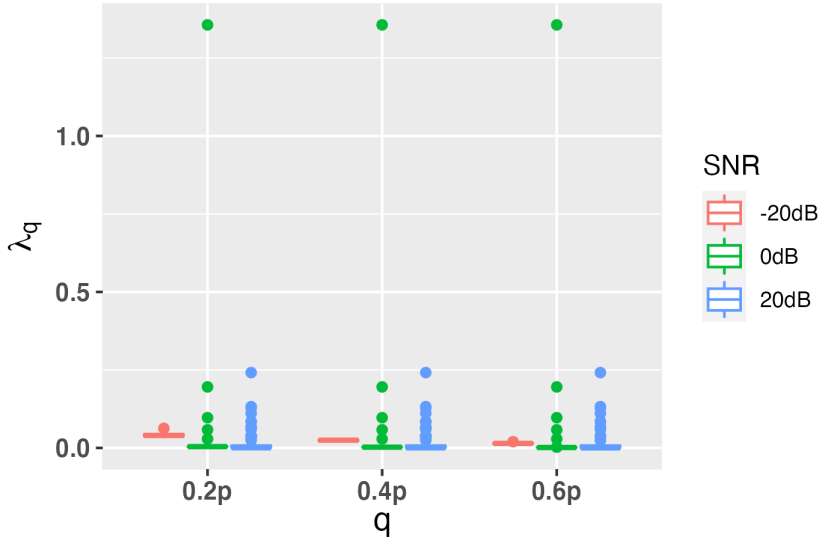


Figure B.2: λ_q boxplots for different q (outlier included) in Scenario I. Parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 50$.

B.2 For BPA-m in Scenario I

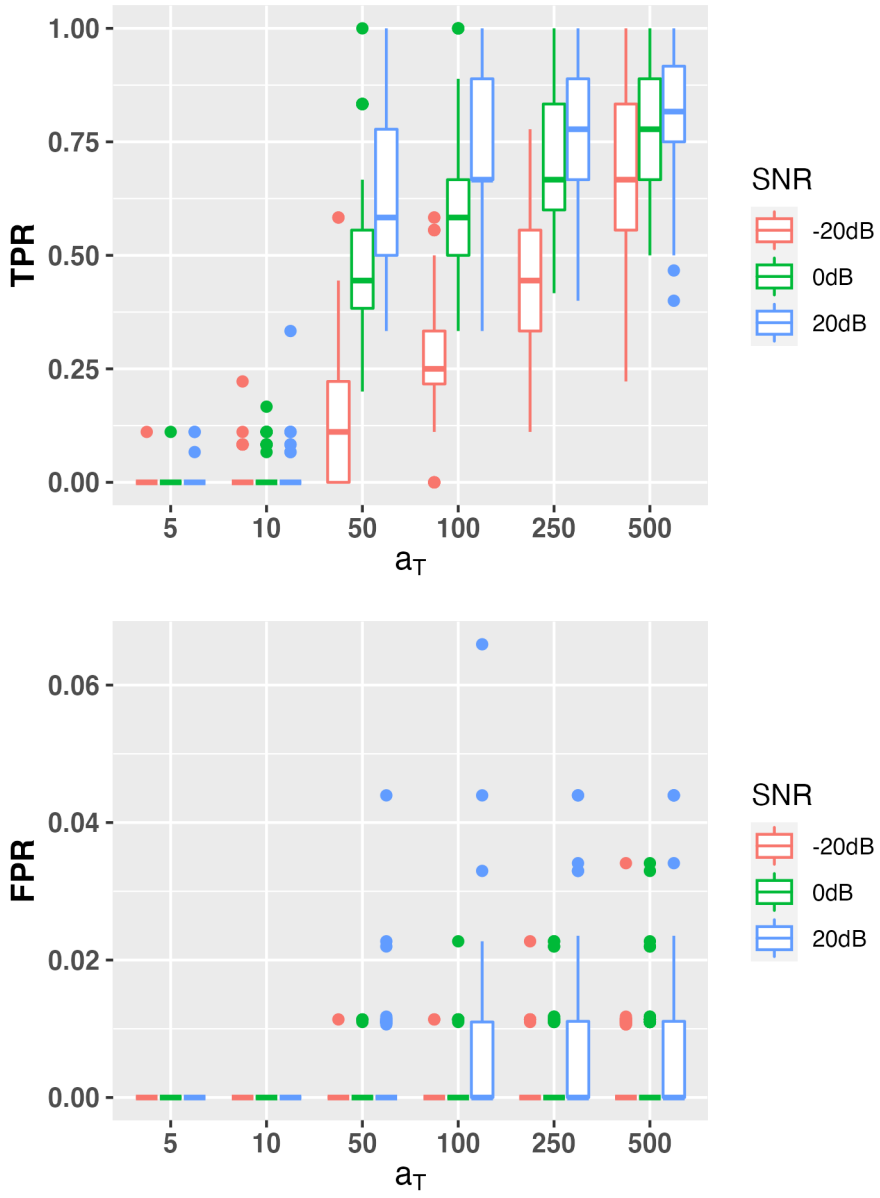


Figure B.3: TPR/FPR for BPA under different a_T values. Other parameters are fixed at $q = 0.4$, $\phi = 0.8$, $B = 1$.

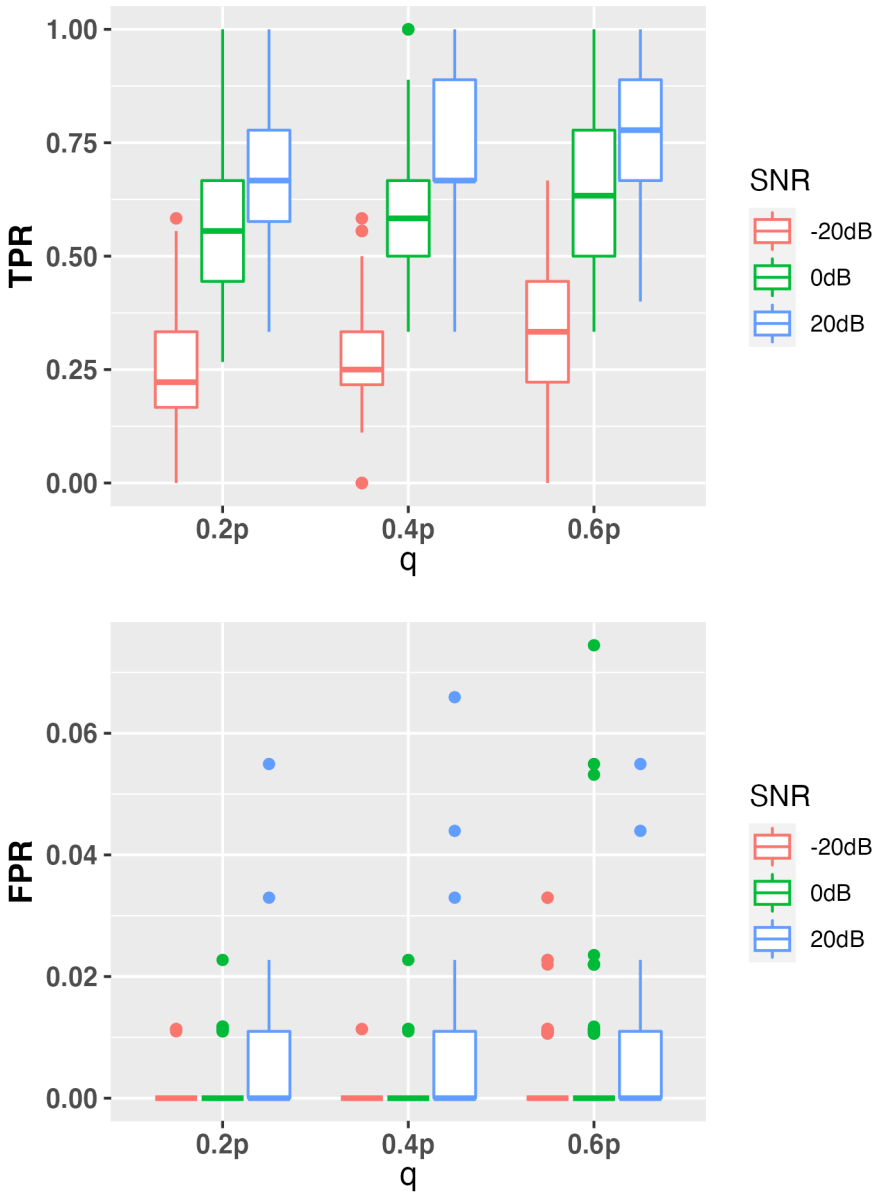


Figure B.4: TPR/FPR for BPA under different q values. Other parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 1$.

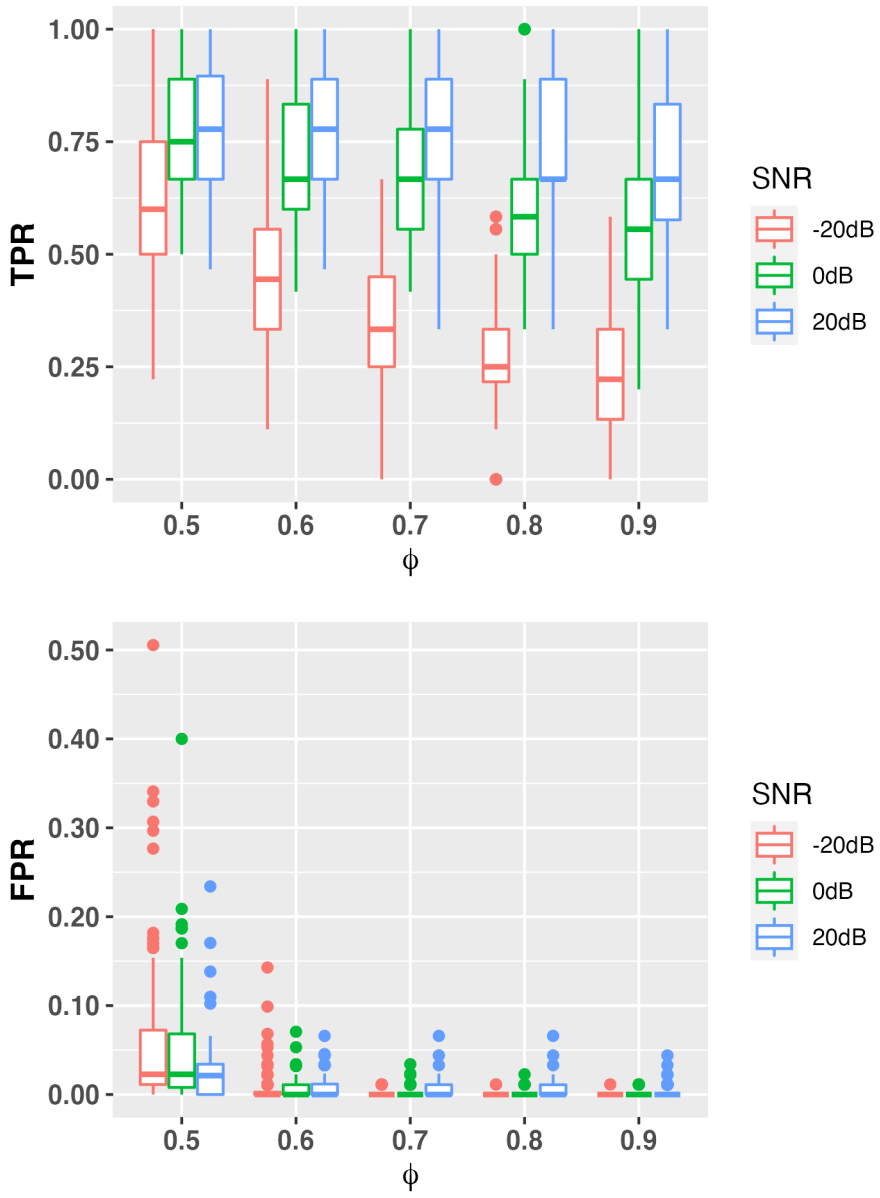
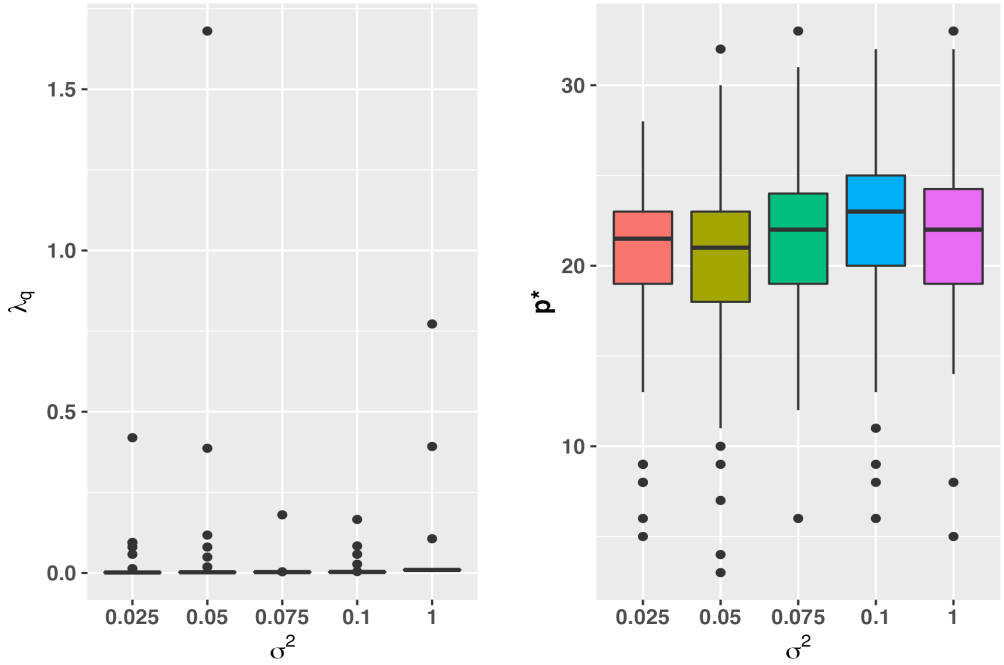


Figure B.5: TPR/FPR for BPA under different ϕ values. Other parameters are fixed at $a_T = 100$, $q = 0.4$, $B = 1$.

B.3 For BPA in Scenario II



(a) λ_q boxplots for different σ^2 levels (b) p^* boxplots for different σ^2 levels

Figure B.6: λ_q selection and number of final predictors for BPA under different noise levels in Scenario II. Parameters are fixed at $a_T = 100$, $q = 0.8$, $\phi = 0.8$.

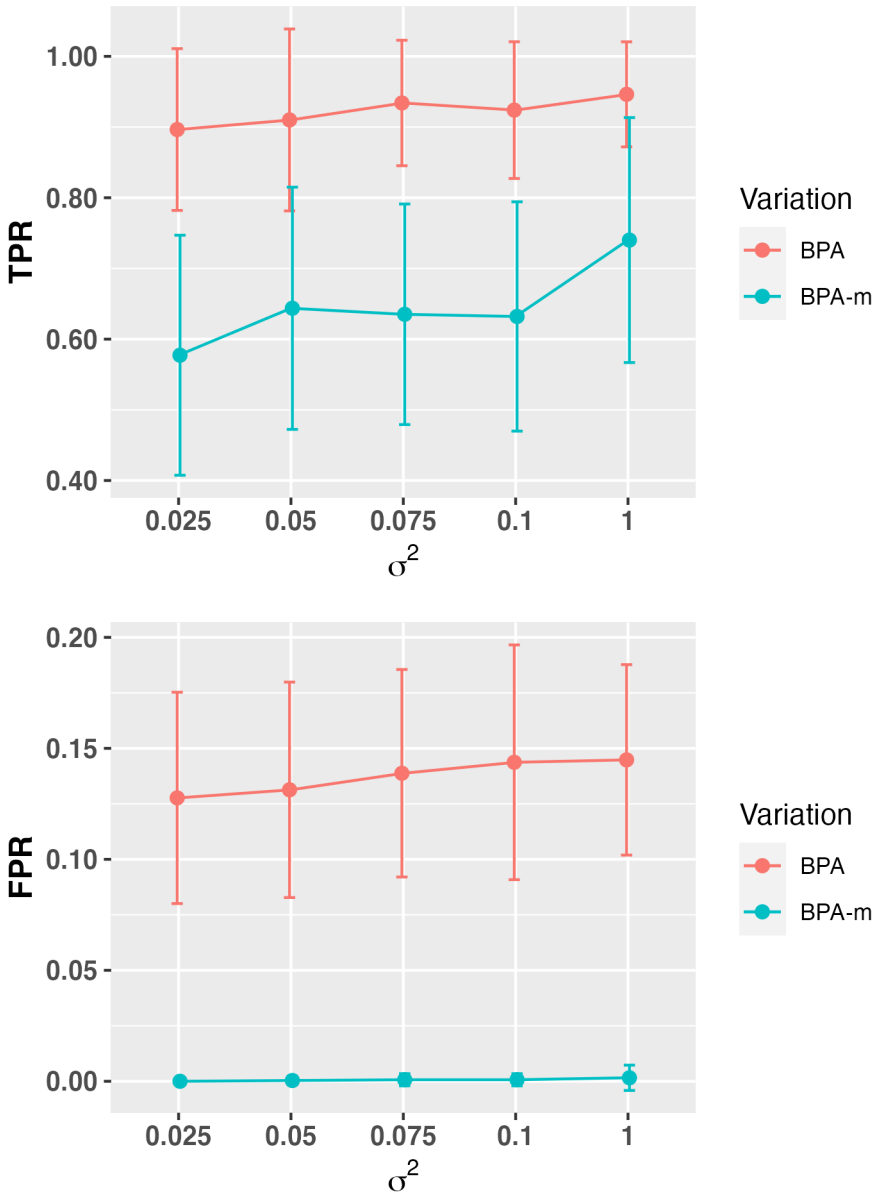


Figure B.7: Error bar TPR/FPR for BPA and BPA-m under different σ^2 values in Scenario II. Other parameters are fixed at $a_T = 100$, $q = 0.4p$, $\phi = 0.8$.

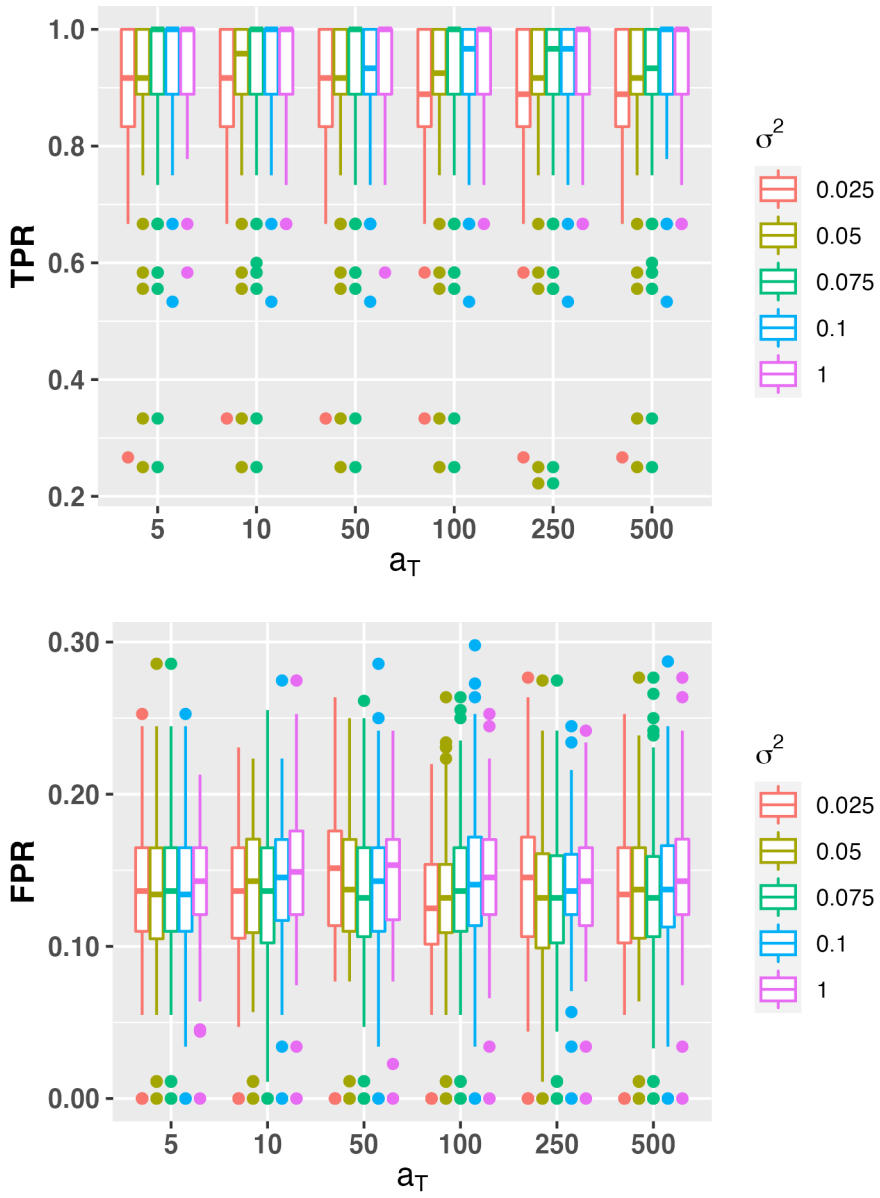


Figure B.8: TPR/FPR for BPA under different a_T values. Other parameters are fixed at $q = 0.4$, $\phi = 0.8$, $B = 50$.

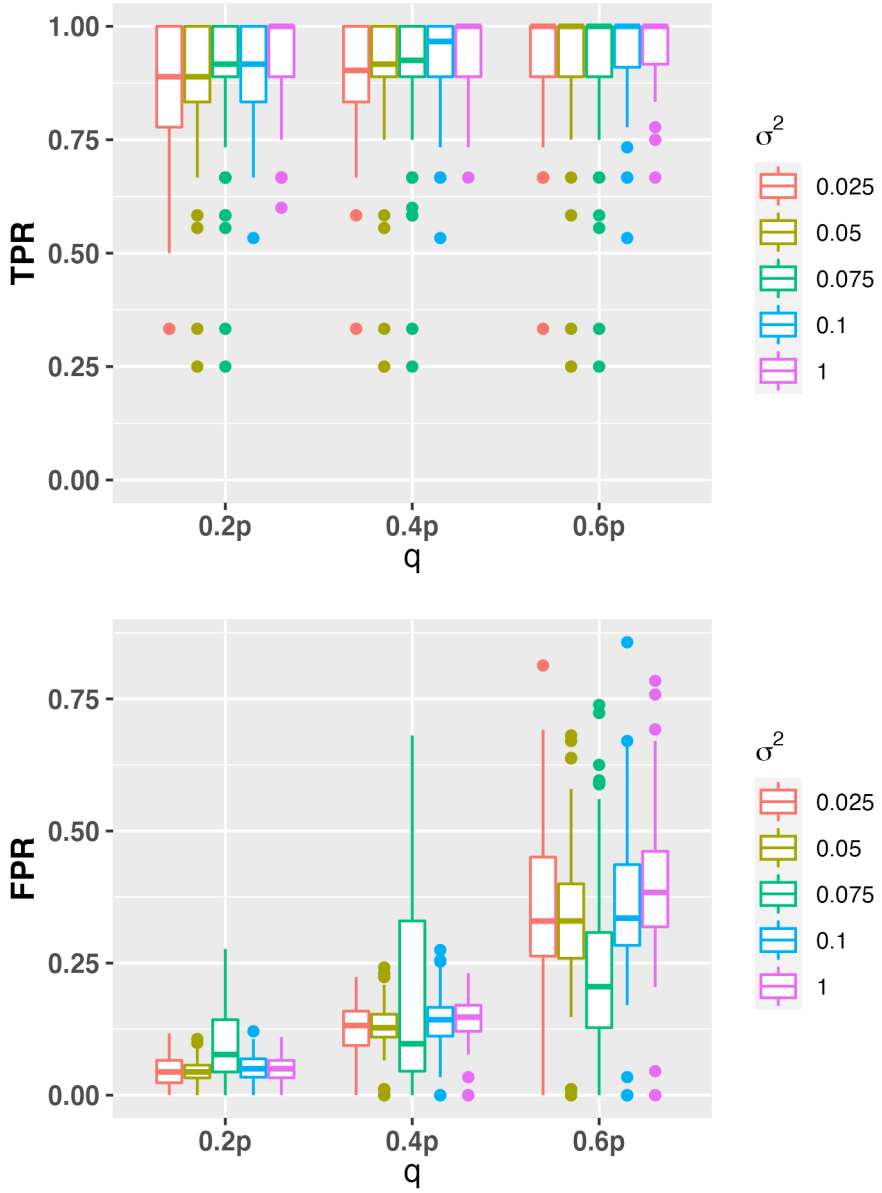


Figure B.9: TPR/FPR for BPA under different q values. Other parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 50$.

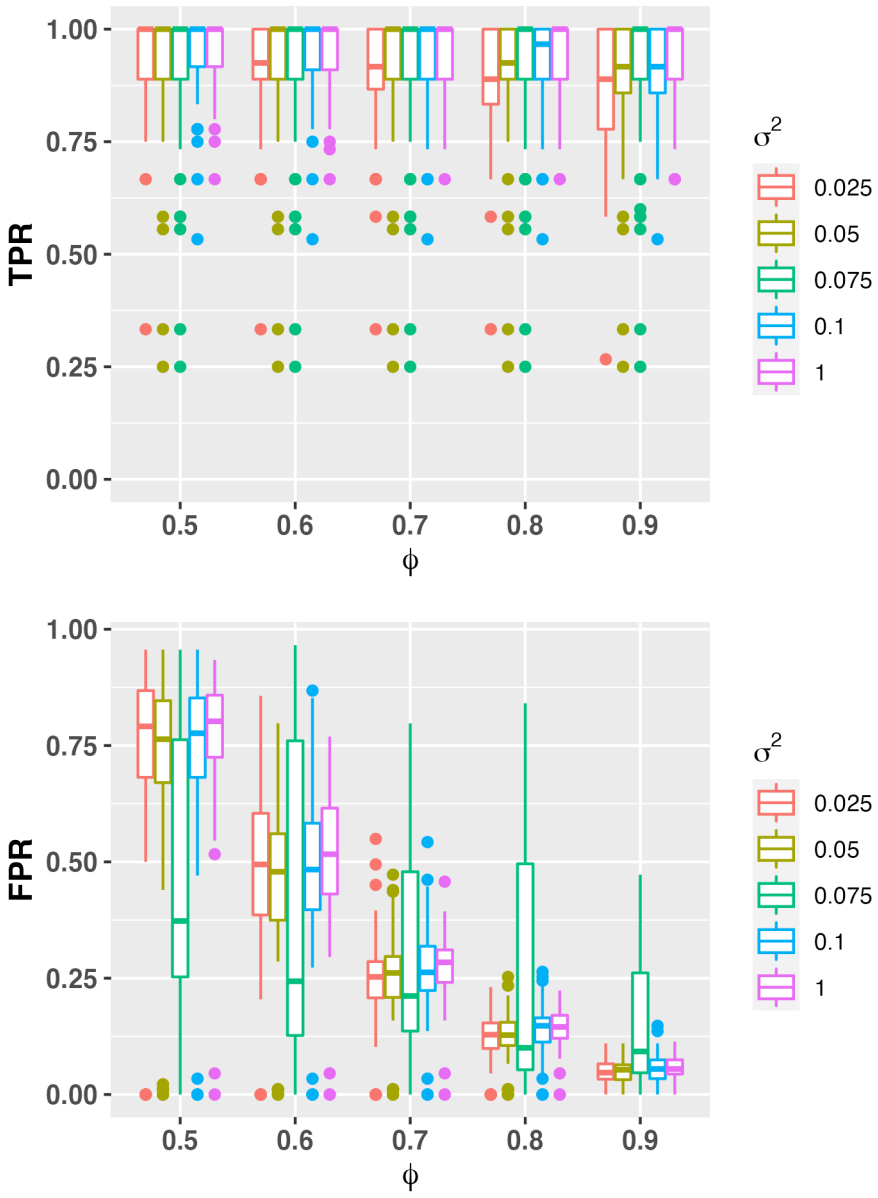


Figure B.10: TPR/FPR for BPA under different ϕ values. Other parameters are fixed at $a_T = 100$, $q = 0.4$, $B = 50$.

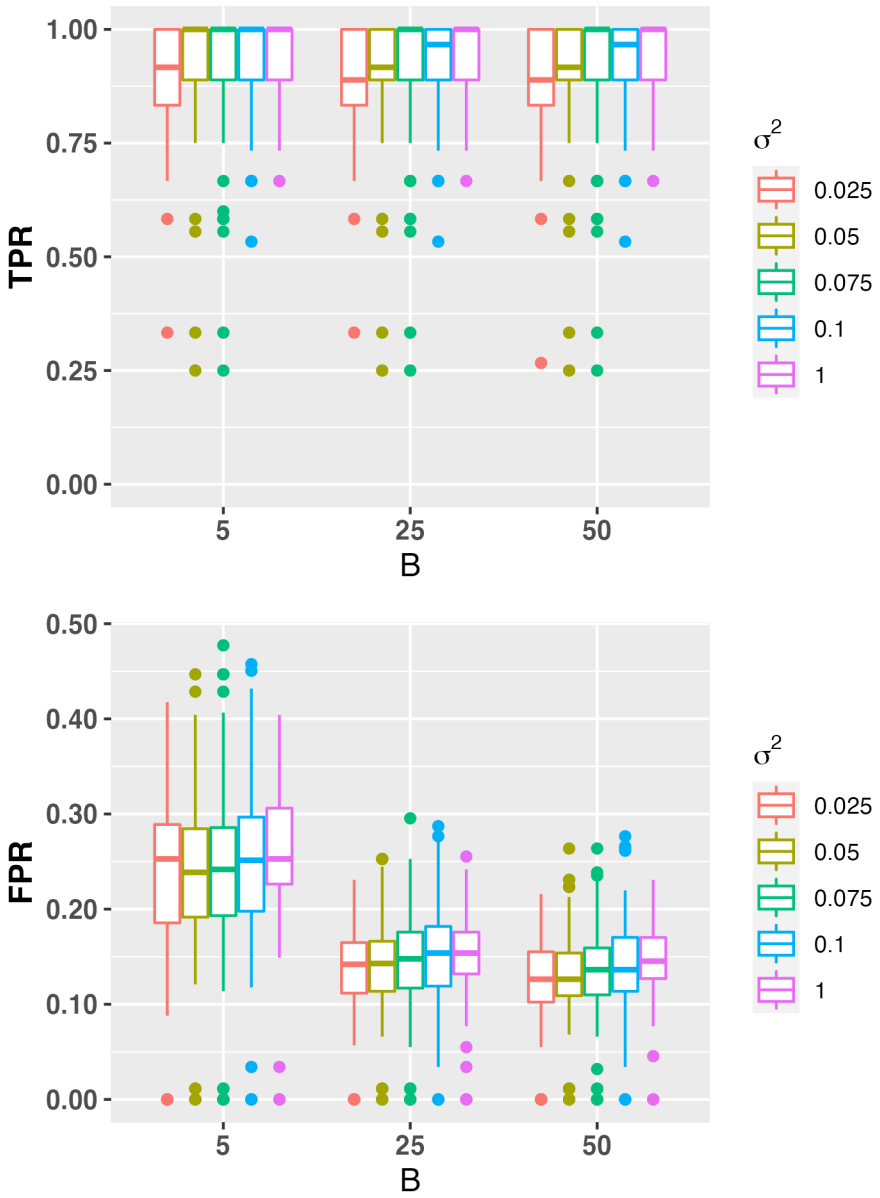


Figure B.11: TPR/FPR for BPA under different B values. Other parameters are fixed at $a_T = 100$, $q = 0.4$, $\phi = 0.8$.

B.4 For BPA-m in Scenario II

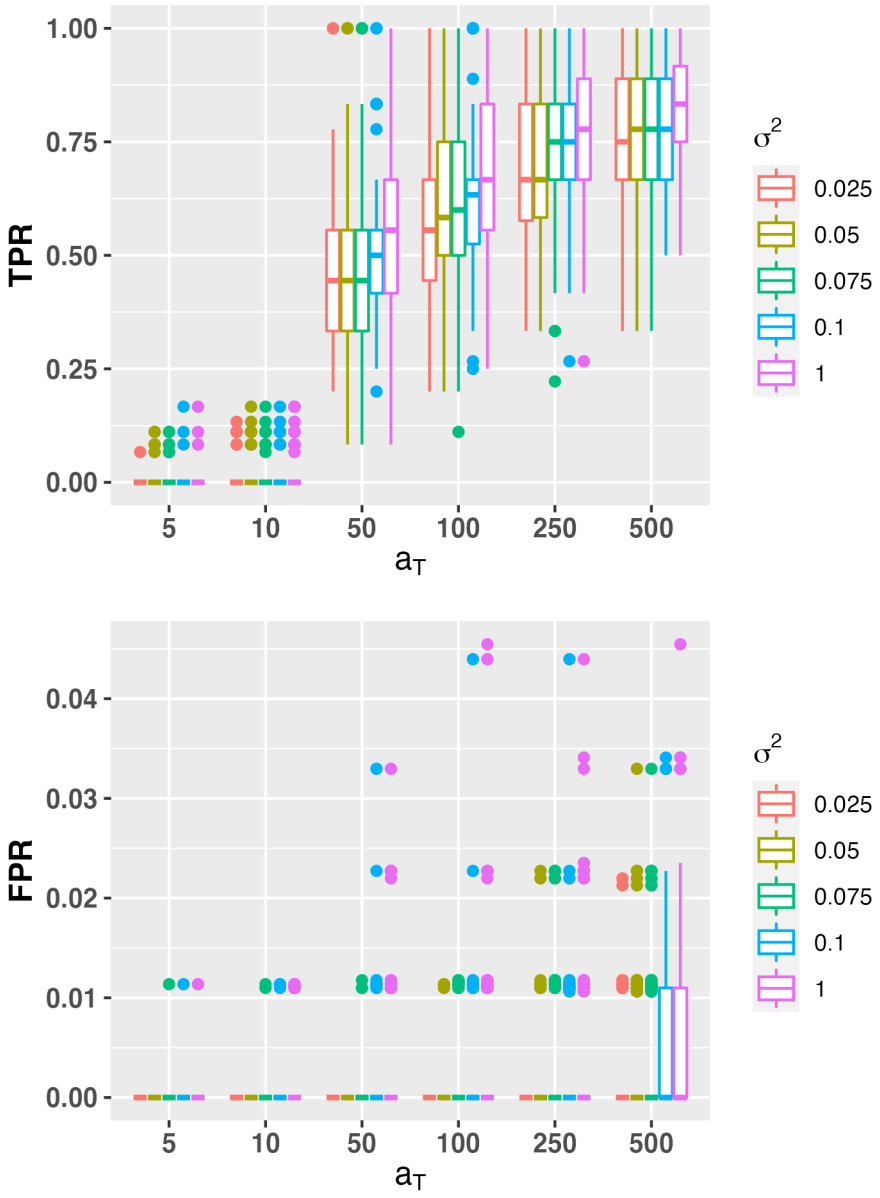


Figure B.12: TPR/FPR for BPA under different a_T values. Other parameters are fixed at $q = 0.4$, $\phi = 0.8$, $B = 1$.

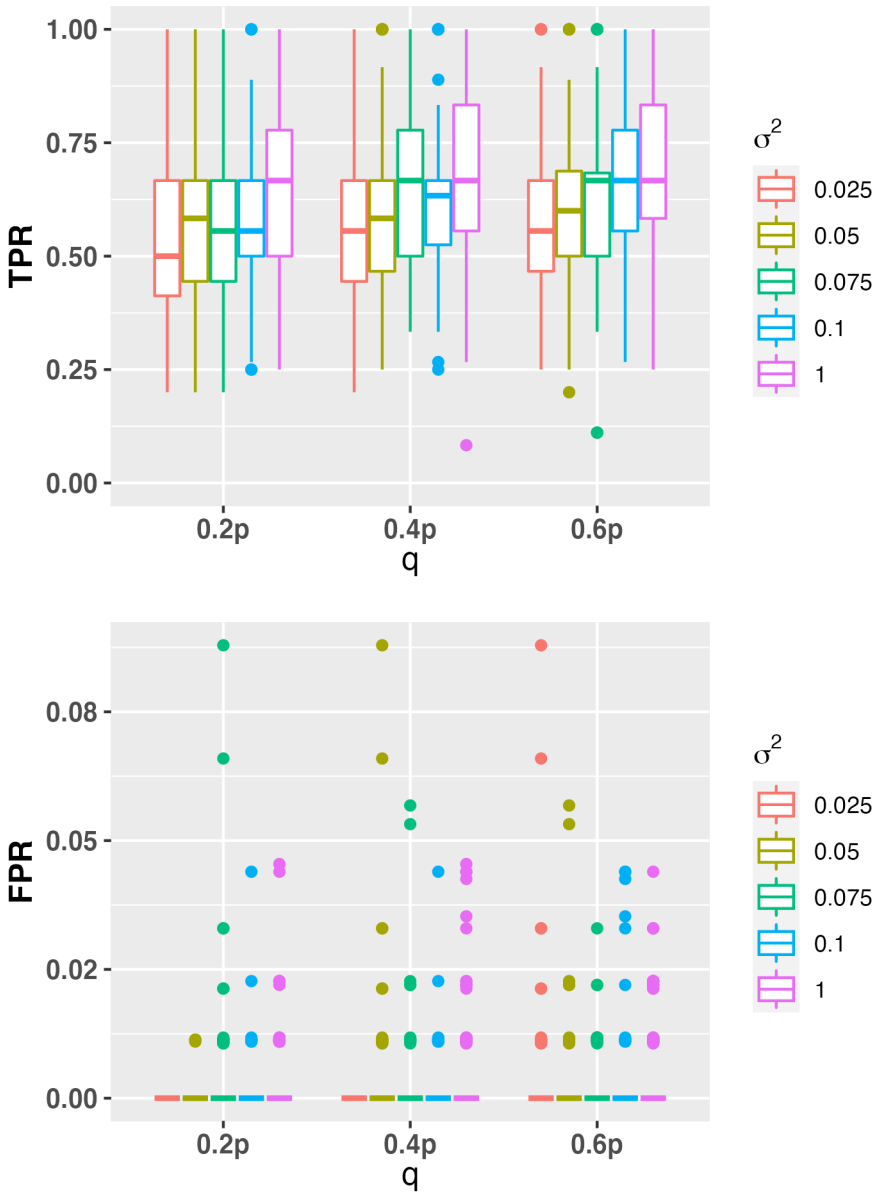


Figure B.13: TPR/FPR for BPA under different q values. Other parameters are fixed at $a_T = 100$, $\phi = 0.8$, $B = 1$.

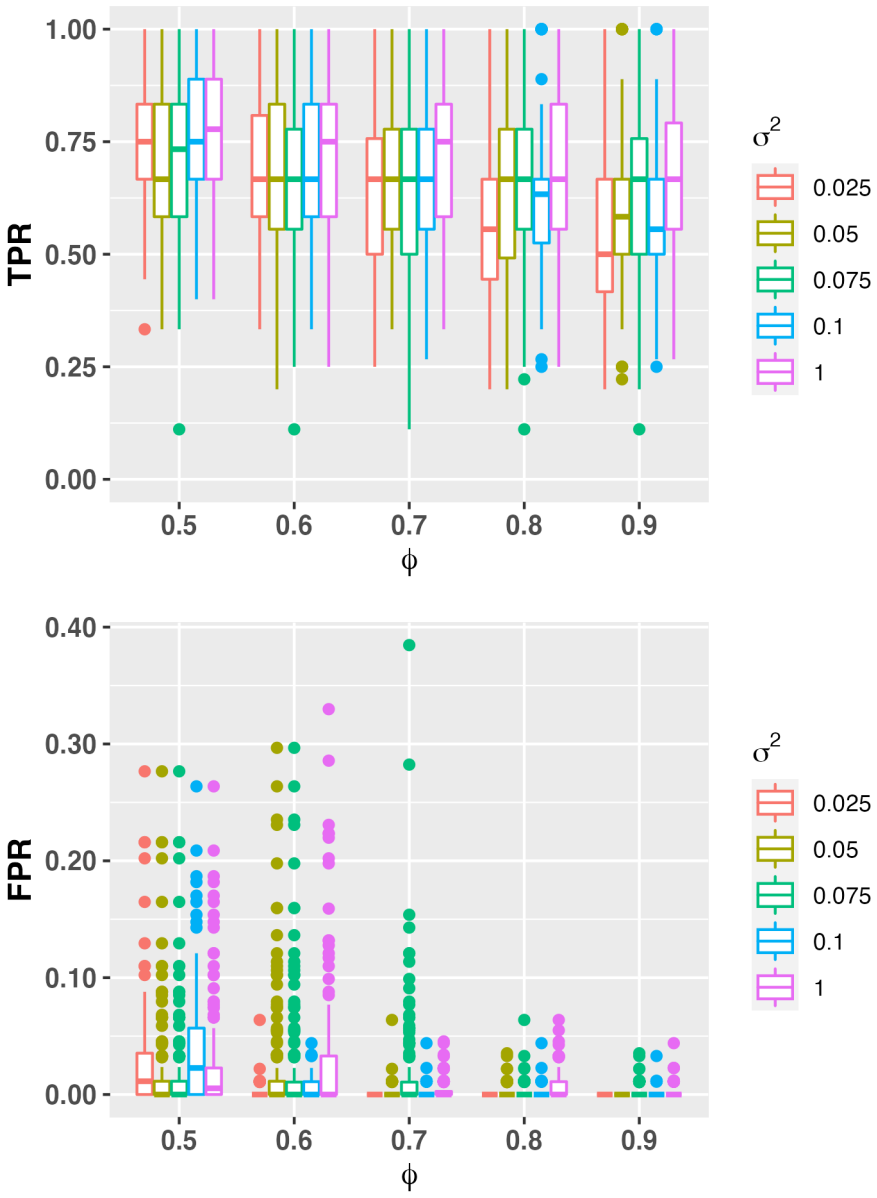


Figure B.14: TPR/FPR for BPA under different ϕ values. Other parameters are fixed at $a_T = 100$, $q = 0.4$, $B = 1$.

Bibliography

Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics 21*(1), 243–247.

Akaike, H. (1970). Statistical predictor identification. *Annals of the Institute of Statistical Mathematics 22*(1), 203–217.

Biau, G., F. Cérou, and A. Guyader (2010). On the rate of convergence of the bagged nearest neighbor estimate. *Journal of Machine Learning Research 11*(2).

Bijral, A. S. (2019, May). On Selecting Stable Predictors in Time Series Models. *arXiv:1905.07659 [cs, stat]*. arXiv: 1905.07659.

Breiman, L. (1996). Bagging predictors. *Machine learning 24*(2), 123–140.

Breiman, L. (1999). Using adaptive bagging to debias regressions. Technical report, Technical Report 547, Statistics Dept. UCB.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984, October). *Classification And Regression Trees* (1 ed.).

Bühlmann, P. and S. Van De Geer (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

Bühlmann, P. and B. Yu (2002). Analyzing bagging. *The Annals of Statistics 30*(4), 927–961.

Bühlmann, P. (2006). Boosting for high-dimensional linear models. *Annals of Statistics 34*(2), 559 – 583.

Bühlmann, P. and S. v. d. Geer (2011, June). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Science & Business Media.

Carvalho, C. M., N. G. POLSON, and J. G. SCOTT (2010). The horseshoe estimator for sparse signals. *Biometrika 97*(2), 465–480.

Cavanaugh, J. E. (1999). A large-sample model selection criterion based on kullback's symmetric divergence. *Statistics Probability Letters 42*(4), 333–343.

Christiano, L. J., M. Eichenbaum, and C. L. Evans (1999). Chapter 2 Monetary policy shocks: What have we learned and to what end? Volume 1 of *Handbook of Macroeconomics*, pp. 65–148. Elsevier.

Connor, G. and R. A. Korajczyk (1993). A test for the number of factors in an approximate factor model. *The Journal of Finance 48*(4), 1263–1291.

Duzan, H. and N. S. B. M. Shariff (2015). Ridge regression for solving the multicollinearity problem: review of methods and models. *Journal of Applied Science*.

Hall, P. and R. J. Samworth (2005). Properties of bagged nearest neighbour classifiers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(3), 363–379.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer.

Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics 12*(1), 55–67.

Hofmann, T., B. Schölkopf, and A. J. Smola (2008). Kernel methods in machine learning. *The Annals of Statistics 36*(3), 1171–1220.

Hotelling, H. (1992). Relations between two sets of variates. In *Breakthroughs in statistics*, pp. 162–190. Springer.

Li, F., C. M. Triggs, B. Dumitrescu, and C. D. Giurcăneanu (2019). The matching pursuit algorithm revisited: A variant for big data and new stopping rules. *Signal Processing 155*, 170–181.

Li, F., C. M. Triggs, B. Dumitrescu, and C. Giurcăneanu (2017). On the number of iterations for the matching pursuit algorithm. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 181–185.

Liu, Y.-Y., J.-J. Slotine, and A.-L. Barabási (2011, May). Controllability of complex networks. *Nature 473*(7346), 167–173.

Lütkepohl, H. (2005). *New introduction to multiple time series analysis*. Springer Science & Business Media.

Makalic, E. and D. F. Schmidt (2016). High-Dimensional Bayesian Regularised Regression with the BayesReg Package. Publisher: arXiv Version Number: 3.

Meinshausen, N. and P. Bühlmann (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(4), 417–473.

Mol, C. D., D. Giannone, and L. Reichlin (2008). Forecasting using a large number of predictors: Is Bayesian shrinkage a valid alternative to principal components? *Journal of Econometrics 146*(2), 318–328.

Muthukrishnan, R. and R. Rohini (2016). Lasso: A feature selection technique in predictive modeling for machine learning. In *2016 IEEE international conference on advances in computer applications (ICACA)*, pp. 18–20. IEEE.

Ng, S. (2013). Variable selection in predictive regressions. *Handbook of economic forecasting 2*, 752–789.

Park, T. and G. Casella (2008, June). The Bayesian Lasso. *Journal of the American Statistical Association 103*(482), 681–686.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science 2*(11), 559–572.

Samworth, R. J. (2012). Optimal weighted nearest neighbour classifiers. *The Annals of Statistics 40*(5), 2733–2763.

Sancetta, A. (2016). Greedy algorithms for prediction. *Bernoulli 22*(2), 1227 – 1277.

Schmidt, D. F. and E. Makalic (2009). MML Invariant Linear Regression. In A. Nicholson and X. Li (Eds.), *AI 2009: Advances in Artificial Intelligence*, Berlin, Heidelberg, pp. 312–321. Springer Berlin Heidelberg.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 461–464.

Shah, R. D. and R. J. Samworth (2013). Variable selection with error control: another look at stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 75*(1), 55–80.

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological) 58*(1), 267–288.

Wold, H. O. A. (1968). *Nonlinear estimation by iterative least square procedures.*

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association 101*(476), 1418–1429.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology) 67*(2), 301–320.