Adjusting for Selection Bias Using Gaussian Process Models

by

Meng Du

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Statistical Sciences
University of Toronto

Abstract

Adjusting for Selection Bias Using Gaussian Process Models

Meng Du

Doctor of Philosophy

Department of Statistical Sciences

University of Toronto

2014

This thesis develops techniques for adjusting for selection bias using Gaussian process models. Selection bias is a key issue both in sample surveys and in observational studies for causal inference. Despite recently emerged techniques for dealing with selection bias in high-dimensional or complex situations, use of Gaussian process models and Bayesian hierarchical models in general has not been explored.

Three approaches are developed for using Gaussian process models to estimate the population mean of a response variable with binary selection mechanism. The first approach models only the response with the selection probability being ignored. The second approach incorporates the selection probability when modeling the response using dependent Gaussian process priors. The third approach uses the selection probability as an additional covariate when modeling the response. The third approach requires knowledge of the selection probability, while the second approach can be used even when the selection probability is not available. In addition to these Gaussian process approaches, a new version of the Horvitz-Thompson estimator is also developed, which follows the conditionality principle and relates to importance sampling for Monte Carlo simulations.

Simulation studies and the analysis of an example due to Kang and Schafer show that the Gaussian process approaches that consider the selection probability are able to not only correct selection bias effectively, but also control the sampling errors well, and therefore can often provide more efficient estimates than the methods tested that are not based on Gaussian process models, in both simple and complex situations. Even the Gaussian process approach that ignores the selection probability often, though not always, performs well when some selection bias is present.

These results demonstrate the strength of Gaussian process models in dealing with selection bias, especially in high-dimensional or complex situations. These results also demonstrate that Gaussian process models can be implemented rather effectively so that the benefits of using Gaussian process models can be realized in practice, contrary to the common belief that highly flexible models are too complex to use practically for dealing with selection bias.

Acknowledgements

I sincerely thank Radford Neal, my thesis advisor, without whose guidance and support this work would have been impossible. I am particularly inspired by his provocative insights and critical scientific attitudes. I also appreciate his humbleness, integrity, and magnanimity. It is very fortunate of me to have been influenced by his great virtues.

This thesis was supported by Radford Neal's NSERC research grants.

Contents

Chapter 1

Introduction

Adjusting for selection bias is a key issue in statistical inference whenever selection probabilities are involved. In high-dimensional or complex situations, dealing with such an issue can be very difficult. Despite a large number of techniques that have emerged recently for dealing with selection bias, Bayesian hierarchical models have not been explored in this area to our best knowledge. In this thesis, I will demonstrate how Gaussian process models, a type of Bayesian hierarchical models, can be effectively utilised for dealing with selection bias.

1.1 The problem

Selection bias arises in both sample surveys and observational studies for causal inference, when sampling of survey units or assigning of treatment exposures is not completely at random, but instead depends on some covariate variables which also affect the outcome of interest.

1.1.1 Selection probability and selection bias in sample survey

Suppose $(\mathbf{x}_1, y_1, r_1), \ldots, (\mathbf{x}_n, y_n, r_n)$ are n independent and identically distributed realizations of a three element random tuple $(\mathbf{X}, Y, R)$. $\mathbf{X}$ is a d-dimensional vector of covariates. Y is the outcome variable of interest. R is a binary variable, indicating if Y is observed or not. The probability that Y is observed given $\mathbf{X} = \mathbf{x}$, denoted by $\nu(\mathbf{x}) = \Pr(R = 1|\mathbf{X} = \mathbf{x})$, is called the selection probability or the selection probability function when considered as a function of $\mathbf{x}$. In this thesis, we assume *strong ignorability* (Rosenbaum and Rubin, 1983) that given $\mathbf{X}$, R and Y are independent. Suppose that the goal is to estimate the population mean of Y, denoted by $\phi = \mathrm{E}[Y] = \mathrm{E}[\mu(\mathbf{X})]$, where $\mu(\mathbf{x}) = \mathrm{E}[Y|\mathbf{X} = x]$ is the mean function of Y. And suppose that $\mathbf{X}$ has a d-dimensional probability measure $F_{\mathbf{X}}$ or a density function $f_{\mathbf{X}}$ when continuous, then

$$\phi = \int \mu(\mathbf{x}) dF_{\mathbf{X}}(\mathbf{x}) = \int \mu(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \tag{1.1}$$

Since both $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ depend on $\mathbf{x}$, any method of estimation for ϕ that ignores both $\mathbf{x}$ and $\nu(\mathbf{x})$ may be biased. For example, the simple average of the observed y_i, i.e. $\sum_{i=1}^{n} y_i r_i / \sum_{i=1}^{n} r_i$, is a biased estimator for ϕ, unless $\mu(\mathbf{x})$ or $\nu(\mathbf{x})$ is a constant function or there is unlikely exact cancellation of biases over regions of $\mathbf{x}$. This type of bias is called *selection bias* in the context of sample surveys. Selection bias may occur in any inference problem that involves selection probabilities. Techniques developed for estimating the population mean should be readily extensible to other problems such as estimating regression coefficients.

In most of this thesis, I will assume that the selection probability function $\nu(\mathbf{x})$ is bounded away from zero by some fixed constant $\zeta > 0$. This assumption is essential since otherwise, there are always some $\mathbf{x}$'s whose selection probabilities are so small that they may almost never be observed with a practical sample size. For convenience, I will also use $\mathbf{x}$, y and r instead of $\mathbf{X}$, Y and R referring to both the random variables and their realizations unless confusion is present.

1.1.2 Connection to propensity scores and confounding bias

Consider a four element random tuple $(\mathbf{X}, Y_{(1)}, Y_{(0)}, T)$. Again, $\mathbf{X}$ is a d-dimensional vector of covariates. T is a binary treatment assignment indicator. $Y_{(1)}$ is the response variable if treatment is received, i.e. $T = 1$; $Y_{(0)}$ is the response variable if control is given, i.e. $T = 0$. Note that of course, only one of $Y_{(1)}$ and $Y_{(0)}$ can be observed. The probability that treatment is assigned given $\mathbf{X} = \mathbf{x}$, denoted by $p(\mathbf{x}) = \Pr(T = 1|\mathbf{X} = \mathbf{x})$, is called the propensity score or the propensity score function when considered as a function of $\mathbf{x}$. Again, we assume *strong ignorability* (Rosenbaum and Rubin, 1983) that given $\mathbf{X}$, $Y_{(1)}$ and $Y_{(0)}$ are independent of T. The goal is to estimate the population treatment effect, i.e. $\mathrm{E}[Y_{(1)}] - \mathrm{E}[Y_{(0)}]$. When the covariate vector $\mathbf{X}$ affects both the responses $Y_{(1)}$ and $Y_{(0)}$ and the treatment assignment T, any estimation procedure for $\mathrm{E}[Y_{(1)}] - \mathrm{E}[Y_{(0)}]$ that ignores both $\mathbf{x}$ and $p(\mathbf{x})$ will be biased unless $\mathrm{E}[Y_{(1)}|\mathbf{X} = \mathbf{x}] - \mathrm{E}[Y_{(0)}|\mathbf{X} = \mathbf{x}]$ or $p(\mathbf{x})$ is a constant function or there is unlikely exact cancellation of biases over regions of $\mathbf{x}$. This type of bias is called *confounding bias* in the context of observational studies for causal inference.

A propensity score problem can be considered as two sample survey problems. Specifically, if a subject receives treatment, the treatment response will be observed with the "selection" probability equal to $p(\mathbf{x})$. If a subject receives control, the control response will be observed with the "selection" probability equal to $1-p(\mathbf{x})$. Therefore, selection bias due to selection probability and confounding bias due to propensity score are equivalent statistical issues under different contexts. Although sometimes, estimating the treatment effect directly can be more efficient than estimating $\mathrm{E}[Y_{(1)}]$ and $\mathrm{E}[Y_{(0)}]$ separately, techniques developed for dealing with selection bias in sample survey should also be applicable for dealing with confounding bias.

1.2 Existing approaches to addressing selection bias

Conventional techniques for adjusting for selection bias include weighting, matching, stratification and covariate adjustment (e.g. Cochran, 1965, 1968; LaLonde, 1986; Dehejia and Wahba, 2002; Dehejia, 2005; Austin, 2008). For example, the Horvitz-Thompson (HT) method (Horvitz and Thompson, 1952) weights the observed responses y's using the inverse of the selection probability, $\nu(\mathbf{x})$, resulting in an unbiased estimator for the population mean ϕ.

When the covariate vector $\mathbf{x}$ is high-dimensional, weighting, matching, stratification or covariate adjustment based on the selection probability $\nu(\mathbf{x})$ only is easier to implement than on $\mathbf{x}$ itself and can still produce unbiased or consistent results (e.g. Rubin, 2001, 2007). However, when $\mathbf{x}$ is high-dimensional, only adjusting on the selection probability will produce inefficient results due to substantial information reduction. Even in low-dimensional situations, regression on $\mathbf{x}$ combined with adjusting on selection probabilities has been recommended for achieving better results (e.g. Cochran, 1957; Cochran and Rubin, 1973). In addition, methods based on adjusting on the selection probability require the knowledge of the selection probabilities. When they are unknown, estimating the selection probabilities may be as difficult as estimating the mean function $\mu(\mathbf{x})$, especially when $\mathbf{x}$ is high-dimensional.

One recently developed class of methods are double-robust (DR) methods (e.g. Robins and Rotnitzky, 1995; Scharfstein et al., 1999; Kang and Schafer, 2007; Rotnitzky et al., 2012). A DR method requires specifying two models: one for the response population, i.e. the y-model; the other for the selection mechanism, i.e. the selection probability function or the ν-model. When the two models are combined properly, a DR estimator remains consistent if one of the two models is correctly specified even if the other is not. There are various ways of constructing a DR method. For example, one can use a function of the selection probability as an additional covariate in a regression model; or regress y on $\mathbf{x}$ within classes stratified on selection probabilities; or apply weighted estimating equations or regression functions with weights equal to the inverse probabilities.

DR methods are more robust than methods based on only a y-model if the ν-model is correctly specified, and more efficient than methods without a y-model if the y-model is correctly specified. However, if both the y-model and the ν-model are misspecified, a DR method may not be better, or may even be worse, than a method using only one of the incorrect models (Kang and Schafer, 2007).

Additionally, although DR methods are “double guarded” for achieving consistent estimation, they do not promise that the estimation is efficient, especially for complex problems, if the y-model is not flexible enough. A DR method with a simple y-model may not perform better than a method without a ν-model but with a flexible y-model, if the simple y-model does not capture important information well enough.

As widely agreed, for a method to achieve efficient results, the covariate information must be used to a maximum extent, yet without overfitting the data. When the problem is complex or high-dimensional, constructing a proper y-model in a traditional class such as linear polynomial regression models, may be quite difficult, if not impossible. The regression model may become extremely complicated, with more parameters than the data can support, and then cause model overfitting. Kang and Schafer (2007, p525) argued that “with many covariates, it becomes difficult to specify a y-model that is sufficiently flexible to capture important nonlinear effects and interactions, yet parsimonious enough to keep the variance of prediction manageably low”.

Such flexible models do exist, however. Many well established Bayesian hierarchical models are highly flexible and fairly easy to implement including, for example, Gaussian process models and Bayesian neural networks. Unfortunately, the capacity of these Bayesian hierarchical models in dealing with selection bias in complex problems has not been widely recognized. Gelman (2007) pointed out the merits of Bayesian hierarchical models for high-dimensional problems only on the conceptual level by simple illustration. The strength of Bayesian hierarchical models has yet to be demonstrated through more sophisticated experiments.

Ideally, a Bayesian hierarchical model should capture all the information contained in the covariate vector $\mathbf{x}$ and therefore produce consistent estimation even without exploiting the selection probability explicitly. Robins and Ritov (1997) have, however, argued through extremely complex worse-case examples that any Bayesian method that does not use the selection probability will fail to be uniformly consistent under the set of semiparametric models that are indexed by the mean function μ only. In their view, uniform consistency is important since in high-dimensional or complex problems, the sample size can never be large enough for estimating the mean function $\mu(\mathbf{x})$ well. Therefore, they claim that methods such as the simple inverse probability weighted (IPW) Horvitz-Thompson (HT) estimator (Horvitz and Thompson, 1952) that is uniformly $\sqrt{n}$-consistent is more desired than any Bayesian method that ignores the selection probability.

In practice, however, the problem may never be as complex as the worst cases considered by Robins and Ritov (1997). Therefore, their argument against Bayesian methods without using the selection probability may not be practically relevant. In addition, from a Bayesian point of view, the worst-case scenarios in Robins and Ritov (1997) only occur with tiny probabilities *a priori*. Bickel and Kleijin

(2012) have shown that over a smaller set of semiparametric models, Bayesian estimators (ignoring $\nu(\mathbf{x})$) can be uniformly $\sqrt{n}$-consistent. Actually, Ritov et al. (2013, Theorem 7.1) have demonstrated that uniformly $\sqrt{n}$-consistent Bayesian estimators (ignoring $\nu(\mathbf{x})$) do exist under the set of semiparametric models considered by Robins and Ritov (1997) except subsets of zero prior probability measure. Therefore, those worst-case scenarios both in Robins and Ritov (1997) and in Ritov et al. (2013) should not present an issue to a Bayesian who trusts that their prior is well matched to realities.

If extremely complex situations do happen, in which the two functions $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ are correlated in a complex manner, Bayesian methods ignoring the selection probability may indeed not do well unless the sample size is huge. Nevertheless, with Bayesian hierarchical models, the selection probability can be easily incorporated in multiple ways. In a simplified finite population example from Wasserman (2004) where $\mathbf{x} \in \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, y is binary and $\phi = \sum_{i=1}^{N} \mu(\mathbf{x}_i)$, Ritov et al. (2013) showed that using the following prior that depends on $\nu(\mathbf{x})$

$$\mu(\mathbf{x}_i) \sim Beta\,(p_T(i), 1 - p_T(i)) \quad \text{with} \quad p_T(i) = \frac{e^{T/\nu(\mathbf{x}_i)}}{1 + e^{T/\nu(\mathbf{x}_i)}} \tag{1.2}$$

where T is an unknown hyperparameter, the Bayesian estimator is uniformly $\sqrt{n}$-consistent for ϕ under the semiparametric models. However, this approach still uses the inverse of the selection probability and the choice of $p_T(i)$ seems arbitrary in our view. The selection probability can actually be incorporated more flexibly if orthodox Bayesian hierarchical models such as Gaussian process models are adopted.

1.3 Bayesian inference using Gaussian process models

This thesis will demonstrate how Gaussian process models can be implemented effectively for dealing with selection bias in both simple and complex problems. Gaussian process models are non-parametric regression models which assign Gaussian process priors to the regression functions of interest. As flexible models, Gaussian process models are able to capture high-order nonlinear and interaction effects without restricting the maximum effect order. Gaussian process models can be implemented to employ the selection probability in several ways. One can assign dependent Gaussian process priors to both the y-model and the selection probability function $\nu(\mathbf{x})$ jointly. With dependent priors, the model will update the hyperparameters not only according to the observed y's but also to the selection indicators r's, or if known, selection probabilities $\nu(\mathbf{x})$, through the prior relationship between the y-model and the selection probability function, thereby effectively adjusting for selection bias in a flexible manner. Alternatively, instead of using dependent priors, one can use the $\nu(\mathbf{x})$ function as an additional covariate in the y-model. Due to the flexibility of Gaussian process models, one need not worry much about how the selection probability should be entered into the y-model. In particular, we do not have to use the inverse of the selection probability as the covariate, because the model can automatically decide the best relationship between y and this additional covariate. Incorporating the selection probability into the y-model flexibly in either way may help achieve more efficient results in either simple or complex situations.

Three approaches are developed in this thesis for using Gaussian process models for the problem of estimating the population mean ϕ as described earlier. The first approach models the mean function only and ignores the selection probability. The second approach models the mean function with the selection probability incorporated using dependent priors. The third approach uses the selection probability as an additional covariate while modeling the mean function. When using the selection probability as a covariate, the selection probabilities must be known at least for the observed covariate vectors. When modeling the mean function with dependent priors, the selection probabilities need not be available, although exploiting the known selection probabilities simplifies the estimating procedure and may also help achieve better results. The estimators based on these three approaches for using Gaussian process models will be compared to other estimators through both simulation experiments and an example due to Kang and Schafer (2007).

1.4 Outline

This thesis consists of five chapters and one appendix. Chapter 2 describes two groups of methods for adjusting for selection bias — those using a model or not, and discusses the fundamental difference between these two groups. Chapter 3 presents in detail how to make inference for ϕ using Gaussian process models and describes the Markov chain Monte Carlo (MCMC) algorithms used for sampling from the posterior distribution of ϕ based on Gaussian process models. Chapter 4 illustrates how Gaussian process methods perform compared to other methods through simulation experiments and an example due to Kang and Schafer. Chapter 5 summarizes the results of this thesis, discusses the limitation of the present work and identifies a number of research directions for future work. Appendix A lists all the figures that do not fit into the main body. The associated computing programs using R language are available at http://www.cs.toronto.edu/~radford/ftp/meng-r-functions.r.

Chapter 2

Methodologies

This chapter will present two types of methodologies for adjusting for selection bias: non-model based frequentist methods and Bayesian methods based on Gaussian process models.

2.1 Methods without a model

This section will review four non-model based frequentist estimators that will be compared with Gaussian process based estimators in the following experimental studies. The four estimators include one naive estimator and three types of Horvitz-Thompson (HT) estimators, denoted by $\widehat{\phi}_{naive}$, $\widehat{\phi}_{HT_1}$, $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$, respectively.

2.1.1 The naive estimator

The naive estimator is the one which ignores the selection probabilities and takes a simple average of the observed y_i's with equal weights, as given by

$$\widehat{\phi}_{naive} = \frac{\sum_{i=1}^{n} y_i r_i}{\sum_{i=1}^{n} r_i} = \frac{1}{n_{eff}} \sum_{i:r_i=1} y_i \tag{2.1}$$

where $n_{eff} = \sum_{i=1}^{n} r_i$ is the effective sample size. Note that $\widehat{\phi}_{naive}$ is not defined when $n_{eff} = 0$.

Since the naive estimator does not consider the selection probabilities at all, it is not expected to perform well when there is substantial correlation between the function of interest and the selection probability function. Clearly, the naive estimator could be severely biased when strong correlation between the two functions is present. However, we may wonder if the naive estimator might be nearly as good as other estimators when the correlation between the two functions is relatively weak. Another situation where the naive estimator might do comparably well as other methods is when the function of interest is restricted within a narrow band (i.e. almost a constant) and when the sample size is small. In this case, even if the correlation between the function of interest and the selection probability function is strong, due to the limited sample size, the sampling error may dominate the selection bias so that estimators that do consider the selection probabilities would have little practical advantage.

The naive estimator will be included in all the experimental studies in this thesis, for the behavior of it may help identify when selection bias is indeed an issue and therefore help decide which scenarios are meaningful to investigate.

2.1.2 The Horvitz-Thompson estimator: type 1

Unlike the naive estimator, the estimator originally given by Horvitz and Thompson (1952) weights each observed y_i with the inverse of the corresponding selection probability $\nu_i = \nu(\mathbf{x}_i)$, provided that ν_i's are available for all $\mathbf{x}_i$'s with $r_i = 1$. This Horvitz-Thompson estimator is defined by

$$\widehat{\phi}_{HT_1} = \frac{1}{n}\sum_{i=1}^{n}\frac{y_i r_i}{\nu_i} = \sum_{i:r_i=1}\frac{y_i}{n\nu_i} \tag{2.2}$$

Clearly, $\widehat{\phi}_{HT_1}$ is unbiased for the population mean ϕ, since for all $i = 1, 2, \ldots, n$,

$$\mathrm{E}\left[\frac{y_i r_i}{\nu_i}\right] = \mathrm{E}\left[\frac{y_i r_i}{\nu(\mathbf{x}_i)}\right] = \mathrm{E}\left[\mathrm{E}\left[\frac{y_i r_i}{\nu(\mathbf{x}_i)}\middle|\mathbf{x}_i\right]\right] = \mathrm{E}\left[\frac{\mu(\mathbf{x}_i)\nu(\mathbf{x}_i)}{\nu(\mathbf{x}_i)}\right] = \mathrm{E}[\mu(\mathbf{x}_i)] = \phi \tag{2.3}$$

and by the law of large numbers, it is also consistent for ϕ.

Although $\widehat{\phi}_{HT_1}$ has arguably been the standard non-model based frequentist estimator for its simplicity and being unbiased when selection bias is present, it bears two obvious drawbacks. The first drawback is evident when all the y_i's equal some non-zero constant $c \neq 0$, in which case $\widehat{\phi}_{HT_1}$ does not equal c. Clearly, this drawback makes $\widehat{\phi}_{HT_1}$ non-equivariant under certain affine transformations of y_i's. For example, when y_i's are binary, reversing the coding, i.e. $y_i^* = 1 - y_i$, will not give the corresponding estimate $1 - \widehat{\phi}_{HT_1}$. Similarly, when y_i's are numerical, we will not get $\widehat{\phi}_{HT_1} + c$, if y_i's are measured from a different origin, i.e. $y_i^* = y_i + c$ with $c \neq 0$. The second flaw of $\widehat{\phi}_{HT_1}$ is common to all non-model based estimators which ignore the covariate vector $\mathbf{x}$. Particularly, by averaging each $y_i r_i/\nu_i$ with an equal weight $1/n$, all the units of observation are treated as equally important, which is often not true in practice. Assuming all the units being equally important is another form of naivety and could conceivably lead to severely inefficient results, especially when there are a large number of clustered units due to randomness.

2.1.3 The Horvitz-Thompson estimator: type 2

A variant of $\widehat{\phi}_{HT_1}$ that has been often used in practice in replacement of $\widehat{\phi}_{HT_1}$ is given by

$$\widehat{\phi}_{HT_2} = \sum_{i=1}^{n} \frac{y_i r_i}{\nu_i} \Big/ \sum_{i=1}^{n} \frac{r_i}{\nu_i} = \sum_{i:r_i=1} \frac{y_i}{\nu_i} \Big/ \sum_{i:r_i=1} \frac{1}{\nu_i} \tag{2.4}$$

Although not unbiased, $\widehat{\phi}_{HT_2}$ is still consistent for ϕ, since by the strong law of large numbers, both $n^{-1}\sum_{i=1}^{n} y_i r_i/\nu_i \to \phi$ and $n^{-1}\sum_{i=1}^{n} r_i/\nu_i \to 1$ with probability one. Note that like the naive estimator, $\widehat{\phi}_{HT_2}$ is not defined when all the r_i's equal zero.

One advantage of $\widehat{\phi}_{HT_2}$ over $\widehat{\phi}_{HT_1}$ is its equivariance under all affine transformations. More clearly, rewrite vas

$$\widehat{\phi}_{HT_2} = \sum_{i=1}^{n} \left(\frac{r_i}{\nu_i} \Big/ \sum_{i=1}^{n} \frac{r_i}{\nu_i} \right) y_i \tag{2.5}$$

Since $\sum_{i=1}^{n} \left(\frac{r_i}{\nu_i} / \sum_{i=1}^{n} \frac{r_i}{\nu_i} \right) \equiv 1$, $\widehat{\phi}_{HT_2}$ simply equals c when all y_i's equal c, therefore is equivariant under all affine transformations. The equivariance of $\widehat{\phi}_{HT_2}$ is extensible to the situation when y is not a constant but has a constant mean value, i.e. $\mu(\mathbf{x}) \equiv c$. In such a case,

$$\begin{aligned}
\mathrm{E}[\widehat{\phi}_{HT_2}|\mathbf{x}_1,\ldots,\mathbf{x}_n] &= \mathrm{E}\left[\sum_{i=1}^{n} \left(\frac{r_i}{\nu_i} \Big/ \sum_{i=1}^{n} \frac{r_i}{\nu_i} \right) y_i \middle| \mathbf{x}_1,\ldots,\mathbf{x}_n \right] \\
&= \sum_{i=1}^{n} \mathrm{E}\left[\left(\frac{r_i}{\nu_i} \Big/ \sum_{i=1}^{n} \frac{r_i}{\nu_i} \right) \middle| \mathbf{x}_1,\ldots,\mathbf{x}_n \right] \times \mathrm{E}[y_i|\mathbf{x}_i] \\
&= \sum_{i=1}^{n} \mathrm{E}\left[\left(\frac{r_i}{\nu_i} \Big/ \sum_{i=1}^{n} \frac{r_i}{\nu_i} \right) \middle| \mathbf{x}_1,\ldots,\mathbf{x}_n \right] \times \mu(\mathbf{x}_i) \\
&= \sum_{i=1}^{n} \mathrm{E}\left[\left(\frac{r_i}{\nu_i} \Big/ \sum_{i=1}^{n} \frac{r_i}{\nu_i} \right) \middle| \mathbf{x}_1,\ldots,\mathbf{x}_n \right] \times c \equiv c, \quad \text{for any } \mathbf{x}_1,\ldots,\mathbf{x}_n.
\end{aligned} \tag{2.6}$$

2.1.4 The Horvitz-Thompson estimator: type 3

The type 3 Horvitz-Thompson estimator replaces ν_i in $\widehat{\phi}_{HT_1}$ with ν_i/ψ, where $\psi = \int \nu(\mathbf{x}) dF_{\mathbf{X}}(\mathbf{x})$ is the marginal selection probability, and then averages $\frac{y_i}{\nu_i/\psi}$ over only the observed units, as given by

$$\widehat{\phi}_{HT_3} = \sum_{i=1}^{n} \frac{y_i r_i}{\nu_i/\psi} \Big/ \sum_{i=1}^{n} r_i = \frac{1}{n_{eff}} \sum_{i:r_i=1} \frac{y_i \psi}{\nu_i} \tag{2.7}$$

Note that like $\widehat{\phi}_{HT_2}$, $\widehat{\phi}_{HT_3}$ is not defined when $n_{eff} = 0$.

By the strong law of large numbers, $\widehat{\phi}_{HT_3}$ is apparently consistent for ϕ as the other Horvitz-

Thompson estimators. In addition, $\widehat{\phi}_{HT_3}$ is conditionally unbiased given $n_{eff} > 0$, although not unbiased marginally. Actually, $\widehat{\phi}_{HT_3}$ is also conditionally unbiased given $r_1, r_2, \ldots, r_n$ as long as $n_{eff} > 0$. To show these, we first have

$$\begin{aligned} \mathrm{E}\left[\frac{y_i}{\nu_i/\psi}\Big| r_i = 1\right] &= \mathrm{E}\left[\frac{y_i}{\nu(\mathbf{x}_i)/\psi}\Big| r_i = 1\right] = \int \mathrm{E}\left[\frac{y_i}{\nu(\mathbf{x}_i)/\psi}\Big| \mathbf{x}_i\right] f_{\mathbf{X}_i|R_i}(\mathbf{x}_i|1) d\mathbf{x}_i \\ &= \int \frac{\mu(\mathbf{x}_i)}{\nu(\mathbf{x}_i)/\psi} \frac{\nu(\mathbf{x}_i) f_{\mathbf{X}}(\mathbf{x}_i)}{\psi} d\mathbf{x}_i = \int \mu(\mathbf{x}_i) f_{\mathbf{X}_i}(\mathbf{x}_i) d\mathbf{x}_i \ = \ \phi \end{aligned} \tag{2.8}$$

and then for all $r_1, r_2, \ldots, r_n$ with $\sum_{i=1}^n r_i > 0$,

$$\begin{aligned} \mathrm{E}[\widehat{\phi}_{HT_3}|r_1, r_2, \ldots, r_n] &= \mathrm{E}\left[\frac{1}{n_{eff}} \sum_{i:r_i=1} \frac{y_i}{\nu_i/\psi}\Big| r_1, r_2, \ldots, r_n\right] = \frac{1}{n_{eff}} \sum_{i:r_i=1} \mathrm{E}\left[\frac{y_i}{\nu_i/\psi}\Big| r_1, r_2, \ldots, r_n\right] \\ &= \frac{1}{n_{eff}} \sum_{i:r_i=1} \mathrm{E}\left[\frac{y_i}{\nu_i/\psi}\Big| r_i = 1\right] = \frac{1}{n_{eff}} \sum_{i:r_i=1} \phi = \phi. \end{aligned} \tag{2.9}$$

This proves the conditional unbiasedness of $\widehat{\phi}_{HT_3}$ given $r_1, r_2, \ldots, r_n$ with $n_{eff} > 0$. Then by

$$\mathrm{E}\left[\widehat{\phi}_{HT_3}\Big| \sum_{i=1}^n r_i > 0\right] = \mathrm{E}\left[\mathrm{E}\left[\widehat{\phi}_{HT_3}|r_1, r_2, \ldots, r_n\right]\Big| \sum_{i=1}^n r_i > 0\right] = \phi, \tag{2.10}$$

$\widehat{\phi}_{HT_3}$ is also unbiased given $n_{eff} > 0$.

2.1.5 Connection of Horvitz-Thompson estimators to importance sampling

Aside from being related to the original Horvitz-Thompson estimator $\widehat{\phi}_{HT_1}$, $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ can also be viewed as estimators based on *importance sampling*, a technique used in Monte Carlo simulations.

We start with a brief description of importance sampling; for more details, see Neal (2001). Suppose we want to estimate the mean of the function $h(a)$ with respect to the distribution f. However, sampling from f is difficult. Instead, suppose that, sampling from an alternative distribution f^* is more convenient and the ratio of f/f^* can be computed easily. With $a_1, \ldots, a_{\tilde{n}}$ sampled from f^*, an estimator for $\mathrm{E}_f[h(a)]$ can be constructed as

$$\widehat{h}_{IM} = \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} h(a_i) f(a_i)/f^*(a_i). \tag{2.11}$$

$\widehat{h}_{IM}$ is actually unbiased for $\mathrm{E}_f[h(a)]$, since

$$\begin{aligned} \mathrm{E}[\widehat{h}_{IM}] &= \mathrm{E}_{f^*}\left[\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} h(a_i)f(a_i)/f^*(a_i)\right] = \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\int h(a_i)\,\frac{f(a_i)}{f^*(a_i)}\,f^*(a_i)\,da_i \\ &= \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\int h(a_i)\,f(a_i)\,da_i \;=\; \mathrm{E}_f[h(a)] \end{aligned} \tag{2.12}$$

$\widehat{h}_{IM}$ is called importance sampling estimator for $\mathrm{E}_f[h(a)]$ and the ratio $f(a_i)/f^*(a_i)$ is called *importance weight.*

Equation (2.11) requires complete knowledge of f and f^*. When f and f^* are only known up to their normalizing constants, as is common for applications in Bayesian inference, replacing $\tilde{n}$ in the denominator by $\sum_{i=1}^{\tilde{n}} f(a_i)/f^*(a_i)$ results in the following alternative estimator for $\mathrm{E}_f[h(a)]$

$$\widetilde{h}_{IM} = \frac{\sum_{i=1}^{\tilde{n}} h(a_i)f(a_i)/f^*(a_i)}{\sum_{i=1}^{\tilde{n}} f(a_i)/f^*(a_i)}. \tag{2.13}$$

$\widetilde{h}_{IM}$ is no longer unbiased but is still consistent for $\mathrm{E}_f[h(a)]$ by the strong law of large numbers.

Now, recall that

$$\widehat{\phi}_{HT_3} = \frac{1}{n_{eff}}\sum_{i:r_i=1}\frac{y_i\psi}{\nu_i} \tag{2.14}$$

where $\psi = \int \nu(\mathbf{x})dF_{\mathbf{X}}(\mathbf{x}) = \Pr(r_i = 1)$. By simply noting that

$$\frac{\psi}{\nu_i} = \frac{f_{\mathbf{X}}(\mathbf{x}_i)}{f_{\mathbf{X}}(\mathbf{x}_i)\nu_i/\psi} = \frac{f_{\mathbf{X}}(\mathbf{x}_i)}{f_{\mathbf{X}|R_i}(\mathbf{x}_i|1)}. \tag{2.15}$$

is the ratio of the marginal density function of $\mathbf{x}_i$ to its conditional density function given $r_i = 1$, $\widehat{\phi}_{HT_3}$ is clearly an importance sampling estimator with $f = f_{\mathbf{X}}$ and $f^* = f_{\mathbf{X}|R}(\cdot|1)$. In other words, restricted to those x_i's with y_i observed, x_i's are considered as sampled from $f_{\mathbf{X}|R}(\cdot|1)$ (with $\tilde{n} = n_{eff}$) instead of from $f_{\mathbf{X}}$. Being an importance sampling estimator alternatively proves that $\widehat{\phi}_{HT_3}$ is unbiased for ϕ given $n_{eff} > 0$.

Similarly, with $f = f_{\mathbf{X}}$ and $f^* = f_{\mathbf{X}|R}(\cdot|1)$,

$$\widehat{\phi}_{HT_2} = \sum_{i:r_i=1}\frac{y_i}{\nu_i}\Big/\sum_{i:r_i=1}\frac{1}{\nu_i} = \sum_{i:r_i=1}\frac{y_i\psi}{\nu_i}\Big/\sum_{i:r_i=1}\frac{\psi}{\nu_i} = \widetilde{h}_{IM}$$

Therefore, $\widehat{\phi}_{HT_2}$ is the type of importance sampling estimator when ψ is unavailable. It should be noted that although $\widehat{\phi}_{HT_2}$ can be considered as an alternative to $\widehat{\phi}_{HT_3}$ when ψ is unavailable, $\widehat{\phi}_{HT_2}$ has its own merit of being equivariant under all affine transformations which $\widehat{\phi}_{HT_3}$ does not have.

2.1.6 MSE of types 1 and 3 Horvitz-Thompson estimators

In this subsection, I will derive the (asymptotic) mean squared errors (MSE) of $\widehat{\phi}_{HT_1}$ and $\widehat{\phi}_{HT_3}$ for estimating ϕ and then compare them. We start with two lemmas.

Lemma 2.1 *Assume that $\nu(\mathbf{x}) > \zeta$ for all $\mathbf{x}$ where $\zeta > 0$ and $E[y^2] < \infty$. Then*

$$\begin{aligned} E\left[\frac{y_i}{\nu_i}\Big| r_i = 1\right] &= \frac{\phi}{\psi} \\ Var\left(\frac{y_i}{\nu_i}\Big| r_i = 1\right) &= \frac{A}{\psi} - \frac{\phi^2}{\psi^2} \end{aligned} \tag{2.16}$$

where

$$A = \int \frac{E[y^2|\mathbf{x}] f_{\mathbf{X}}(\mathbf{x})}{\nu(\mathbf{x})} d\mathbf{x} \tag{2.17}$$

is a finite constant not depending on i.

Proof As noted earlier, $f_{\mathbf{X}|R}(\mathbf{x_i}|r_i = 1) = \frac{f_{\mathbf{X}}(\mathbf{x_i})\nu(\mathbf{x_i})}{\psi}$. Therefore,

$$\begin{aligned} \mathrm{E}\left[\frac{y_i}{\nu_i}\Big| r_i = 1\right] &= \mathrm{E}\left[\mathrm{E}\left[\frac{y_i}{\nu(\mathbf{x}_i)}\Big|\mathbf{x}_i\right]\Big| r_i = 1\right] \\ &= \mathrm{E}\left[\frac{\mu(\mathbf{x}_i)}{\nu(\mathbf{x}_i)}\Big| r_i = 1\right] = \int \frac{\mu(\mathbf{x})}{\nu(\mathbf{x})}\frac{f_{\mathbf{X}}(\mathbf{x})\nu(\mathbf{x})}{\psi} \mathrm{d}\mathbf{x} = \frac{\phi}{\psi} \end{aligned} \tag{2.18}$$

$$\begin{aligned} \mathrm{E}\left[\frac{y_i^2}{\nu_i^2}\Big| r_i = 1\right] &= \mathrm{E}\left[\mathrm{E}\left[\frac{y_i^2}{\nu^2(\mathbf{x}_i)}\Big|\mathbf{x}_i\right]\Big| r_i = 1\right] \\ &= \int \frac{\mathrm{E}[y^2|\mathbf{x}]}{\nu^2(\mathbf{x})}\frac{f_{\mathbf{X}}(\mathbf{x})\nu(\mathbf{x})}{\psi} \mathrm{d}\mathbf{x} = \frac{\int \frac{\mathrm{E}[y^2|\mathbf{x}] f_{\mathbf{X}}(\mathbf{x})}{\nu(\mathbf{x})} \mathrm{d}\mathbf{x}}{\psi} = \frac{A}{\psi} \end{aligned} \tag{2.19}$$

And then,

$$\mathrm{Var}\left(\frac{y_i}{\nu_i}\Big| r_i = 1\right) = \frac{A}{\psi} - \frac{\phi^2}{\psi^2} \tag{2.20}$$

Note that the finiteness of A is guaranteed by $\nu(\mathbf{x}) > \zeta > 0$ and $\mathrm{E}[y^2] < \infty$. This completes the proof of Lemma 2.1.

Lemma 2.2 *Assume that $\psi > 0$. Then*

$$E\left[\frac{1}{n_{eff}}\Big| \sum_{i=1}^{n} r_i > 0\right] = \frac{1}{n\psi} + o\left(\frac{1}{n^{3/2}}\right). \tag{2.21}$$

Proof We first note that when $\psi = 1$, $n_{eff} = n$ with probability one and therefore

$$\mathrm{E}\left[\frac{1}{n_{eff}}\middle|\sum_{i=1}^{n} r_i > 0\right] = \mathrm{E}\left[\frac{1}{n_{eff}}\right] = \frac{1}{n} \tag{2.22}$$

When $0 < \psi < 1$, $\frac{1}{n_{eff}}$ is not defined if $n_{eff} = 0$. However, since $\Pr(n_{eff} = 0) = (1-\psi)^n$ goes to zero exponentially fast, we can assign an arbitrary value to $\frac{1}{n_{eff}}$ at $n_{eff} = 0$ without causing any practical concerns. Then, we have by the generalized central limit theorem that

$$\frac{1/n_{eff} - 1/n\psi}{\sqrt{n\psi(1-\psi)/(n\psi)^4}} \quad \longrightarrow \quad N(0,1), \text{ as } n \to \infty \tag{2.23}$$

That is, for any Borel set $A \subset \mathcal{R}$,

$$\Pr\left(\frac{1/n_{eff} - 1/n\psi}{\sqrt{n\psi(1-\psi)/(n\psi)^4}} \in A\right) = \Pr\left(Z \in A\right) + o(1) \tag{2.24}$$

where $Z \sim N(0,1)$. Therefore,

$$\mathrm{E}\left[\frac{1/n_{eff} - 1/n\psi}{\sqrt{n\psi(1-\psi)/(n\psi)^4}}\right] = \mathrm{E}[Z] + o(1) = o(1) \tag{2.25}$$

And thus,

$$\mathrm{E}\left[\frac{1}{n_{eff}}\right] = \frac{1}{n\psi} + \sqrt{n\psi(1-\psi)/(n\psi)^4} \times o(1) = \frac{1}{n\psi} + o\left(\frac{1}{n^{3/2}}\right). \tag{2.26}$$

And since $\Pr(\sum_{i=1}^{n} r_i = 0) = (1-\psi)^n$ which goes to zero faster than $O\left(\frac{1}{n^{3/2}}\right)$,

$$\mathrm{E}\left[\frac{1}{n_{eff}}\middle|\sum_{i=1}^{n} r_i > 0\right] = \frac{1}{n\psi} + o\left(\frac{1}{n^{3/2}}\right). \tag{2.27}$$

This completes the proof of Lemma 2.2.

The MSE of $\widehat{\phi}_{HT_1}$ and $\widehat{\phi}_{HT_3}$ are given by Theorem 2.1 and Theorem 2.2, respectively.

Theorem 2.1 *Assume that $\nu(\mathbf{x}) > \zeta$ for all $\mathbf{x}$ where $\zeta > 0$ and $E[y^2] < \infty$. Then $\widehat{\phi}_{HT_1}$ has the following finite mean squared error*

$$MSE(\widehat{\phi}_{HT_1}) = \frac{(1-\psi)\phi^2}{n\psi} + \frac{\psi A - \phi^2}{n\psi} \tag{2.28}$$

where A is as defined in (2.17).

Proof Recall that $\widehat{\phi}_{HT_1}$ is unbiased for ϕ. Therefore,

$$\begin{aligned} MSE(\widehat{\phi}_{HT_1}) &= \mathrm{Var}(\widehat{\phi}_{HT_1}) \\ &= \mathrm{E}\left[\mathrm{Var}\left(\widehat{\phi}_{HT_1}|r_1, r_2, \ldots, r_n\right)\right] + \mathrm{Var}\left(\mathrm{E}\left[\widehat{\phi}_{HT_1}|r_1, r_2, \ldots, r_n\right]\right) \qquad (2.29) \end{aligned}$$

From Lemma 2.1,

$$\begin{aligned} \mathrm{Var}\left(\widehat{\phi}_{HT_1}|r_1, r_2, \ldots, r_n\right) &= \mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} y_i r_i/\nu_i \Big| r_1, r_2, \ldots, r_n\right) \\ &= \frac{n_{eff}}{n^2}\mathrm{Var}\left(y_i/\nu_i \Big| r_i = 1\right) = \frac{n_{eff}}{n^2}\left(\frac{A}{\psi} - \frac{\phi^2}{\psi^2}\right), \qquad (2.30) \end{aligned}$$

and then,

$$\begin{aligned} \mathrm{E}\left[\mathrm{Var}\left(\widehat{\phi}_{HT_1}|r_1, r_2, \ldots, r_n\right)\right] &= \mathrm{E}\left[\frac{n_{eff}}{n^2}\left(\frac{A}{\psi} - \frac{\phi^2}{\psi^2}\right)\right] \\ &= \frac{n\psi}{n^2}\left(\frac{A}{\psi} - \frac{\phi^2}{\psi^2}\right) = \frac{\psi A - \phi^2}{n\psi}. \qquad (2.31) \end{aligned}$$

Also from Lemma 2.1,

$$\begin{aligned} \mathrm{E}\left[\widehat{\phi}_{HT_1}|r_1, r_2, \ldots, r_n\right] &= \mathrm{E}\left[\frac{1}{n}\sum_{i=1}^{n} y_i r_i/\nu_i \Big| r_1, r_2, \ldots, r_n\right] \\ &= \frac{n_{eff}}{n}\mathrm{E}\left[y_i/\nu_i \Big| r_i = 1\right] = \frac{n_{eff}}{n}\frac{\phi}{\psi} \qquad (2.32) \end{aligned}$$

and then,

$$\mathrm{Var}\left(\mathrm{E}\left[\widehat{\phi}_{HT_1}|r_1, r_2, \ldots, r_n\right]\right) = \mathrm{Var}\left(\frac{n_{eff}}{n}\frac{\phi}{\psi}\right) = \frac{n\psi(1-\psi)}{n^2}\frac{\phi^2}{\psi^2} = \frac{(1-\psi)\phi^2}{n\psi} \qquad (2.33)$$

Therefore

$$MSE(\widehat{\phi}_{HT_1}) = \frac{(1-\psi)\phi^2}{n\psi} + \frac{\psi A - \phi^2}{n\psi}. \qquad (2.34)$$

This completes the proof of Theorem 2.1.

Theorem 2.2 *Assume that $\nu(\mathbf{x}) > \zeta$ for all $\mathbf{x}$ where $\zeta > 0$ and $E[y^2] < \infty$. Then $\widehat{\phi}_{HT_3}$ has the following asymptotic mean squared error, regardless its undefined value when $n_{eff} = \sum_{i=1}^{n} r_i = 0$.*

$$MSE\left(\widehat{\phi}_{HT_3}\right) = \frac{(\psi A - \phi^2)}{n\psi} + o\left(\frac{1}{n^{3/2}}\right) \qquad (2.35)$$

where A is as defined in (2.17).

Proof Recall that $\widehat{\phi}_{HT_3}$ is unbiased for ϕ given $\sum_{i=1}^n r_i > 0$ or given $r_1, r_2, \ldots, r_n$ with $\sum_{i=1}^n r_i > 0$. Therefore,

$$\begin{aligned}
MSE\left(\widehat{\phi}_{HT_3}\middle|\sum_{i=1}^n r_i > 0\right) &= \mathrm{Var}\left(\widehat{\phi}_{HT_3}\middle|\sum_{i=1}^n r_i > 0\right) \\
&= \mathrm{E}\left[\mathrm{Var}\left(\widehat{\phi}_{HT_3}|r_1, r_2, \ldots, r_n\right)\middle|\sum_{i=1}^n r_i > 0\right] \\
&+ \mathrm{Var}\left(\mathrm{E}\left[\widehat{\phi}_{HT_3}|r_1, r_2, \ldots, r_n\right]\middle|\sum_{i=1}^n r_i > 0\right) \\
&= \mathrm{E}\left[\mathrm{Var}\left(\widehat{\phi}_{HT_3}|r_1, r_2, \ldots, r_n\right)\middle|\sum_{i=1}^n r_i > 0\right] + \mathrm{Var}\left(\phi\middle|\sum_{i=1}^n r_i > 0\right) \\
&= \mathrm{E}\left[\mathrm{Var}\left(\widehat{\phi}_{HT_3}|r_1, r_2, \ldots, r_n\right)\middle|\sum_{i=1}^n r_i > 0\right] \\
&= \mathrm{E}\left[\mathrm{Var}\left(\frac{\psi}{n_{eff}}\sum_{i=1}^n y_i r_i/\nu_i\middle|r_1, r_2, \ldots, r_n\right)\middle|\sum_{i=1}^n r_i > 0\right] \\
&= \mathrm{E}\left[\mathrm{Var}\left(\frac{\psi}{n_{eff}}\sum_{i:r_i=1}^n y_i/\nu_i\middle|r_1, r_2, \ldots, r_n\right)\middle|\sum_{i=1}^n r_i > 0\right] \\
&= \mathrm{E}\left[\frac{\psi^2}{n_{eff}}\mathrm{Var}\left(\frac{y_i}{\nu_i}\middle|r_i = 1\right)\middle|\sum_{i=1}^n r_i > 0\right] \qquad (2.36)
\end{aligned}$$

From Lemma 2.1,

$$MSE\left(\widehat{\phi}_{HT_3}\middle|\sum_{i=1}^n r_i > 0\right) = \mathrm{E}\left[\frac{\psi^2}{n_{eff}}\left(\frac{A}{\psi} - \frac{\phi^2}{\psi^2}\right)\middle|\sum_{i=1}^n r_i > 0\right] = (\psi A - \phi^2)\mathrm{E}\left[\frac{1}{n_{eff}}\middle|\sum_{i=1}^n r_i > 0\right] \quad (2.37)$$

and then,

$$\begin{aligned}
MSE\left(\widehat{\phi}_{HT_3}\right) &= (1 - (1-\psi)^n)(\psi A - \phi^2)\mathrm{E}\left[\frac{1}{n_{eff}}\middle|\sum_{i=1}^n r_i > 0\right] \\
&+ (1-\psi)^n MSE\left(\widehat{\phi}_{HT_3}\middle|\sum_{i=1}^n r_i = 0\right) \qquad (2.38)
\end{aligned}$$

Since $(1-\psi)^n$ goes to zero faster than $O\left(\frac{1}{n^{3/2}}\right)$, regardless the undefined value of $\widehat{\phi}_{HT_3}$ when $\sum_{i=1}^n r_i = 0$, from Lemma 2.2,

$$MSE\left(\widehat{\phi}_{HT_3}\right) = \frac{(\psi A - \phi^2)}{n\psi} + o\left(\frac{1}{n^{3/2}}\right) \qquad (2.39)$$

This completes the proof of Theorem 2.2.

From Theorem 2.1 and Theorem 2.2, we have that

$$MSE(\widehat{\phi}_{HT_1}) - MSE(\widehat{\phi}_{HT_3}) = \frac{(1-\psi)\phi^2}{n\psi} + o\left(\frac{1}{n}\right) \tag{2.40}$$

That is, for sufficiently large n,

$$MSE(\widehat{\phi}_{HT_1}) > MSE(\widehat{\phi}_{HT_3}), \ \ \text{unless} \ \ \psi = 1 \ \ \text{or} \ \ \phi = 0. \tag{2.41}$$

2.1.7 A short summary

Note that, in the situation where the observations are missing completely at random, i.e., when all the selection probabilities equal a constant ν, both $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ reduce to the naive estimator, which takes a simple average of the observed y_i, i.e. $\frac{1}{n_{eff}} \sum_{i:r_i=1} y_i$. However, the original Horvitz-Thompson estimator, which equals $\frac{1}{n\nu} \sum_{i:r_i=1} y_i$ in this situation ignores the number of actually observed y_i's, i.e. $n_{eff} = \sum_{i=1}^n r_i$. Ignoring the ancillary effective sample size $n_{eff} = \sum_{i=1}^n r_i$, conflicts with the conditionality principle, which has been widely accepted in both theory and practice.

Although the Horvitz-Thompson estimators have the merit of adjusting for the selection probabilities through weighting, they have ignored all other possible aspects of variation among the covariates $\mathbf{x}_i$'s. A regression model with appropriately chosen predictors could potentially produce estimates of higher accuracy than the Horvitz-Thompson estimators. However, for a conventional regression model, e.g. a polynomial regression model fit by least squares, the complexity and flexibility of the model must be limited to avoid overfitting, especially in high dimensional situations. Instead, Gaussian process regression, a non-parametric method, can model various aspects of the covariates more flexibly than the traditional regression methods, without the risk of overfitting. Also, unlike the Horvitz-Thompson estimators which treat all the units of observation as equally important, a Gaussian process regression model will weight the importance of each $\mathbf{x}_i$ by its distance from other covariates, and therefore could be expected to be a more powerful method of inference. More detailed discussion on Gaussian process regression will come in the next section.

We may also note that when $n_{eff} = \sum_{i=1}^n r_i = 0$, all the estimators introduced in this section except $\widehat{\phi}_{HT_1}$ are not defined. In practice, the situation where $\sum_{i=1}^n r_i = 0$ is of no interest. In numerical experiments, when the simulated sample size is relatively small, $\sum_{i=1}^n r_i = 0$ may happen by chance. Then, to avoid numerical computer errors, we need to assign some values to these estimators when $\sum_{i=1}^n r_i = 0$. However, which values to assign should not be a critical issue, since with reasonably large sample size this situation may not happen at all. More about this issue will be discussed in Chapter 4.

2.2 Bayesian inference using Gaussian process models

This section will give a general introduction to Bayesian inference using Gaussian process models and briefly discuss how Gaussian process models can be applied to the problem considered in this thesis. Details on how to derive the posterior estimator for the population mean of the response variable and how to implement Gaussian process models through Markov chain Monte Carlo (MCMC) algorithms will be given in Chapter 3.

2.2.1 Gaussian process models

Gaussian process models are Bayesian models with Gaussian process priors. A Gaussian process is a stochastic process whose values at any finite set of points have a multivariate Gaussian distribution. All these multivariate Gaussian distributions must be compatible with each other, in the sense that they produce the same marginal distributions for the same subsets of points. A Gaussian process is entirely specified through a mean function and a covariance function, as a finite-dimensional multivariate Gaussian distribution is entirely specified by a mean vector and a covariance matrix.

Gaussian process models have long been used for Bayesian regression analysis where the regression predictors are assigned Gaussian process priors. Suppose $h(\mathbf{x})$ is a function of interest for some random variable z. For example, $h(\mathbf{x})$ can be the mean function of the response variable y or the (selection) probability function of the selection indicator r. $h(\mathbf{x})$ can often be modeled through a latent function $g(\mathbf{x})$ (i.e. the predictor) as

$$h(\mathbf{x}) = \widetilde{h}(g(\mathbf{x})) \tag{2.42}$$

where $\widetilde{h}$ is the link function from $g(\mathbf{x})$ to $h(\mathbf{x})$. When z is binary, $\widetilde{h}$ can be, for example, the expit function such that

$$h(\mathbf{x}) = \frac{1}{1 + \exp(-g(\mathbf{x}))}. \tag{2.43}$$

When z is numerical, $\widetilde{h}$ is typically the identity function such that

$$h(\mathbf{x}) = g(\mathbf{x}). \tag{2.44}$$

In either case, the latent function g can be assigned Gaussian process priors. Then inference about $h(\mathbf{x})$ will be made through modeling $g(\mathbf{x})$.

In practice, it is typical to let the prior mean of $g(\mathbf{x})$ equal zero for all $\mathbf{x}$, unless specific prior information is available. When $\mathrm{E}[g(\mathbf{x})] = 0$ *a priori* for all $\mathbf{x}$, the Gaussian process model is determined by its covariance function

$$C(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{Cov}\left(g(\mathbf{x}_i), g(\mathbf{x}_j)\right). \tag{2.45}$$

Typical covariance functions for Gaussian process models (Rasmussen and William, 2006; Neal, 1998) include

$$C(\mathbf{x}_i, \mathbf{x}_j; \sigma_0^2,\ \lambda^2,\ \eta^2,\ \ell,\ r) = \sigma_0^2 + \sum_{k=1}^{d} \lambda_k^2 x_{ik} x_{jk} + \eta^2 \exp\left\{ - \left(\sum_{k=1}^{d} \left(\frac{x_{ik} - x_{jk}}{\ell_k} \right)^2 \right)^{r/2} \right\} \tag{2.46}$$

and

$$C(\mathbf{x}_i, \mathbf{x}_j; \sigma_0^2,\ \lambda^2,\ \eta^2,\ \ell,\ r) = \sigma_0^2 + \sum_{k=1}^{d} \lambda_k^2 x_{ik} x_{jk} + \eta^2 \exp\left\{ - \sum_{k=1}^{d} \left(\frac{|x_{ik} - x_{jk}|}{\ell_k} \right)^{r} \right\} \tag{2.47}$$

where

$$\sigma_0^2,\ \ \lambda^2 = (\lambda_1^2, \dots, \lambda_d^2),\ \ \eta^2,\ \ \ell = (\ell_1, \dots, \ell_d),\ \ \text{and}\ \ r \tag{2.48}$$

are hyperparameters which can be either fixed or assigned higher level priors. Note that the two types of covariance functions above are equivalent when $r = 2$ or $d = 1$.

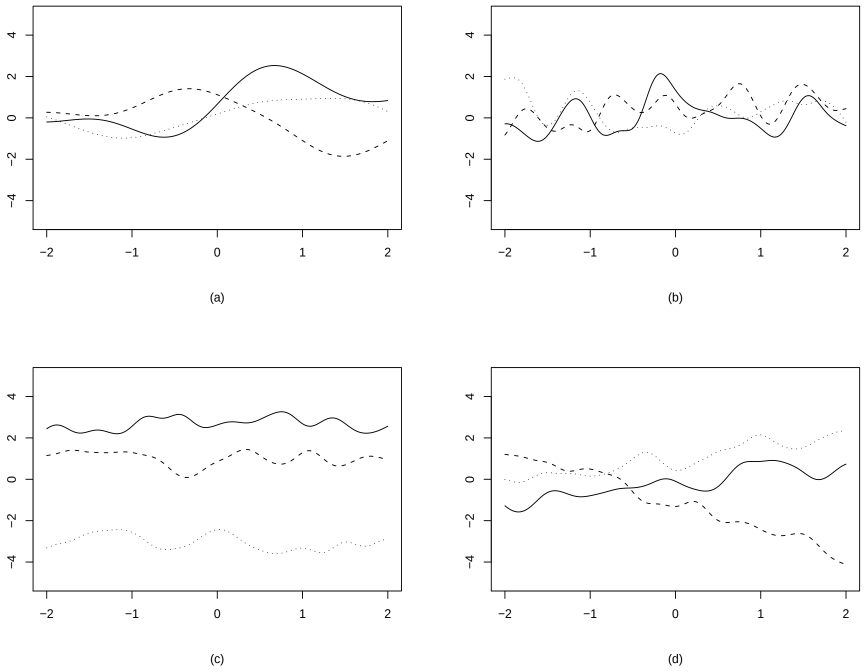
With a covariance function given by (2.46) or (2.47), a Gaussian process model can be denoted by

$$\mathcal{GP}(\sigma_0^2,\ \lambda^2,\ \eta^2,\ \ell,\ r) \tag{2.49}$$

It has been well-known (Rasmussen and William, 2006; Neal, 1998) that when $0 < r \leq 2$, both of these types of covariance functions are positive semi-definite. When $r = 2$, the corresponding (random) functions produced by these covariance functions are analytic and therefore differentiable to infinite order.

The covariance function is the crucial ingredient of a Gaussian process model, as it defined the functions that can be fit by the observed data. The first two terms of the covariance function in (2.46) or (2.47) are equivalent to a linear regression model and will be explained more in the next subsection. The exponential term, the key component of the covariance function, determines all the nonlinear effects and interactions in the function it can model, with the length-scale hyperparameters ℓ's controlling the relevance of each covariate and the exponent r controlling the smoothness of the produced function. The overall scaling hyperparameter η controls the magnitude of the exponential component or the marginal variance of the function produced with the exponential component only.

Some example functions produced by a one-dimensional ($d = 1$) Gaussian process model with the covariance function (2.46) are illustrated in Figure 2.1. As shown in Figure 2.1 (a) and (b), functions produced by an exponential covariance function with a length-scale ℓ equal to 1 look less wiggly than those with a length-scale ℓ equal to 0.3, since with a larger length-scale, distant x's are more correlated. In Figure 2.1 (c), a constant term $\sigma_0^2 = 1.5^2$ is added to the covariance function, resulting a (random) vertical shift to the functions that would have been produced without it. In addition to the constant term, a linear term is included as in Figure 2.1 (d) where all the functions exhibit a (random) linear trend.



(a)

(b)

(c)

(d)

Figure 2.1: Three sample functions from each of the following Gaussian process priors: (a) $C(x_i, x_j) = \exp\{-(x_i - x_j)^2\}$; (b) $C(x_i, x_j) = \exp\{-(\frac{x_i - x_j}{0.3})^2\}$; (c) $C(x_i, x_j) = 1.5^2 + 0.4^2 \exp\{-(\frac{x_i - x_j}{0.3})^2\}$; (d) $1.5^2 + 1.2^2 x_i x_j + 0.4^2 \exp\{-(\frac{x_i - x_j}{0.3})^2\}$.

In this thesis, a slightly modified version of (2.47) will be considered as

$$C(\mathbf{x}_i, \mathbf{x}_j; \sigma_0^2,\ \lambda^2,\ \eta^2,\ \ell,\ r) = \sigma_0^2 + \frac{1}{d}\sum_{k=1}^{d} \lambda_k^2 x_{ik} x_{jk} + \eta^2 \exp\left\{-\frac{1}{d}\sum_{k=1}^{d}\left(\frac{|x_{ik} - x_{jk}|}{\ell_k}\right)^r\right\} \tag{2.50}$$

This covariance function has $\lambda_k^2 x_{ik} x_{jk}$ and $\left(\frac{|x_{ik} - x_{jk}|}{\ell_k}\right)^r$ averaged over the d dimensions. The merit of having the averages instead of the sums of $\lambda_k^2 x_{ik} x_{jk}$'s and $\left(\frac{|x_{ik}-x_{jk}|}{\ell_k}\right)^r$'s is that the correlation between $g(\mathbf{x}_i)$ and $g(\mathbf{x}_j)$ will be more stable as the dimensionality d changes, assuming that $x_1, x_2, \ldots, x_d$ have the same marginal distributions for each value of d, the linear coefficients $\lambda_1, \lambda_2 \ldots, \lambda_d$ have the same marginal priors, and the length-scale hyperparameters $l_1, l_2 \ldots, l_d$ have the same marginal priors.

2.2.2 Connection to conventional regression

Gaussian process regression models are more sophisticated and flexible than conventional regression models in that it can model complex functions without incurring overfitting issues. However, it still has some connection to the classical regression models. Indeed, the classical models can be considered as simple cases of the Gaussian process models. To illustrate the connection, consider the following multiple linear regression model.

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_d x_{id} + \epsilon_i, \;\; i = 1, \ldots, n && (2.51) \\ &= f(\mathbf{x}_i) + \epsilon_i, \;\; i = 1, \ldots, n && (2.52) \end{aligned}$$

where

$$f(\mathbf{x}_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_d x_{id} \tag{2.53}$$

is the linear predictor at the observation $\mathbf{x}_i$ and the noises ϵ_i's are i.i.d. with normal distribution $\mathcal{N}(0, \delta^2)$. Suppose we assign independent Gaussian priors to the parameters $\beta_i, \; i = 0, 1, \ldots, d$, with zero means and variances $\sigma_i^2, \; i = 0, 1, \ldots, d$. That is,

$$\beta_0 \sim \mathcal{N}(0, \sigma_0^2) \;\perp\!\!\!\perp\; \beta_1 \sim \mathcal{N}(0, \sigma_1^2) \;\perp\!\!\!\perp\; \cdots \;\perp\!\!\!\perp\; \beta_d \sim \mathcal{N}(0, \sigma_d^2) \tag{2.54}$$

Then, given the covariates $\mathbf{x}_i = (x_{i1}, \ldots, x_{id}), \; i = 1, 2, \ldots, n$, the linear predictors at each $\mathbf{x}_i$ will have a multivariate normal prior with a zero mean vector and a covariance matrix equal to

$$\begin{aligned} & \mathrm{Cov}\left([\, f(\mathbf{x}_1), \, f(\mathbf{x}_2), \, \ldots, \, f(\mathbf{x}_n) \,]^T \right) \\ = \; & \mathrm{Cov}(\mathbf{X}\beta) = \mathbf{X}\mathrm{Cov}(\beta)\mathbf{X}^T = \left[\sigma_0^2 + \sum_{k=1}^{d} \sigma_i^2 x_{ik} x_{jk} \right]_{i,j}, && (2.55) \end{aligned}$$

where

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nd} \end{pmatrix}. \tag{2.56}$$

Clearly, the above covariance matrix corresponds to the first two terms of the Gaussian process covariance functions given by (2.46), (2.47) and (2.50). This correspondence also explains why the first two terms of those covariance functions superimpose a constant shift and a linear trend on the functions produced with only the exponential part of the covariance function as shown in Figure 2.1 (c) and (d). The exponential component of the covariance function produces more complex functions than a single linear regression model does, with the overall scaling hyperparameter η controlling the magnitude of this exponential part of the whole function. Because of its flexibility, Gaussian process models with covariance functions defined by (2.46), (2.47) or (2.50), are considered non-parametric models.

2.2.3 Modeling two correlated functions

For the problem considered in this thesis, we need to model not only the mean function $\mu(\mathbf{x})$ of y, but also the selection probability function $\nu(\mathbf{x})$ and their correlation. Two strategies for incorporating the selection probability through Gaussian process models will be discussed in this subsection and next subsection, respectively.

Since $\nu(\mathbf{x})$ may be correlated with $\mu(\mathbf{x})$, assigning dependent priors may be an effective way for modeling the correlation between these two functions. Let g_μ and g_ν be the latent functions corresponding to $\mu(\mathbf{x})$ and $\nu(x)$, respectively. That is,

$$\mu(\mathbf{x}) = \tilde{\mu}(g_\mu) = \tilde{\mu}(g_\mu(x)) \tag{2.57}$$

$$\nu(\mathbf{x}) = \tilde{\nu}(g_\nu) = \tilde{\nu}(g_\nu(x)) \tag{2.58}$$

where $\tilde{\mu}$ and $\tilde{\nu}$ are the link functions from the latent functions g_μ and g_ν to $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$, respectively. As mentioned earlier, when y is numerical, $\tilde{\mu}$ is simply the identity function. Next, define g_μ and g_ν by

$$g_\mu = g_1 + g_0 \quad \textit{and} \quad g_\nu = g_2 + g_0 \tag{2.59}$$

where, given hyperparameters, g_1, g_2 and g_0 are functions with independent Gaussian process priors.

That is,

$$\begin{aligned} g_1 &\sim \mathcal{GP}_1 = \mathcal{GP}(\sigma_{0,1}^2,\ \lambda_1^2,\ \eta_1^2,\ \ell_1,\ r_1)\ \perp\!\!\!\perp \\ g_2 &\sim \mathcal{GP}_2 = \mathcal{GP}(\sigma_{0,2}^2,\ \lambda_2^2,\ \eta_2^2,\ \ell_2,\ r_2)\ \perp\!\!\!\perp \\ g_0 &\sim \mathcal{GP}_0 = \mathcal{GP}(\sigma_{0,0}^2,\ \lambda_0^2,\ \eta_0^2,\ \ell_0,\ r_0) \end{aligned} \tag{2.60}$$

given $\sigma_{0,h}^2,\ \lambda_h^2 = (\lambda_{h1}^2, \ldots, \lambda_{hd}^2),\ \eta_h^2,\ \ell_h = (\ell_{h1}, \ldots, \ell_{hd}),\ r_h,\ h = 1, 2, 0$. In general, $g_1,\ g_2$ and g_0 may not necessarily be marginally independent, since $g_1,\ g_2$ and g_0 may have some of the hyperparameters equal, or their hyperparameters may be dependent *a priori* with higher level priors. With the above strategy, g_μ and g_ν are correlated *a priori* through g_0.

For particular covariates $\mathbf{x}_1, \cdots, \mathbf{x}_n$, denote the corresponding values of the latent functions g_μ and g_ν by

$$\mathbf{g}_\mu^{(n)} = \begin{pmatrix} g_\mu(\mathbf{x}_1) \\ g_\mu(\mathbf{x}_2) \\ \vdots \\ g_\mu(\mathbf{x}_n) \end{pmatrix} = \begin{pmatrix} g_{\mu,1} \\ g_{\mu,2} \\ \vdots \\ g_{\mu,n} \end{pmatrix} \quad \text{and} \quad \mathbf{g}_\nu^{(n)} = \begin{pmatrix} g_\nu(\mathbf{x}_1) \\ g_\nu(\mathbf{x}_2) \\ \vdots \\ g_\nu(\mathbf{x}_n) \end{pmatrix} = \begin{pmatrix} g_{\nu,1} \\ g_{\nu,2} \\ \vdots \\ g_{\nu,n} \end{pmatrix}. \tag{2.61}$$

By the strategy (2.59), the latent vectors $\mathbf{g}_\mu^{(n)}$ and $\mathbf{g}_\nu^{(n)}$ will have the following joint multivariate Gaussian distribution

$$\begin{pmatrix} \mathbf{g}_\mu^{(n)} \\ \mathbf{g}_\nu^{(n)} \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} K_1 + K_0 & K_0 \\ K_0 & K_2 + K_0 \end{pmatrix}\right) \tag{2.62}$$

where

$$K_h = \left[C(\mathbf{x}_i, \mathbf{x}_j; \sigma_{0,h}^2,\ \lambda_h^2,\ \eta_h^2,\ \ell_h,\ r_h)\right]_{i,j},\quad h = 1, 2, 0 \tag{2.63}$$

with the covariance function $C(\mathbf{x}_i, \mathbf{x}_j; \sigma_{0,h}^2,\ \lambda_h^2,\ \eta_h^2,\ \ell_h,\ r_h)$ defined by (2.46), (2.47) or (2.50).

When the response variable y is real-valued, a noise term should be added into the regression model as

$$y = \mu(\mathbf{x}) + \epsilon = g_\mu(\mathbf{x}) + \epsilon \tag{2.64}$$

where ϵ typically has a Gaussian distribution $\mathcal{N}(0, \delta^2)$. The noise standard deviation δ can either be a fixed constant or be adaptable with a prior distribution that is independent of the latent function g_μ.

The strategy described above, however, has its limitation. Particularly, by (2.59), prior correlations between $g_\mu(\mathbf{x})$ and $g_\nu(\mathbf{x})$ are always positive for all $\mathbf{x}$'s. If instead, let $g_\nu = g_2 - g_0$, all the correlations will be negative. However, in practice, we may not know whether g_μ and g_ν should be positively or negatively correlated. Nevertheless, this scheme is simple and yet will help reveal the fundamental issues involved. Some discussion on how to expand this strategy for more general situations are given in the last chapter of the thesis.

2.2.4 Using selection probability as a covariate

Instead of modeling $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ jointly with dependent priors, one can alternatively use the selection probability $\nu(\mathbf{x})$ or a transformation of it as an additional covariate. More specifically, let $x_{d+1} = h(\nu(\mathbf{x}))$ and $\mathbf{x}^* = (x_1, x_2, \ldots, x_d, x_{d+1})$, where h is some inversible transformation. (Note that non-inversible h is also useful, but $\nu(\mathbf{x})$ would not be fully exploited.) With $\mathbf{x}^*$ being the covariate vector, $\mu(\mathbf{x})$ can be re-written as

$$\mu(\mathbf{x}) = \tilde{\mu}(g_\mu(\mathbf{x})) = \tilde{\mu}\left(g_\mu^*\left(\mathbf{x}, h\left(\nu(\mathbf{x})\right)\right)\right) = \mu^*(\mathbf{x}^*) \tag{2.65}$$

where g_μ is a function with d arguments, $x_1, \ldots, x_d$, while g_μ^* has $d+1$ arguments, $x_1, \ldots, x_d, x_{d+1}$. Note that it is g_μ^* that will be assigned Gaussian process priors with the $(d+1)$-dimensional covariate vector $\mathbf{x}^*$. With a Gaussian process model, the selection of h is less crucial compared to some existing methods that also use $h(\nu(\mathbf{x}))$ as an additional covariate, since the Gaussian process model will automatically figure out the best relationship between $\mu^*(\mathbf{x}^*)$ and its covariate $x_{d+1} = h\left(\nu(\mathbf{x})\right)$. Particularly, $h\left(\nu(\mathbf{x})\right)$ does not have to be the inverse selection probability that is popularly used in the literature.

Compared to the strategy (2.59) described in the previous subsection, this strategy only has one Gaussian process model and therefore is conceivably easier to implement and faster to compute. However, with the selection probability as an additional covariate, this strategy requires knowledge of the selection probability function $\nu(\mathbf{x})$ for at least the observed $\mathbf{x}$'s. It is also often desired that $\mu^*(\mathbf{x}^*)$ can be predicted at all $\mathbf{x}^*$ or at a fairly large number of $\mathbf{x}^*$ so that the error due to approximating an integral by a finite sum can be reduced to a minimum degree by averaging $\mu^*(\mathbf{x}^*)$ over all these $\mathbf{x}^*$. The strategy defined by (2.59), however, does not require knowing any ν values by modeling the selection probability function $\nu(\mathbf{x})$ simultaneously, although knowing some ν values may help improve the efficiency of the estimation. Therefore, the strategy (2.59) has wider applications.

When the selection probability ν is used as an additional covariate, the covariance function (2.50) becomes

$$\begin{aligned} C(\mathbf{x}_i^*, \mathbf{x}_j^*; \sigma_0^2,\, \lambda^2,\, \eta^2,\, \ell,\, r) &= \sigma_0^2 + \frac{1}{d}\sum_{k=1}^{d} \lambda_k^2 x_{ik} x_{jk} + \lambda_{d+1}^2 x_{i,d+1} x_{j,d+1} \\ &+ \eta^2 \exp\left\{ -\frac{1}{d}\sum_{k=1}^{d} \left(\frac{|x_{ik} - x_{jk}|}{\ell_k}\right)^r - \left(\frac{|x_{i,d+1} - x_{j,d+1}|}{\ell_{d+1}}\right)^r \right\} \end{aligned} \tag{2.66}$$

where $\mathbf{x}_i^* = (x_{i1}, \ldots, x_{id}, x_{i,d+1})$ and $\mathbf{x}_j^* = (x_{j1}, \ldots, x_{jd}, x_{j,d+1})$ with $x_{i,d+1} = h(\nu(\mathbf{x}_i))$ and $x_{j,d+1} = h(\nu(\mathbf{x}_j))$ being the additional covariates and h being some inverse function. Note that $x_{i,d+1}$ and $x_{j,d+1}$ are not included in count of covariates for scaling $\lambda_k^2 x_{ik} x_{jk}$ and $\left(\frac{|x_{ik}-x_{jk}|}{\ell_k}\right)^r$. And for $k = 1, \ldots, d$, $\lambda_k^2 x_{ik} x_{jk}$ and $\left(\frac{|x_{ik}-x_{jk}|}{\ell_k}\right)^r$ are still scaled by d instead of $d+1$, so that the approach that uses the selection probability as a covariate is more directly comparable to the approaches that do not, in the sense that they treat the d-dimensional covariate vector $\mathbf{x}$ the same way.

Chapter 3

Implementing Gaussian process models

In Bayesian analysis, it is typical to estimate an unknown quantity, e.g. the population mean ϕ, by its posterior mean value given the observed data. However, with Gaussian process priors, one can rarely obtain the analytical form for the posterior distribution of the quantity of interest or for the posterior distribution of the corresponding latent function. Therefore, Monte Carlo methods, typically Markov chain Monte Carlo (MCMC) sampling, are essential for implementing the Gaussian process models. This chapter first discusses how inference can be made for the population mean ϕ, assuming an MCMC sample are already obtained from the posterior distribution of the latent vectors $\mathbf{g}_\mu^{(n)}$ and $\mathbf{g}_\nu^{(n)}$, where $\mathbf{g}_\mu^{(n)}$ and $\mathbf{g}_\nu^{(n)}$ are defined as in (2.61). Description of the adopted MCMC sampling schemes and how to implement them follow next.

3.1 Inference from the posterior distribution

This section first derives the formulas for the estimators for the population mean based on the Gaussian process models discussed in Section 2.2, and then analyzes the different sources of errors involved in these estimators.

3.1.1 Obtaining the estimators for the population mean

In this subsection, the formula for the estimator for the population mean ϕ is first derived under the strategy (2.59) as in Subsection 2.2.3, with both the mean function μ and the selection probability function ν being modeled. The formulas for the estimator under the same strategy (2.59) but with ν known, for the estimator that ignores ν, and for the estimator that uses the known ν as a covariate as in Subsection 2.2.4 are special cases of the first formula as will be illustrated later.

Recall that

$$\phi = \int \mu(\mathbf{x})\, dF_{\mathbf{X}} = \int \tilde{\mu}(g_\mu(\mathbf{x}))\, dF_{\mathbf{X}} = \phi(g_\mu) \tag{3.1}$$

which is a functional of the latent function g_μ. Therefore, the estimation of the posterior mean of ϕ will be based on the posterior distribution of the latent vectors $\mathbf{g}_\mu^{(n)}$ and $\mathbf{g}_\nu^{(n)}$ and the conditional distribution of the latent function g_μ given the latent vectors $\mathbf{g}_\mu^{(n)}$ and $\mathbf{g}_\nu^{(n)}$. Given the observations D_n : $(\mathbf{x}_1, y_1, r_1),\ \ldots,\ (\mathbf{x}_n, y_1, r_n)$, denote the joint posterior distribution of $\mathbf{g}_\mu^{(n)}$ and $\mathbf{g}_\nu^{(n)}$ by $P(\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)} | D_n)$. Given the hyperparameters $\sigma_{0,h}^2,\ \lambda_h^2 = (\lambda_{h1}^2, \ldots, \lambda_{hd}^2),\ \eta_h^2,\ \ell_h = (\ell_{h1}, \ldots, \ell_{hd}),\ r_h,\ h = 1, 2, 0$, denote the conditional distribution of the latent function g_μ given the latent vectors $\mathbf{g}_\mu^{(n)}$ and $\mathbf{g}_\nu^{(n)}$ by $P(g_\mu | \mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)})$. Note that $P(g_\mu | \mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)})$ is a functional of the latent function g_μ. For any particular $\mathbf{x} = (x_1, \ldots, x_d)$, $P(g_\mu \mid \mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)})$ reduces to the conditional distribution of the latent variable $g_\mu(\mathbf{x})$ given $\mathbf{g}_\mu^{(n)}$ and $\mathbf{g}_\nu^{(n)}$, which is

$$P(g_\mu(\mathbf{x}) \mid \mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}) \sim \mathcal{N}\left(m_c\left(\mathbf{x}; \mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}\right), v_c(\mathbf{x})\right) \tag{3.2}$$

where

$$m_c\left(\mathbf{x}; \mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)})\right) = \begin{pmatrix} \mathbf{k}_{\mathbf{x},\mu}^{(n)} \\ \mathbf{k}_{\mathbf{x},\nu}^{(n)} \end{pmatrix}^T \begin{pmatrix} K_1 + K_0 & K_0 \\ K_0 & K_2 + K_0 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{g}_\mu^{(n)} \\ \mathbf{g}_\nu^{(n)} \end{pmatrix} \tag{3.3}$$

and

$$v_c(\mathbf{x}) = k_{\mathbf{x},\mu} - \begin{pmatrix} \mathbf{k}_{\mathbf{x},\mu}^{(n)} \\ \mathbf{k}_{\mathbf{x},\nu}^{(n)} \end{pmatrix}^T \begin{pmatrix} K_1 + K_0 & K_0 \\ K_0 & K_2 + K_0 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{k}_{\mathbf{x},\mu}^{(n)} \\ \mathbf{k}_{\mathbf{x},\nu}^{(n)} \end{pmatrix} \tag{3.4}$$

with $K_h,\ h = 1, 2, 0$, defined as in (2.63) and

$$\begin{aligned}
k_{\mathbf{x},\mu} &= \mathrm{Var}(g_\mu(\mathbf{x})) = C(\mathbf{x}, \mathbf{x}; \sigma^2_{0,1}, \lambda^2_1, \eta^2_1, \ell_1, r_1) \\
&= \sigma^2_{0,1} + \sum_{k=1}^{d} \lambda^2_{1k} x^2_k + \eta^2_1 && (3.5) \\
\mathbf{k}^{(n)}_{\mathbf{x},\mu} &= \left[\mathrm{Cov}(g_\mu(\mathbf{x}), g_\mu(\mathbf{x}_i))\right]_i \\
&= \left[C(\mathbf{x}, \mathbf{x}_i; \sigma^2_{0,1}, \lambda^2_1, \eta^2_1, \ell_1, r_1) + C(\mathbf{x}, \mathbf{x}_i; \sigma^2_{0,0}, \lambda^2_0, \eta^2_0, \ell_0, r_0)\right]_i && (3.6) \\
\mathbf{k}^{(n)}_{\mathbf{x},\nu} &= \left[\mathrm{Cov}(g_\mu(\mathbf{x}), g_\nu(\mathbf{x}_i))\right]_i \\
&= \left[C(\mathbf{x}, \mathbf{x}_i; \sigma^2_{0,0}, \lambda^2_0, \eta^2_0, \ell_0, r_0)\right]_i && (3.7)
\end{aligned}$$

Then the posterior mean of ϕ, denoted by $\phi_{post} = \mathrm{E}[\phi|D_n]$, can be expressed as

$$\begin{aligned}
\phi_{post} &= \int \left(\int \left(\int \tilde{\mu}(g_\mu(\mathbf{x}))\, dF_{\mathbf{X}}(\mathbf{x}) \right) dP(g_\mu \mid \mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu) \right) dP(\mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu | D_n) \\
&= \int \left(\int \left(\int \tilde{\mu}(g_\mu(\mathbf{x}))\, dP(g_\mu \mid \mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu) \right) dF_{\mathbf{X}}(\mathbf{x}) \right) dP(\mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu | D_n) \\
&= \int \left(\int \mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x})) \mid \mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu \right] dF_{\mathbf{X}}(\mathbf{x}) \right) dP(\mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu | D_n) && (3.8)
\end{aligned}$$

Clearly three steps of computations are involved in obtaining ϕ_{post}. First, we need to obtain

$$\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x})) \mid \mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu \right] = \int \tilde{\mu}(g_\mu(\mathbf{x}))\, dP(g_\mu(\mathbf{x}) \mid \mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu) \qquad (3.9)$$

for each $\mathbf{x}$. According to (3.2), when $\tilde{\mu}$ is the identity function, we simply have

$$\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x})) \mid \mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu \right] = m_c\left(\mathbf{x}; \mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu \right) \qquad (3.10)$$

where $m_c\left(\mathbf{x}; \mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu \right)$ is as in (3.3). When $\tilde{\mu}$ is the probit function, i.e. $\tilde{\mu} = \Phi$, analytic result can be obtained for $\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x})) \mid \mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu \right]$ as

$$\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x})) \mid \mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu \right] = \Phi\left(\frac{m_c\left(\mathbf{x}; \mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu \right)}{1 + v_c(\mathbf{x})} \right) \qquad (3.11)$$

where $m_c\left(\mathbf{x}; \mathbf{g}^{(n)}_\mu, \mathbf{g}^{(n)}_\nu \right)$ and $v_c(\mathbf{x})$ are as in (3.3) and (3.4), respectively.

Proof of (3.11) Consider

$$\begin{aligned}
&\frac{1}{\sqrt{2\pi}\sigma}\int \Phi(x)\exp\left\{-\frac{1}{2\sigma^2}(x-\theta)^2\right\}\,dx \\
=\;& \frac{1}{2\pi\sigma}\int_{-\infty}^{\infty}\int_{-\infty}^{x}\exp\left\{-\frac{1}{2}y^2-\frac{1}{2\sigma^2}(x-\theta)^2\right\}dy\,dx \\
=\;& \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\sigma u+\theta}\exp\left\{-\frac{1}{2}y^2-\frac{1}{2}u^2\right\}dy\,du,\quad \text{where } u=\frac{x-\theta}{\sigma} \\
=\;& \int_{-\infty}^{\infty}\int_{-\infty}^{\frac{\theta}{\sqrt{1+\sigma^2}}}\phi(w)\phi(v)\,dw\,dv,\quad \text{where } w=\frac{y-\sigma u}{\sqrt{1+\sigma^2}},\ v=\frac{\sigma y+u}{\sqrt{1+\sigma^2}} \\
=\;& \Phi\left(\frac{\theta}{\sqrt{1+\sigma^2}}\right)
\end{aligned} \tag{3.12}$$

This completes the proof of 3.11.

In general, (3.9) can not be obtained analytically. Then $\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x}))\mid \mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right]$ needs to be estimated by numerical methods. One good scheme for estimating $\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x}))\mid \mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right]$ is to average $\tilde{\mu}(g_\mu(\mathbf{x}))$ over equally spaced quantiles of $P\left(g_\mu(\mathbf{x})\mid \mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right)$. Note that, since $P\left(g_\mu(\mathbf{x})\mid \mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right)$ is a one-dimensional normal distribution, its quantiles can be easily obtained. Let $\tilde{g}_1,\ldots,\tilde{g}_{\tilde{n}}$ be the $\frac{0.5}{\tilde{n}}$, $\frac{1.5}{\tilde{n}}$, $\ldots$, $\frac{\tilde{n}-0.5}{\tilde{n}}$ quantiles of $P\left(g_\mu(\mathbf{x})\mid \mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right)$. Then,

$$\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x}))\mid \mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right]\approx\frac{1}{\tilde{n}}\sum_{s=1}^{\tilde{n}}\tilde{\mu}(\tilde{g}_s) \tag{3.13}$$

The error of estimating $\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x}))\mid \mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right]$ by (3.13) converges to zero at a rate proportional to $\frac{1}{\tilde{n}^2}$.

Second, we need to estimate

$$\phi_{cond}\left(\mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right)=\int \mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x}))\mid \mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right]\,dF_{\mathbf{X}}(\mathbf{x}) \tag{3.14}$$

where $\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x})|\mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right]$ is either in its exact form as in (3.10) or (3.11) or in the estimated form as in (3.13). When $F_{\mathbf{X}}$ is available, we may sample $\mathbf{x}_1^*,\,\cdots,\,\mathbf{x}_N^*\overset{iid}{\sim}F_{\mathbf{X}}$ with N much larger than the sample size n and then estimate $\phi_{cond}\left(\mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right)$ by averaging over $\mathbf{x}_j^*$ as

$$\phi_{cond}\left(\mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right)\quad\approx\quad\frac{1}{N}\sum_{j=1}^{N}\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x}_j^*))\mid \mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right] \tag{3.15}$$

When $F_{\mathbf{X}}$ is not available, $\phi_{cond}\left(\mathbf{g}_\mu^{(n)},\mathbf{g}_\nu^{(n)}\right)$ will be estimated using the observed $\mathbf{x}$ only and (3.15) becomes

$$\phi_{cond}\left(\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}\right) \quad \approx \quad \frac{1}{n}\sum_{j=1}^{n} \tilde{\mu}(g_\mu(\mathbf{x}_j)) \tag{3.16}$$

where $(g_\mu(\mathbf{x}_1), \ldots, g_\mu(\mathbf{x}_n))^T = \mathbf{g}_\mu^{(n)}$ and the step of computing $\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x})|\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}\right]$ has been skipped.

Third, we need to estimate

$$\phi_{post} = \int \phi_{cond}\left(\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}\right)\, dP(\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}|D_n) \tag{3.17}$$

where $\phi_{cond}\left(\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}\right)$ is as in (3.14). Estimating $\int \phi_{cond}\left(\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}\right)\, dP(\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}|D_n)]$ requires an MCMC sample of $\mathbf{g}_\mu^{(n)}$ and $\mathbf{g}_\nu^{(n)}$ given the observations D_n. Suppose we have the following MCMC sample of size B drawn from $P(\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}|D_n)$

$$\begin{pmatrix} \mathbf{g}_{\mu,1}^{(n)} \\ \mathbf{g}_{\nu,1}^{(n)} \end{pmatrix}, \begin{pmatrix} \mathbf{g}_{\mu,2}^{(n)} \\ \mathbf{g}_{\nu,2}^{(n)} \end{pmatrix}, \ldots, \begin{pmatrix} \mathbf{g}_{\mu,B}^{(n)} \\ \mathbf{g}_{\nu,B}^{(n)} \end{pmatrix} \tag{3.18}$$

where

$$\mathbf{g}_{\mu,b}^{(n)} = \begin{pmatrix} g_{\mu,1,b} \\ g_{\mu,2,b} \\ \vdots \\ g_{\mu,n,b} \end{pmatrix} \quad \text{and} \quad \mathbf{g}_{\nu,b}^{(n)} = \begin{pmatrix} g_{\nu,1,b} \\ g_{\nu,2,b} \\ \vdots \\ g_{\nu,n,b} \end{pmatrix}, \ b = 1, 2, \ldots, B. \tag{3.19}$$

Then ϕ_{post} can be estimated by

$$\phi_{post} = \int \phi_{cond}\left(\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}\right)\, dP(\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}|D_n) \approx \frac{1}{B}\sum_{b=1}^{B} \phi_{cond}\left(\mathbf{g}_{\mu,b}^{(n)}, \mathbf{g}_{\nu,b}^{(n)}\right) \tag{3.20}$$

where $\phi_{cond}\left(\mathbf{g}_{\mu,b}^{(n)}, \mathbf{g}_{\nu,b}^{(n)}\right)$'s can be estimated by either (3.15) or (3.16).

Now, we have obtained all the three steps of computations for the posterior mean ϕ_{post} of ϕ. The three steps are summarized in the following formulas with $\widehat{\phi}_{post}$ denoting the estimator for ϕ_{post}. When sampling additional $\mathbf{x}_1^*, \ldots, \mathbf{x}_N^*$ from $F_\mathbf{X}$ is possible, we have

$$\widehat{\phi}_{post} = \begin{cases} \frac{1}{B}\sum_{b=1}^{B} \frac{1}{N}\sum_{j=1}^{N} m_c\left(\mathbf{x}_j^*; \mathbf{g}_{\mu,b}^{(n)}, \mathbf{g}_{\nu,b}^{(n)}\right), & \tilde{\mu} \text{ is identity} \\ \frac{1}{B}\sum_{b=1}^{B} \frac{1}{N}\sum_{j=1}^{N} \Phi\left(\frac{m_c\left(\mathbf{x}_j^*; \mathbf{g}_{\mu,b}^{(n)}, \mathbf{g}_{\nu,b}^{(n)}\right)}{1+v_c(\mathbf{x}_j^*)}\right), & \tilde{\mu} \text{ is probit} \\ \frac{1}{B}\sum_{b=1}^{B} \frac{1}{N}\sum_{j=1}^{N} \frac{1}{\tilde{n}}\sum_{s=1}^{\tilde{n}} \tilde{\mu}(\tilde{g}_{j,b,s}), & \text{otherwise} \end{cases} \tag{3.21}$$

where $m_c\left(\mathbf{x}_j^*; \mathbf{g}_{\mu,b}^{(n)}, \mathbf{g}_{\nu,b}^{(n)}\right)$ and $v_c(\mathbf{x}_j^*)$ are as in (3.2) and $\tilde{g}_{j,b,1}, \ldots, \tilde{g}_{j,b,\tilde{n}}$ are the $\frac{0.5}{\tilde{n}}$, $\frac{1.5}{\tilde{n}}$, $\ldots$, $\frac{\tilde{n}-0.5}{\tilde{n}}$ quantiles of $P\left(g_\mu(\mathbf{x}_j^*) \mid \mathbf{g}_{\mu,b}^{(n)}, \mathbf{g}_{\nu,b}^{(n)}\right)$. When sampling $\mathbf{x}_1^*, \ldots, \mathbf{x}_N^*$ from $F_{\mathbf{X}}$ is not possible, we have

$$\widehat{\phi}_{post} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{n}\sum_{j=1}^{n}\tilde{\mu}(g_{\mu,j,b}) \tag{3.22}$$

where $(g_{\mu,1,b}, \ldots, g_{\mu,n,b})^T = \mathbf{g}_{\mu,b}^{(n)}$.

As noted, (3.21) and (3.22) are derived when both the mean function and the selection probability function ν are modeled by the strategy (2.59). By the same strategy (2.59), when ν is known, $\mathbf{g}_{\nu,b}^{(n)}$'s in (3.21) become the fixed $\mathbf{g}_\nu^{(n)}$ corresponding to the known values of ν at $\mathbf{x}_1, \ldots, \mathbf{x}_n$. When the selection probability is ignored, $P\left(g_\mu(\mathbf{x}_j^*) \mid \mathbf{g}_{\mu,b}^{(n)}, \mathbf{g}_{\nu,b}^{(n)}\right)$ reduces to $P\left(g_\mu(\mathbf{x}_j^*) \mid \mathbf{g}_{\mu,b}^{(n)}\right)$. Consequently $m_c\left(\mathbf{x}; \mathbf{g}_{\mu,b}^{(n)}, \mathbf{g}_{\nu,b}^{(n)}\right)$ reduces to $m_c\left(\mathbf{x}; \mathbf{g}_{\mu,b}^{(n)}\right)$ as the mean of $g_\mu(\mathbf{x})$ with respect to $P\left(g_\mu(\mathbf{x}_j^*) \mid \mathbf{g}_{\mu,b}^{(n)}\right)$; and $v_c(\mathbf{x}_j^*)$ also becomes the variance of $g_\mu(\mathbf{x})$ with respect to $P\left(g_\mu(\mathbf{x}_j^*) \mid \mathbf{g}_{\mu,b}^{(n)}\right)$. When the selection probability is used as a covariate, the formula (3.21) is in the same form as when the selection probability is ignored, except that the covariate vector $\mathbf{x}$ becomes $(d+1)$-dimensional with the additional covariate is an inversible transformation of ν as discussed in Subsection 2.2.4. In all situations, the formula (3.22) remains in the same form but with $\mathbf{g}_{\mu,b}^{(n)}$'s sampled from different posterior distributions.

3.1.2 Sources of errors involved in estimating the population mean

The three steps of computing $\widehat{\phi}_{post}$ involve different degrees of errors. When computing $\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x}))|\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}\right]$, if $\tilde{\mu}$ is the identity or the probit function, exact result can be obtained. When $\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x}))|\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}\right]$ is estimated using (3.13), the error of estimation converges to zero at a rate proportional to $\frac{1}{\tilde{n}^2}$. Since the time taken by estimating $\mathrm{E}\left[\tilde{\mu}(g_\mu(\mathbf{x}))|\mathbf{g}_\mu^{(n)}, \mathbf{g}_\nu^{(n)}\right]$ using (3.13) is negligible compared to the time taken by MCMC sampling, $\tilde{n}$ can be chosen arbitrarily large so that the error involved in this step is negligible compared to the errors involved in the other two steps.

The error of estimation involved in estimating (3.14) using (3.16) can not be controlled or evaluated, since the $\mathbf{x}_j$ used for estimation are fixed over MCMC iterations. Instead, when (3.15) is used for estimating (3.14), if N is large enough, the error involved in this step of estimation is negligible compared to the error due to MCMC sampling. However, when $\mathbf{x}$ is high-dimensional, sampling a sufficiently large number N of $\mathbf{x}_j^*$'s can be computationally costing (e.g. taking too much computer memory space). Alternatively, instead of using fixed $\mathbf{x}_1^*, \ldots, \mathbf{x}_N^*$ for all MCMC iterations, $\mathbf{x}_1^*, \ldots, \mathbf{x}_N^*$ can be sampled independently for each MCMC iteration. With independently selected $\mathbf{x}_j^*$'s, (3.21) becomes

$$\widehat{\phi}_{post} = \begin{cases} \frac{1}{B}\sum_{b=1}^{B}\frac{1}{N}\sum_{j=1}^{N} m_c\left(\mathbf{x}_{j,b}^*;\mathbf{g}_{\mu,b}^{(n)},\mathbf{g}_{\nu,b}^{(n)}\right), & \tilde{\mu} \text{ is identity} \\ \frac{1}{B}\sum_{b=1}^{B}\frac{1}{N}\sum_{j=1}^{N} \Phi\left(\frac{m_c\left(\mathbf{x}_{j,b}^*;\mathbf{g}_{\mu,b}^{(n)},\mathbf{g}_{\nu,b}^{(n)}\right)}{1+v_c(\mathbf{x}_{j,b}^*)}\right), & \tilde{\mu} \text{ is probit} \\ \frac{1}{B}\sum_{b=1}^{B}\frac{1}{N}\sum_{j=1}^{N}\frac{1}{\tilde{n}}\sum_{s=1}^{\tilde{n}}\tilde{\mu}(\tilde{g}_{j,b,s}^*), & \text{otherwise} \end{cases} \tag{3.23}$$

where $\mathbf{x}_{1,b}^*,\ldots,\mathbf{x}_{N,b}^*$ are the random sample of $\mathbf{x}$ at each iteration b and $\tilde{g}_{j,b,1}^*,\ldots,\tilde{g}_{j,b,\tilde{n}}^*$ are the corresponding quantiles of $P\left(g_\mu(\mathbf{x}_{j,b}^*) \mid \mathbf{g}_{\mu,b}^{(n)},\mathbf{g}_{\nu,b}^{(n)}\right)$. Denote

$$\widehat{\phi}_{cond,b} = \begin{cases} \frac{1}{N}\sum_{j=1}^{N} m_c\left(\mathbf{x}_{j,b}^*;\mathbf{g}_{\mu,b}^{(n)},\mathbf{g}_{\nu,b}^{(n)}\right), & \tilde{\mu} \text{ is identity} \\ \frac{1}{N}\sum_{j=1}^{N} \Phi\left(\frac{m_c\left(\mathbf{x}_{j,b}^*;\mathbf{g}_{\mu,b}^{(n)},\mathbf{g}_{\nu,b}^{(n)}\right)}{1+v_c(\mathbf{x}_{j,b}^*)}\right), & \tilde{\mu} \text{ is probit} \\ \frac{1}{N}\sum_{j=1}^{N}\frac{1}{\tilde{n}}\sum_{s=1}^{\tilde{n}}\tilde{\mu}(\tilde{g}_{j,b,s}^*), & \text{otherwise} \end{cases} \quad b = 1,\ldots,B \tag{3.24}$$

Then $\widehat{\phi}_{post}$ is the MCMC sample average of $\widehat{\phi}_{cond,1},\ldots,\widehat{\phi}_{cond,B}$, that is,

$$\widehat{\phi}_{post} = \frac{1}{B}\sum_{b=1}^{B}\widehat{\phi}_{cond,b}. \tag{3.25}$$

Since $\mathbf{x}_{j,b}^*$'s vary randomly at each iteration b, the error due to estimating (3.14) using (3.15) also varies randomly over the MCMC iterations. Then this source of error is part of the random uncertainty of $\widehat{\phi}_{cond,1},\ldots,\widehat{\phi}_{cond,B}$ and therefore can be evaluated through evaluating the standard deviation of $\widehat{\phi}_{cond,1},\ldots,\widehat{\phi}_{cond,B}$. The error due to estimating (3.17) using a finite MCMC sample is the major contributor to the overall error involved in computing $\widehat{\phi}_{post}$ by either (3.23) or (3.22). But the error due to MCMC sampling can also be evaluated through evaluating the standard deviation of $\widehat{\phi}_{cond,b}$'s.

The standard deviation of $\widehat{\phi}_{cond,b}$'s can be estimated as follows.

$$\begin{aligned} s.d.(\widehat{\phi}_{cond,b}) &\approx \sqrt{\frac{1}{B-\tau_B}\sum_{b=1}^{B}\left(\widehat{\phi}_{cond,b}-\widehat{\phi}_{post}\right)^2} \\ &= \sqrt{\frac{1}{B/\tau_B-1}\sum_{b=1}^{B}\frac{1}{\tau_B}\left(\widehat{\phi}_{cond,b}-\widehat{\phi}_{post}\right)^2} \end{aligned} \tag{3.26}$$

where τ_B is the autocorrelation time of $\{\widehat{\phi}_{cond,1},\,\cdots,\,\widehat{\phi}_{cond,B}\}$ and B/τ_B is the corresponding effective sample size of $\{\widehat{\phi}_{cond,1},\,\cdots,\,\widehat{\phi}_{cond,B}\}$. Then the standard error of $\widehat{\phi}_{post}$, i.e. the estimated standard deviation of $\widehat{\phi}_{post}$, can be obtained by

$$\begin{aligned} s.e.(\widehat{\phi}_{post}) &= s.e.\left(\frac{1}{B}\sum_{b=1}^{B}\widehat{\phi}_{cond,b}\right) \\ &= \sqrt{\frac{1}{B/\tau_B}\times\left(s.d.(\widehat{\phi}_{cond,b})\right)^2} \\ &\approx \sqrt{\frac{1}{B/\tau_B}\left(\frac{1}{B-\tau_B}\sum_{b=1}^{B}\left(\widehat{\phi}_{cond,b}-\widehat{\phi}_{post}\right)^2\right)}. \qquad (3.27) \end{aligned}$$

Various methods of estimating τ_B are available in the literature and will be discussed in the next section. It is noted that the error of estimation for τ_B has nothing to do with the standard deviation of $\widehat{\phi}_{post}$, but only affects the accuracy of the estimated standard deviation of $\widehat{\phi}_{post}$, $s.e.(\widehat{\phi}_{post})$.

3.2 MCMC algorithms for implementing Gaussian process models

In practice, we normally do not have enough information to fix the hyperparameters in a Gaussian process model. Instead, we need to assign priors to the hyperparameters at a higher level so that the hyperparameters can also be updated according to the observed data. Therefore, we need to alternate between two steps of Markov chain Monte Carlo (MCMC) updating: 1) updating the hyperparameters given the current latent vector(s) and the observed data; 2) updating the latent vector(s) given the current hyperparameters and the observed data. This section will first describe the respective MCMC algorithms for the two steps of updating and then explain how to combine these two steps. The initializing policy and the stopping rule for MCMC updating will also be discussed in the end of this section.

3.2.1 Univariate slice sampling

Univariate slice sampling (Neal, 2003) is a good choice for sampling one hyperparameter at a time, given the latent vector(s) and the other hyperparameters. Compared to Metropolis-Hastings sampling, univariate slice sampling does not depend crucially on the choice of a scaling parameter and is therefore easier to tune, especially when the spread of the target distribution varies over the MCMC iterations.

Suppose we need to sample from a univariate distribution $f(y)$ where $f(y)$ is known only up to the normalizing constant. The one-step univariate slice sampling for sampling from $f(y)$ is illustrated in Figure 3.1.

Algorithm 1

```
Input: current state $y$
output: new state $y*$

1. Choose a step width $w$ (not dramatically crucial)
2. Determine the slice
       $u \sim Unif[0,1]$
       $\log z \leftarrow \log f(y) + \log u$
   (i.e. $z \sim Unif[0, f(y)]$)
3. Determine the interval to sample from
       $u \sim Unif(0, w)$
       $L \leftarrow y - u;\ R \leftarrow y + w - u$
4. Propose $y^*$ from $Unif(L, R)$
5. If $log(f(y^*)) \geq log(z)$: accept and return $y^*$
6. Else: if $y^* > y$, $R \leftarrow y^*$ else $L \leftarrow y^*$
7. Go back to step 4
```

Figure 3.1: Univariate slice sampling

3.2.2 Metropolis-Hastings sampling with proposal from the prior

The well known multivariate Metropolis-Hastings (MH) algorithm (Hastings, 1970) may be suitable for sampling the latent vectors given the hyperparameters and the observed data. However, it requires a careful choice of the proposal distribution. A possible good choice for the proposal distribution is based on the (joint) prior of the latent vector(s) which is a multivariate Gaussian distribution as given in (2.62). For simplicity, denote the (joint) prior distribution of the latent vector(s) by $\pi(\mathbf{g}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. We can then select $\mathbf{v} \sim \pi(\mathbf{g})$ and propose

$$\mathbf{g}^* = (1 - \epsilon^2)^{1/2}\mathbf{g} + \epsilon\mathbf{v} \tag{3.28}$$

where $0 < \epsilon < 1$ is a scaling constant (Neal, 1998). The transition from $\mathbf{g}$ to the proposed $\mathbf{g}^*$ follows a normal distribution with a mean vector $(1 - \epsilon^2)^{1/2}\mathbf{g}$ and a covariance matrix $\epsilon^2\Sigma$.

Let $T(\mathbf{g}^*; \mathbf{g})$ be the transition probability from $\mathbf{g}$ to $\mathbf{g}^*$. Then $T(\mathbf{g}^*; \mathbf{g})$ satisfies the detailed balance with respect to $\pi(\mathbf{g}) \sim \mathcal{N}(\mathbf{0}, \Sigma)$ as shown by the following.

$$
\begin{aligned}
& \pi(\mathbf{g})\, T(\mathbf{g}^*; \mathbf{g}) \\
= \; & \frac{1}{(2\pi)^n |\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\mathbf{g}^T \Sigma^{-1} \mathbf{g}\right\} \times \\
& \frac{1}{(2\pi\epsilon)^n |\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{g}^* - (1-\epsilon^2)^{1/2}\mathbf{g})^T (\epsilon^2 \Sigma)^{-1} (\mathbf{g}^* - (1-\epsilon^2)^{1/2}\mathbf{g})\right\} \\
= \; & \frac{1}{(2\pi)^n |\Sigma|^{\frac{1}{2}} (2\pi\epsilon)^n |\Sigma|^{\frac{1}{2}}} \times \\
& \exp\left\{-\frac{1}{2}\mathbf{g}^T \Sigma^{-1} \mathbf{g} - \frac{1}{2}\epsilon^{-2} \mathbf{g}^{*T} \Sigma^{-1} \mathbf{g}^* + \epsilon^{-2}(1-\epsilon^2)^{1/2} \mathbf{g}^T \Sigma^{-1} \mathbf{g}^* - \frac{1}{2}(\epsilon^{-2} - 1)\mathbf{g}^T \Sigma^{-1} \mathbf{g}\right\} \\
= \; & \frac{1}{(2\pi)^n |\Sigma|^{\frac{1}{2}} (2\pi\epsilon)^n |\Sigma|^{\frac{1}{2}}} \times \\
& \exp\left\{-\frac{1}{2}\epsilon^{-2} \mathbf{g}^T \Sigma^{-1} \mathbf{g} - \frac{1}{2}\epsilon^{-2} \mathbf{g}^{*T} \Sigma^{-1} \mathbf{g}^* + \epsilon^{-2}(1-\epsilon^2)^{1/2} \mathbf{g}^T \Sigma^{-1} \mathbf{g}^*\right\} \\
= \; & \pi(\mathbf{g}^*)\, T(\mathbf{g}; \mathbf{g}^*), \qquad (3.29)
\end{aligned}
$$

where the last equation is obvious since the preceding expression is symmetric in $\mathbf{g}$ and $\mathbf{g}^*$. The detailed balance of $T(\mathbf{g}^*; \mathbf{g})$ with respect to $\pi(\mathbf{g})$ results in an acceptance probability α that only depends on the likelihood function L as shown by

$$\alpha = \min\left(\frac{\pi(\mathbf{g}^*) L(\mathbf{g}^*) T(\mathbf{g}; \mathbf{g}^*)}{\pi(\mathbf{g}) L(\mathbf{g}) T(\mathbf{g}^*; \mathbf{g})}, 1\right) = \min\left(\frac{L(\mathbf{g}^*)}{L(\mathbf{g})}, 1\right) \qquad (3.30)$$

However, the scaling parameter ϵ needs to be carefully selected for obtaining good performance. An alternative method that automatically selects the scaling parameter is given in the next subsection.

3.2.3 Elliptical slice sampling

The elliptical slice sampling (ESS) scheme by Murray, Adams and MacKay (2010) automatizes the selection of ϵ in the previous Metropolis-Hastings sampling, by applying the slice sampling on an ellipse so that ϵ can be chosen by uniformly sampling θ from the ellipse with $\epsilon = \cos\theta$. The algorithm of the ESS is described in Figure 3.2.

The elliptical slice sampling algorithm illustrated in Figure 3.2 is only suitable for Gaussian priors with a zero mean vector. When the latent vector g has a non-zero prior mean $\mathbf{a}$, this algorithm needs to be slightly modified as shown in Figure 3.3.

Algorithm 2

```
Input: current state $\mathbf{g}$
output: new state $\mathbf{g}^*$.

1. Choose ellipse: $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$
2. Determine Log-likelihood threshold:
       $u \sim Unif[0,1]$
       $\log z \leftarrow \log L(\mathbf{g}) + \log u$
   (i.e. $z \sim Unif[0, L(\mathbf{g})]$)
3. Draw an initial proposal, also defining a bracket:
       $\theta \sim Unif[0, 2\pi]$
       $[\theta_{min}, \theta_{max}] \leftarrow [\theta - 2\pi, \theta]$
4. Let $\mathbf{g}^* \leftarrow \mathbf{g}\cos\theta + \mathbf{v}\sin\theta$
5. If $\log L(\mathbf{g}^*) \geq \log z$: Accept and return $\mathbf{g}^*$
6. else: Shrink the bracket and try a new point: if $\theta < 0$ then $\theta_{min} \leftarrow \theta$ else $\theta_{max} \leftarrow \theta$
7. Go to step 4
```

Figure 3.2: Elliptical slice sampling.

Algorithm 2.1

```
Input: current state $\mathbf{g}$, with prior $\mathcal{N}(\mathbf{a}, \mathbf{\Sigma})$.
output: update state $\mathbf{g}^*$.
1. Let $\tilde{\mathbf{g}} = \mathbf{g} - \mathbf{a}$
2. Update $\tilde{\mathbf{g}} \rightarrow \tilde{\mathbf{g}}^*$ using Algorithm 2.
3. Return $\mathbf{g}^* = \tilde{\mathbf{g}}^* + \mathbf{a}$
```

Figure 3.3: Elliptical slice sampling with non-zero prior mean.

3.2.4 Combining univariate slice sampling and elliptical slice sampling

As discussed earlier, the hyperparameters and the latent vector(s) need to be updated alternately. More specifically, each hyperparameter will be updated using **Algorithm 1**; after updating each hyperparameter, the latent vector(s) will be updated for a fixed number (e.g. 5) of times using **Algorithm 2** or **2.1**; then the next hyperparameter will be updated using **Algorithm 1** until all the hyperparameters have been updated once. The latest values of the hyperparameters and the latent vector(s) form one MCMC iteration. Figure 3.4 illustrates the combined algorithm of univariate slice sampling and elliptical slice sampling for alternately updating the hyperparameters and the latent vector(s).

```
Algorithm 3

Input: current states: latent vector $\mathbf{g}$ and hyperparameter $\zeta_1,\ \zeta_2,\ \dots,\ \zeta_k$
Output: new states: latent vector $\mathbf{g}*$ and hyperparameter $\zeta_1^*,\ \zeta_2^*,\ \dots,\ \zeta_k^*$

1. Update hyperparameter $\zeta_1 \to \zeta_1^*$ using Algorithm 1
2. Update latent vector $\mathbf{g} \to \mathbf{g}^{*\mathbf{1}}$ using Algorithm 2 or 2.1 for $s$ iterations (e.g. $s = 5$).
3. Repeat steps 1-2 alternately for $\zeta_2,\ \dots,\ \zeta_k$:
       $$\zeta_2 \to \zeta_2^*,\ \mathbf{g}^{*\mathbf{1}} \to \mathbf{g}^{*\mathbf{2}},\ \dots,\ \zeta_k \to \zeta_k,\ \mathbf{g}^{*(\mathbf{k-1})} \to \mathbf{g}^{*\mathbf{k}}$$
4. Return $\zeta 1^*,\ \zeta_2^*,\ \dots,\ \zeta_k^*$ and $\mathbf{g}^* = \mathbf{g}^{*\mathbf{k}}$
```

Figure 3.4: Combining univariate slice sampling and elliptical slice sampling.

3.2.5 Initializing and stopping MCMC updating

There are multiple ways of initializing a MCMC updating. For the experiments and the example considered in this thesis, all the hyperparameters that need to be updated will be initialized with the mean values of their corresponding prior distributions; the latent vector(s) will be initialized with the vector of zeros. Given these initial values, the latent vector(s) will first be updated for 100 times using **Algorithm 2** or **2.1** so that the latent vector(s) can be well catered to the observed data. The latest latent vector(s) will be considered as the new initial values. Then the hyperparameters and the latent vector(s) will be updated using **Algorithm 3**. The updating will be stopped when enough iterations have been obtained. With all the iterations obtained, an initial portion (e.g. 1/5) should be discarded since at these initial iterations the MCMC sampling may not have converged well.

The desired number of iterations is determined by the desired effective sample sizes of the MCMC sample of the population mean ϕ and of the MCMC samples of various other functions of the hyperparameters and the latent vector(s). The effective sample size of a MCMC sample equals the number of MCMC iterations (after discarding the non-convergent initial portion) divided by its autocorrelation time. To estimate the autocorrelation time, the autoregression method by Thompson (2010) will be adopted for its demonstrated advantage over other available methods. For the experiments and the example considered in this thesis, 100 will be chosen as the desired effective MCMC sample size for ϕ and 20 is chosen for all the other functions of state considered. If time allows, one can, of course, obtain larger effective sizes by running MCMC updating for longer time to achieve more accurate estimation.

Chapter 4

Experimental studies

This chapter investigates, through computer simulated experiments, the behaviors of the Gaussian process estimators as described in Section 2.2 with comparison to the Horvitz-Thompson estimators and the naive estimator. In Section 4.1, only the non-model based Horvitz-Thompson estimators and naive estimator are studied in a very simple scenario. High-dimensional experiments with both the Gaussian process estimators and the non-model based estimators are carried out under various scenarios in Section 4.2. An example from the literature is also studied in Section 4.3.

Recall that with n iid observations $(\mathbf{x}_i, y_i, r_i)$, $i = 1, \ldots, n$, the non-model based estimators are defined as

$$\begin{aligned}
\widehat{\phi}_{naive} &= \sum_{i=1}^{n} y_i r_i \Big/ \sum_{i=1}^{n} r_i. \\
\widehat{\phi}_{HT_1} &= \frac{1}{n} \sum_{i=1}^{n} \frac{y_i r_i}{\nu_i} \\
\widehat{\phi}_{HT_2} &= \sum_{i=1}^{n} \frac{y_i r_i}{\nu_i} \Big/ \sum_{i=1}^{n} \frac{r_i}{\nu_i} \\
\widehat{\phi}_{HT_3} &= \psi \sum_{i=1}^{n} \frac{y_i r_i}{\nu_i} \Big/ \sum_{i=1}^{n} r_i.
\end{aligned}$$

where ν_i is the selection probability at $\mathbf{x}_i$ and $\psi = \int \nu(\mathbf{x}) dF_{\mathbf{X}}(\mathbf{x})$ is the marginal selection probability. As mentioned earlier that these estimators (except $\widehat{\phi}_{HT_1}$) are not defined when $\sum_{i=1}^{n} r_i = 0$. Although the case where all the observations are missing is of no practical interest, to avoid numerical errors in computer simulations, let

$$\widehat{\phi}_{naive} = \widehat{\phi}_{HT_2} = \widehat{\phi}_{HT_3} = 0, \text{ when } \sum_{i=1}^{n} r_i = 0$$

through all the experiments and the example considered in this thesis.

4.1 Non-model based estimators in a special scenario

This section studies the Horvitz-Thompson estimators as well as the naive estimator in a simple special scenario described next. Consider a partition of the covariate vector space $\mathcal{X}$ into two subspaces, $\mathcal{X} = \mathcal{X}_0 \dot{\cup} \mathcal{X}_1$, as shown in Figure 4.1. Let $\mu = 0$ and $\nu = \nu_0$, when $\mathbf{x} \in \mathcal{X}_0$ and $\mu = 1$ and $\nu = \nu_1$, when $\mathbf{x} \in \mathcal{X}_1$. Then μ and ν are perfectly correlated (unless $\nu_0 = \nu_1$). Let $p_0 = \Pr(\mathbf{x} \in \mathcal{X}_0)$ and $p_1 = \Pr(\mathbf{x} \in \mathcal{X}_1)$. Then

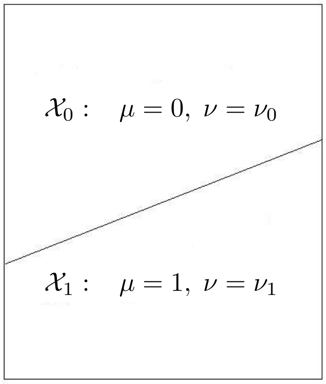

$\mathcal{X}_0 :\quad \mu = 0,\ \nu = \nu_0$
$\mathcal{X}_1 :\quad \mu = 1,\ \nu = \nu_1$

Figure 4.1: The special scenario: $\mathcal{X} = \mathcal{X}_0 \dot{\cup} \mathcal{X}_1$.

$$\begin{aligned} \phi &= \int \mu(\mathbf{x}) dF_{\mathbf{X}} = p_1 \\ \psi &= \int \nu(\mathbf{x}) dF_{\mathbf{X}} = p_0\nu_0 + p_1\nu_1 \end{aligned} \tag{4.1}$$

Note that since $\mu = 0$ for all $\mathbf{x} \in \mathcal{X}_0$ and $\mu = 1$ for all $\mathbf{x} \in \mathcal{X}_1$, y is fully determined by $\mathbf{x}$ as

$$\Pr(y = 0 | \mathbf{x} \in \mathcal{X}_0) = 1 \text{ and } \Pr(y = 1 | \mathbf{x} \in \mathcal{X}_1) = 1 \tag{4.2}$$

Because of this, $\widehat{\phi}_{HT_1}$ does not depend on ν_0 as shown by

$$\widehat{\phi}_{HT_1} = \frac{1}{n}\sum_{i=1}^{n}\frac{y_i r_i}{\nu_i} = \frac{1}{n}\sum_{i:\mathbf{x}_i\in\mathcal{X}_0}\frac{0\times r_i}{\nu_0} + \frac{1}{n}\sum_{i:\mathbf{x}_i\in\mathcal{X}_1}\frac{y_i r_i}{\nu_1} = \frac{1}{n}\sum_{i:\mathbf{x}_i\in\mathcal{X}_1}\frac{y_i r_i}{\nu_1} \tag{4.3}$$

However, if we flip y around, i.e. let $y^* = 1 - y$, then it is ν_1 rather than ν_0 that $\widehat{\phi}_{HT_1}$ does not depend on. This is not surprising, due to the non-equivariance of $\widehat{\phi}_{HT_1}$ under certain affine transformations.

To compare the Horvitz-Thompson estimators under this special scenario, we first look at their (asymptotic) mean square errors (MSE). Numerical studies through computer simulations follow next.

4.1.1 MSE of the Horvitz-Thompson estimators

First consider two trivial cases, where $p_1 = 0$ or 1. When $p_1 = 0$, all estimators equal zero and estimate ϕ perfectly.

When $p_1 = 1$, we have

$$\widehat{\phi}_{naive} = \widehat{\phi}_{HT_2} = \widehat{\phi}_{HT_3} = \begin{cases} 0, & \text{if } \sum_{i=1}^n r_i = 0 \\ 1, & \text{if } \sum_{i=1}^n r_i \neq 0 \end{cases} \tag{4.4}$$

That is, $\widehat{\phi}_{naive}$, $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ estimate $\phi = p_1 = 1$ almost perfectly except when all the observations are missing, which happens with a probability $(1-\nu_1)^n \to 0$ as $n \to \infty$. Therefore, the above three estimators have their mean square error equal to the probability that $\sum_{i=1}^n r_i = 0$, that is,

$$MSE(\widehat{\phi}_{naive}) = MSE(\widehat{\phi}_{HT_2}) = MSE(\widehat{\phi}_{HT_3}) = (1-\nu_1)^n \tag{4.5}$$

When $p_1 = 1$, we also have

$$\widehat{\phi}_{HT_1} = \frac{\sum_{i=1}^n r_i}{n\nu_1}, \tag{4.6}$$

with an MSE equal to

$$MSE(\widehat{\phi}_{HT_1}) = \text{Var}(\widehat{\phi}_{HT_1}) = \text{Var}\left(\frac{\sum_{i=1}^n r_i}{n\nu_1}\right) = \frac{n\nu_1(1-\nu_1)}{n^2\nu_1^2} = \frac{1-\nu_1}{n\nu_1} \tag{4.7}$$

where the first equation is due to $\widehat{\phi}_{HT_1}$ being unbiased. $\widehat{\phi}_{HT_1}$ in (4.6) can be viewed as the ratio of the number of observed y's to the expected number of observed y's.

For general p_1 values, following directly from (2.28) and (2.35), we have

$$\begin{aligned} MSE(\widehat{\phi}_{HT_1}) &= \frac{1}{n\psi}\frac{\nu_0 p_0 p_1}{\nu_1} + \frac{(1-\psi)p_1^2}{n\psi} = \frac{1}{n\psi}\frac{\nu_0 p_0 p_1 + (1-\psi)\nu_1 p_1^2}{\nu_1} \\ &= \frac{1}{n\psi}\frac{\nu_0 p_0 p_1 + \nu_1 p_1^2 - \psi\nu_1 p_1^2}{\nu_1} = \frac{1}{n\psi}\frac{\psi p_1 - \psi\nu_1 p_1^2}{\nu_1} = \frac{p_1(1-\nu_1 p_1)}{n\nu_1} \end{aligned} \tag{4.8}$$

and

$$MSE(\widehat{\phi}_{HT_3}) = \frac{1}{n\psi}\frac{\nu_0 p_0 p_1}{\nu_1} + o\left(\frac{1}{n^{3/2}}\right) \tag{4.9}$$

Comparing (4.8) and (4.9), we expect that for reasonably large n, $\widehat{\phi}_{HT_3}$ would dominate $\widehat{\phi}_{HT_1}$ in terms of MSE, unless $\psi = 1$ or $p_1 = 0$.

Rewrite (4.9) as

$$MSE(\widehat{\phi}_{HT_3}) \approx \frac{1}{n\psi}\frac{\nu_0 p_0 p_1}{\nu_1} = \frac{\nu_0 p_0 p_1}{n(p_0\nu_0 + p_1\nu_1)\nu_1} = \frac{p_0 p_1}{n(p_0 + p_1\nu_1/\nu_0)\nu_1} \tag{4.10}$$

From (4.10), we expect that when n is reasonably large, the MSE of $\widehat{\phi}_{HT_3}$ would decrease when ν_1 increases and increase when ν_0 increases. As also noted, the MSE of $\widehat{\phi}_{HT_1}$, however, does not change when ν_0 changes.

4.1.2 Simulation studies

For numerical studies with computer simulations, the following values of p_1, ν_0, ν_1 and n are considered

p_1:	0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95
ν_0:	0.1, 0.2, 0.5, 0.6, 0.9, 1
ν_1:	0.1, 0.2, 0.5, 0.6, 0.9, 1
n:	30, 100, 500, 1000

For each combination of ν_0 and ν_1, the root mean squared error (RMSE) of each estimator is plotted versus p_1. The plots for n=30, 100 and 500 are given in Figures 4.2 - 4.4, respectively. For n=1000, the results are similar to those of n=500 and are not present here in order to save space.

As observed from Figures 4.2 - 4.4, the naive estimator has relatively bigger RMSE when $\nu_0 \neq \nu_1$ than when $\nu_0 = \nu_1$ due to selection bias. The selection bias is more an issue than the sampling error for the naive estimator when n is large. For example, when $n = 1000$ (not shown here), the RMSE of the naive estimator is the smallest when $\nu_0 = \nu_1$ for all p_1's. This is also true for $n = 500$, except when $p_1 = 0.05$ or 0.95 where the selection bias is a less important contributor to the MSE than the sampling error. The $\widehat{\phi}_{HT_1}$ estimator has its RMSE decrease when ν_1 increases. As pointed out earlier, the MSE or RMSE of $\widehat{\phi}_{HT_1}$ remains the same as ν_0 changes. For $\widehat{\phi}_{HT_2}$, the RMSE decreases when either ν_0 or ν_1 increases with a more obvious trend when ν_0 changes. The RMSE of $\widehat{\phi}_{HT_3}$, however, increases when ν_0 increases and decreases when ν_1 increases, as expected from (4.10). When $\nu_0 = \nu_1 = 1$, all the methods converge to the naive method and therefore have the same RMSE.

As expected from (4.8) and (4.9), $\widehat{\phi}_{HT_1}$ always has bigger RMSE than $\widehat{\phi}_{HT_3}$, except for a few cases when $n = 30$ or 100 where the asymptotic result in (4.9) may not apply. Although $\widehat{\phi}_{HT_1}$ is always unbiased and the naive estimator is not when $\nu_0 \neq \nu_1$, for certain values of ν_0, ν_1 and p_1, $\widehat{\phi}_{HT_1}$ even has larger RMSE than the naive estimator due to selection bias being dominated by sampling error.

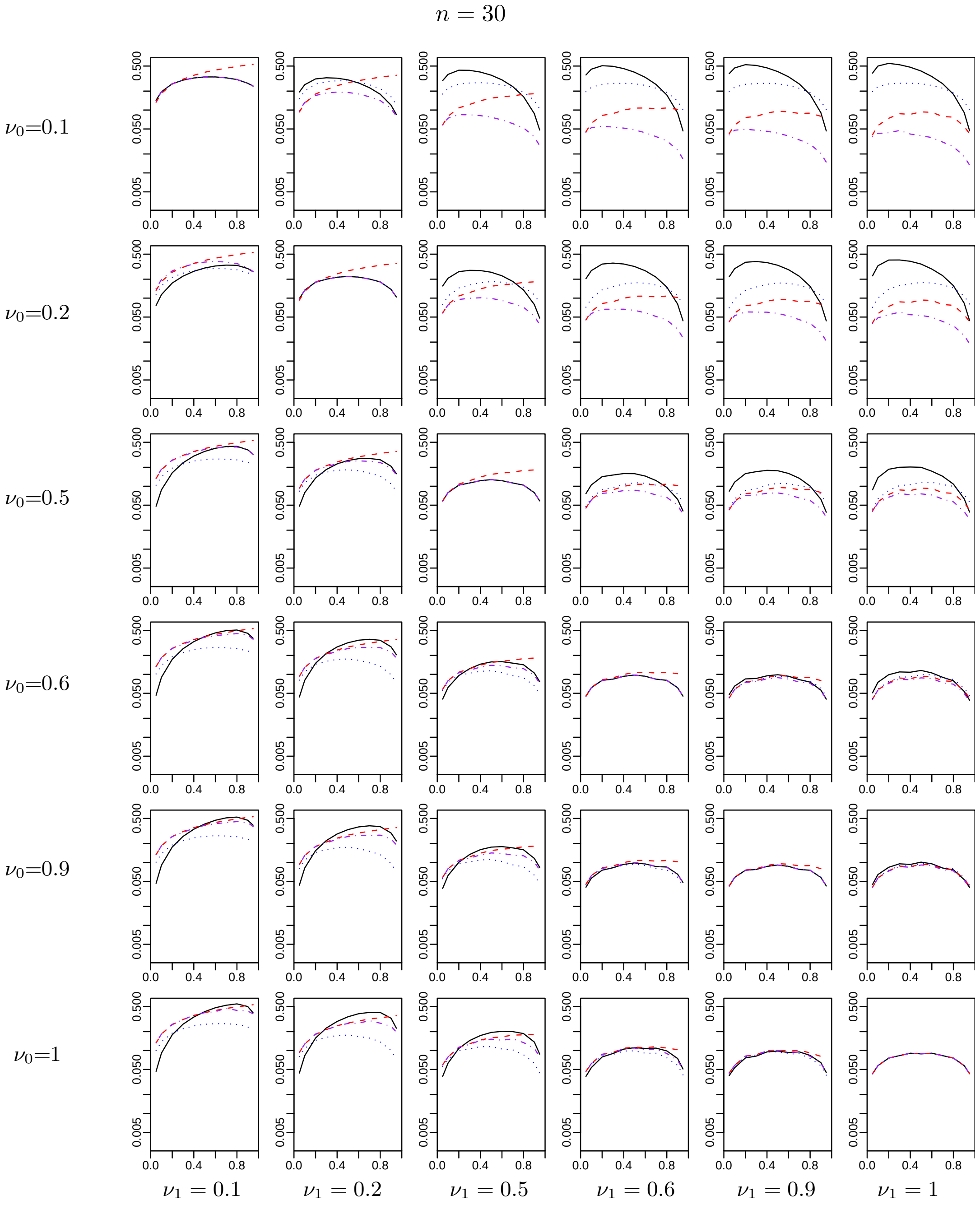


Figure 4.2: Root mean squared error (RMSE) (in log scale) v.s. p_1: $\widehat{\phi}_{naive}$ (solid black), $\widehat{\phi}_{HT_1}$ (dashed red), $\widehat{\phi}_{HT_2}$ (dotted blue), $\widehat{\phi}_{HT_3}$ (dash-dotted purple). $\nu_0 = 0.1$, 0.2, 0.5, 0.8, 0.9, 1 runs from top to bottom; $\nu_1 = 0.1$, 0.2, 0.5, 0.8, 0.9, 1 runs from left to right; $n = 30$.

$\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ both reduce to the naive estimator when $\nu_0 = \nu_1$ and therefore have the same RMSE when $\nu_0 = \nu_1$. Since as noted earlier, when ν_0 increases, the RMSE of $\widehat{\phi}_{HT_2}$ deceases while the RMSE of $\widehat{\phi}_{HT_3}$ increases, it is not surprising that $\widehat{\phi}_{HT_2}$ outperforms $\widehat{\phi}_{HT_3}$ for $\nu_0 > \nu_1$ and is outperformed by $\widehat{\phi}_{HT_2}$ for $\nu_0 < \nu_1$.

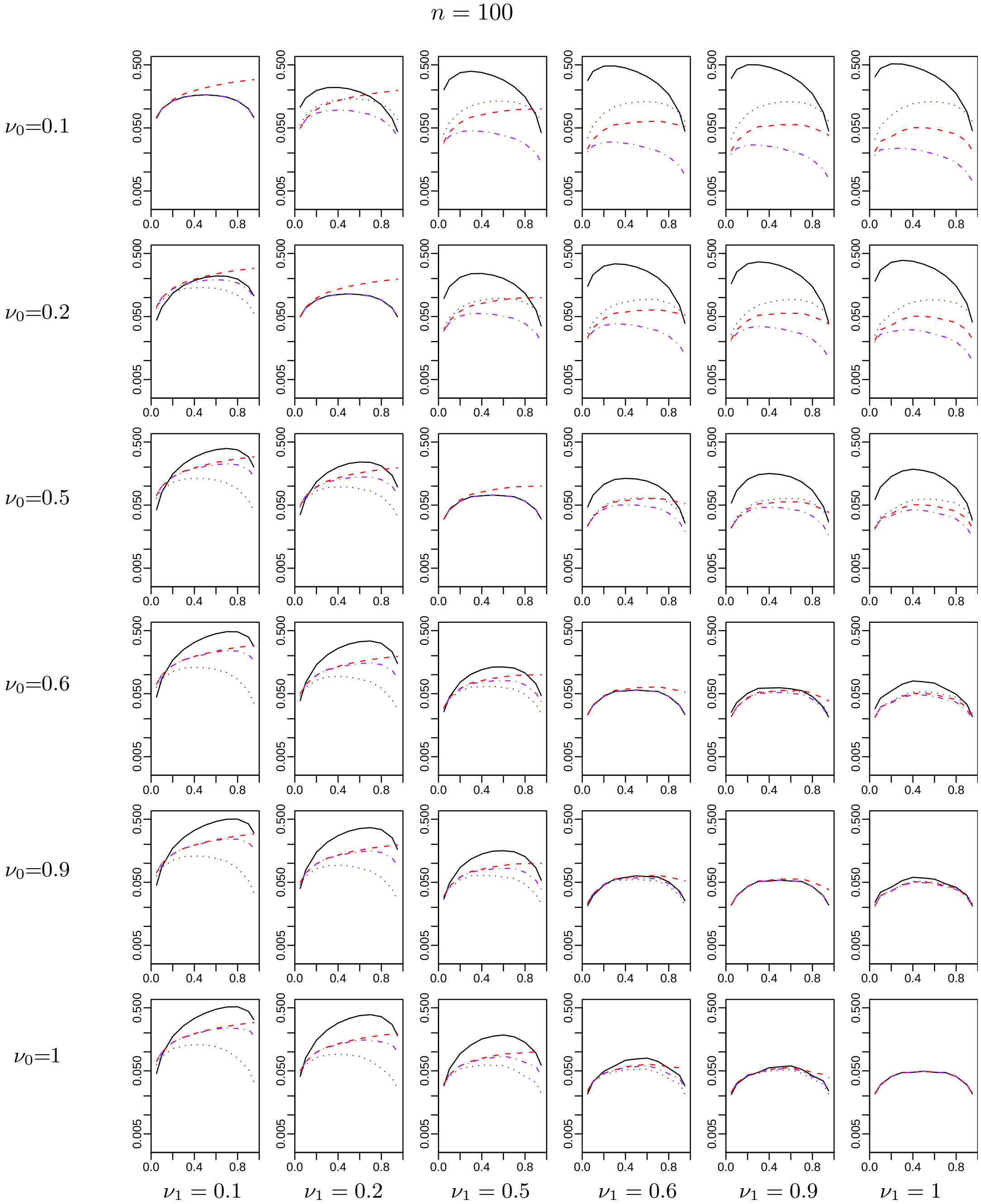


Figure 4.3: Root mean squared error (RMSE) (in log scale) v.s. p_1: $\widehat{\phi}_{naive}$ (solid black), $\widehat{\phi}_{HT_1}$ (dashed red), $\widehat{\phi}_{HT_2}$ (dotted blue), $\widehat{\phi}_{HT_3}$ (dash-dotted purple). $\nu_0 = 0.1,\ 0.2,\ 0.5,\ 0.8,\ 0.9,\ 1$ runs from top to bottom; $\nu_1 = 0.1,\ 0.2,\ 0.5,\ 0.8,\ 0.9,\ 1$ runs from left to right; $n = 100$.

Note that under this special scenario, the correlation between $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ with respect to $\mathbf{x}$ equals $p_0p_1\nu_1 - p_0p_1\nu_0$. Therefore, when $\nu_0 < \nu_1$ where $\widehat{\phi}_{HT_3}$ outperforms $\widehat{\phi}_{HT_2}$, $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ are positively correlated; when $\nu_0 > \nu_1$ where $\widehat{\phi}_{HT_3}$ is outperformed by $\widehat{\phi}_{HT_3}$, $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ are negatively correlated. Since $\widehat{\phi}_{HT_3}$ is not equivariant if y is flipped to $y^* = 1 - y$, it will perform differently when $\mu(\mathbf{x})$ and

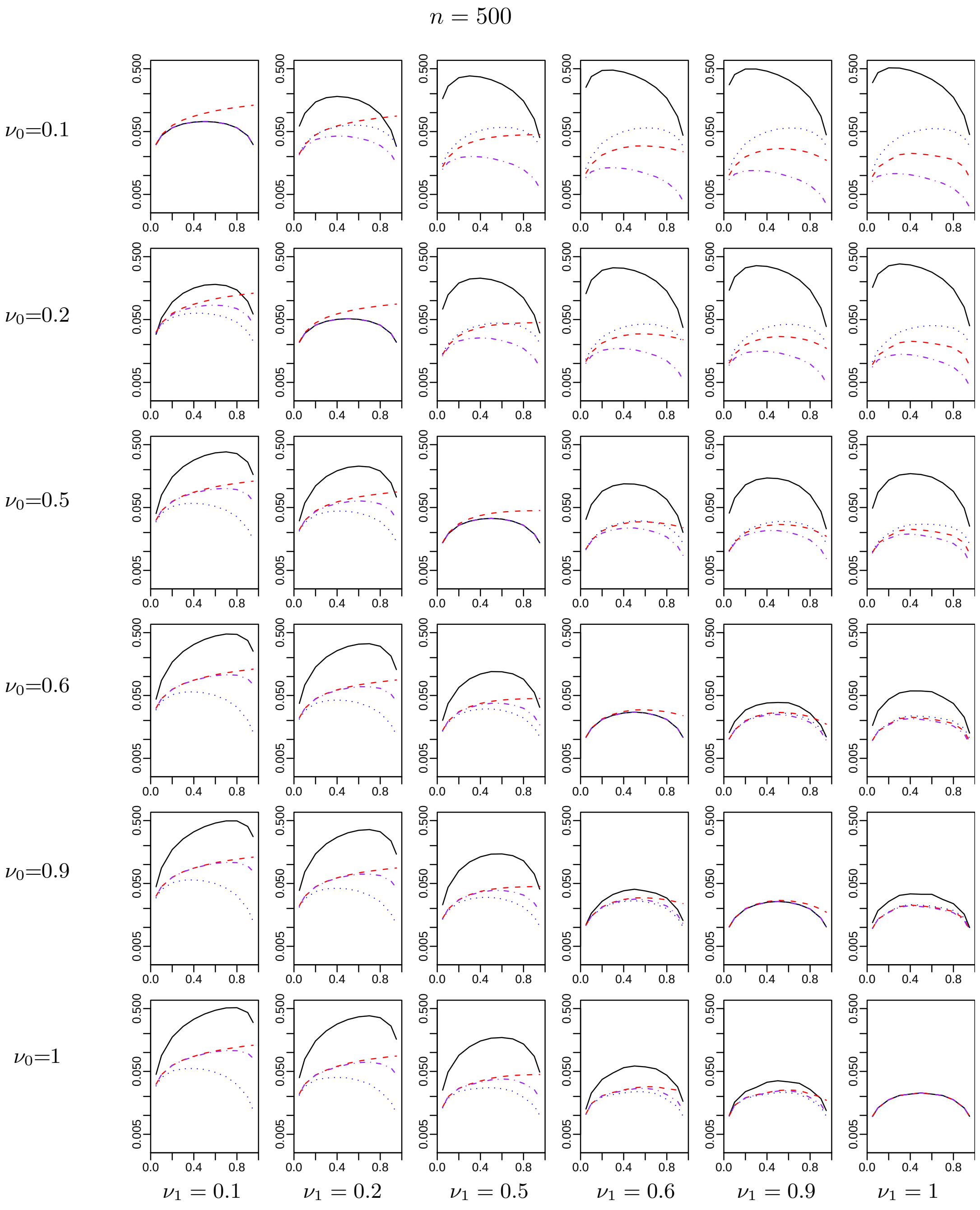


Figure 4.4: Root mean squared error (RMSE) (in log scale) v.s. p_1: $\widehat{\phi}_{naive}$ (solid black), $\widehat{\phi}_{HT_1}$ (dashed red), $\widehat{\phi}_{HT_2}$ (dotted blue), $\widehat{\phi}_{HT_3}$ (dash-dotted purple). $\nu_0 = 0.1$, 0.2, 0.5, 0.8, 0.9, 1 runs from top to bottom; $\nu_1 = 0.1$, 0.2, 0.5, 0.8, 0.9, 1 runs from left to right; $n = 500$.

$\nu(\mathbf{x})$ are correlated in different directions. $\widehat{\phi}_{HT_2}$, however, will not be affected by the sign of the correlation between $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ due to its equivariance under all affine transformation as discussed earlier. Whether $\widehat{\phi}_{HT_3}$ will always be better than $\widehat{\phi}_{HT_2}$ when $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ are positively correlated and worse when $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ are negatively correlated remains unclear and will be discussed more in later sections.

As when $\sum_{i=1}^{n} r_i = 0$, the values of $\widehat{\phi}_{naive}$, $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ are arbitrary, we may also consider, for example, $\widehat{\phi}_{naive} = \widehat{\phi}_{HT_2} = \widehat{\phi}_{HT_3} = 1/2$ at $\sum_{i=1}^{n} r_i = 0$. For $\phi \in (0,1)$, $1/2$ seems a more reasonable guess than zero when nothing is observed. By such definition, the MSE of $\widehat{\phi}_{naive}$, $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ when $p_1 = 0$ or $p_1 = 1$ becomes

$$MSE(\widehat{\phi}_{naive}) = MSE(\widehat{\phi}_{HT_2}) = MSE(\widehat{\phi}_{HT_3}) = \begin{cases} \frac{1}{4}(1-\nu_0)^n, & \text{if } p_1 = 0 \\ \frac{1}{4}(1-\nu_1)^n, & \text{if } p_1 = 1 \end{cases} \tag{4.11}$$

which will lead to symmetry with respect to p_1 when $\nu_0 = \nu_1$. When n is large, $\Pr(\sum_{i=1}^{n} r_i = 0)$ is negligible and therefore, regardless of the value of $\widehat{\phi}_{naive}$, $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ when $\sum_{i=1}^{n} r_i = 0$, the RMSE of these three estimators will exhibit symmetry about p_1 when $\nu_0 = \nu_1$, as shown by those plots on the diagonal positions in each of Figures 4.2 - 4.4. For $\widehat{\phi}_{HT_1}$ which favors zero when none or few of $y's$ are observed will not have such a property, even when n is large (unless $\nu_0 = \nu_1$=1).

In summary, when potential selection bias exists, an estimator that ignores the selection bias is not desired. However, a good estimator should not only be unbiased, but also be able to control the overall MSE under various situations. According to the results of this simple experiment, $\widehat{\phi}_{HT_3}$ is more desirable than $\widehat{\phi}_{HT_1}$ whenever ψ is known or easy to obtain, however, neither $\widehat{\phi}_{HT_2}$ nor $\widehat{\phi}_{HT_3}$ dominates the other in all situations considered.

4.2 High-dimensional experiments

This section studies both the Gaussian process estimators and the non-model based frequentist estimators through computer simulated experiments under various scenarios with different dimensionalities, d. In all the experiments considered, the response variable y is binary with $\mu(\mathbf{x}) = \Pr(y = 1|\mathbf{x}) = 1 - \Pr(y = 0|\mathbf{x})$. An example due to Kang and Schafer with y being real-valued is studied in the next section.

4.2.1 Methods of estimation

In total, there are nine estimators to be studied in this section. They are grouped into two categories: non-model based estimators and Gaussian process model based estimators.

Non-model based estimators

The non-model based estimators include the naive estimator defined in (2.1) and the three Horvitz-Thompson estimators defined in (2.2), (2.4) and (2.14). The reason to have the naive estimator is to help identify when selection bias is indeed an issue and how severe an issue it is. These four estimators assume no models for the response variable y and completely ignore the covariate vector $\mathbf{x}$.

Gaussian process model based estimators

Gaussian process model based estimators (or Gaussian process estimators) are built on latent functions that have priors based on Gaussian process models. Latent functions are connected to the functions of interest by link functions. For its computational convenience, the probit link function is used throughout this section. That is,

$$\begin{aligned} \mu(\mathbf{x}) &= \tilde{\mu}(g_\mu(\mathbf{x})) = \Phi(g_\mu(\mathbf{x})) \\ \nu(\mathbf{x}) &= \tilde{\nu}(g_\nu(\mathbf{x})) = (1-\zeta)\Phi(g_\nu(\mathbf{x})) + \zeta,\ 0 < \zeta < 1 \end{aligned} \tag{4.12}$$

where $g_\mu(\mathbf{x})$ and $g_\nu(\mathbf{x})$ are latent functions, Φ is the probit link function, i.e. the cumulative distribution function of the standard normal, and ζ is a constant that keeps $\nu(\mathbf{x})$ away from zero.

A Gaussian process model is characterized by its covariance function. When the selection probability is not used as a covariate, the following covariance function as in (2.50) is used.

$$C(\mathbf{x}_i, \mathbf{x}_j; \sigma_0^2,\ \lambda^2,\ \eta^2,\ \ell,\ r) = \sigma_0^2 + \frac{1}{d}\sum_{k=1}^{d} \lambda_k^2 x_{ik} x_{jk} + \eta^2 \exp\left\{ -\frac{1}{d}\sum_{k=1}^{d} \left(\frac{|x_{ik} - x_{jk}|}{\ell_k} \right)^r \right\} \tag{4.13}$$

When the selection probability is used as a covariate, the following covariance function as in (2.66) is used instead,

$$\begin{aligned} C(\mathbf{x}_i^*, \mathbf{x}_j^*; \sigma_0^2,\ \lambda^2,\ \eta^2,\ \ell,\ r) &= \sigma_0^2 + \frac{1}{d}\sum_{k=1}^{d} \lambda_k^2 x_{ik} x_{jk} + \lambda_{d+1}^2 x_{i,d+1} x_{j,d+1} \\ &+ \eta^2 \exp\left\{ -\frac{1}{d}\sum_{k=1}^{d} \left(\frac{|x_{ik} - x_{jk}|}{\ell_k} \right)^r - \left(\frac{|x_{i,d+1} - x_{j,d+1}|}{\ell_{d+1}} \right)^r \right\} \end{aligned} \tag{4.14}$$

As before, given the hyperparameters σ_0^2, λ^2, η^2, ℓ, r, a corresponding Gaussian process model is denoted by

$$\mathcal{GP}(\sigma_0^2,\ \lambda^2,\ \eta^2,\ \ell,\ r) \tag{4.15}$$

Given the observed data, a posterior sample of the latent function $g_\mu(\mathbf{x})$ based on its Gaussian process prior can be obtained by MCMC algorithms described in Section 3.2. From the posterior sample of $g_\mu(\mathbf{x})$, the population mean $\phi = \mathrm{E}[\mu(\mathbf{x})]$ can be estimated according to formula (3.21), (3.22) or (3.23).

As discussed in earlier chapters, there are three ways of constructing Gaussian process estimators for estimating ϕ: ignoring the selection probability ν, using a joint dependent prior of g_μ and g_ν, and using the selection probability ν as a covariate.

1. Ignoring the selection probability (GP_I)

First, consider ignoring the selection probability $\nu(\mathbf{x})$ and modeling $\mu(\mathbf{x})$ only. As discussed earlier, a model based method without incorporating the selection probability may do a good job if the relationship between y and $\mathbf{x}$ is modeled nearly correct.

Let

$$g_\mu = g_1 \tag{4.16}$$

where g_1 has a Gaussian process prior as follows

$$g_1 \sim \mathcal{GP}_1 = \mathcal{GP}(\sigma_{0,1}^2,\ \lambda_1^2,\ \eta_1^2,\ \ell_1,\ r_1). \tag{4.17}$$

This approach is denoted by GP_I and the estimator for ϕ obtained by this approach is denoted by $\widehat{\phi}_{GP_I}$.

2. Using a joint dependent prior (GP_T and GP_E)

Second, consider assigning a joint dependent prior for $g_\mu(\mathbf{x})$ and $g_\nu(\mathbf{x})$, so that $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ are jointly modeled. By incorporating the selection probability ν, we expect that the efficiency of estimating ϕ should be improved, if μ and ν are indeed related.

To assign a joint dependent prior to $g_\mu(\mathbf{x})$ and $g_\nu(\mathbf{x})$, the strategy described by (2.59)-(2.60) is adopted. That is,

$$g_\mu = g_1 + g_0 \quad \text{and} \quad g_\nu = g_2 + g_0 \tag{4.18}$$

where g_1, g_2 and g_0 have the following Gaussian process priors

$$\begin{aligned} g_1 &\sim \mathcal{GP}_1 = \mathcal{GP}(\sigma_{0,1}^2,\, \lambda_1^2,\, \eta_1^2,\, \ell_1,\, r_1) \perp\!\!\!\perp \\ g_2 &\sim \mathcal{GP}_2 = \mathcal{GP}(\sigma_{0,2}^2,\, \lambda_2^2,\, \eta_2^2,\, \ell_2,\, r_2) \perp\!\!\!\perp \\ g_0 &\sim \mathcal{GP}_0 = \mathcal{GP}(\sigma_{0,0}^2,\, \lambda_0^2,\, \eta_0^2,\, \ell_0,\, r_0). \end{aligned} \tag{4.19}$$

where "$\perp\!\!\!\perp$" denotes *independent* given the hyperparameters.

To apply this strategy, two situations need to be considered.

- When the selection probabilities $\nu_1 = \nu(\mathbf{x}_1), \nu_2 = \nu(\mathbf{x}_2), \ldots, \nu_n = \nu(\mathbf{x}_n)$ are known, the prior for g_μ becomes the prior probability measure conditional on the latent vector $\mathbf{g}_\nu^{(n)} = (g_{\nu,1}, g_{\nu,2}, \ldots, g_{\nu,n})^T$, where $g_{\nu,1}, g_{\nu,2}, \ldots, g_{\nu,n}$ are the latent variables corresponding to $\nu_1, \nu_2, \ldots, \nu_n$.

  The GP method applied under this situation is denoted by GP_T and the corresponding estimator for ϕ is denoted by $\widehat{\phi}_{GP_T}$.

- When the selection probabilities are unknown, we assign the same joint dependent prior to g_μ and g_ν, but estimate both μ and ν.

  The GP method applied under this situation is denoted by GP_E and the corresponding estimator for ϕ is denoted by $\widehat{\phi}_{GP_E}$.

If the Gaussian process model is appropriate and the selection probabilities are known correctly, we expect $\widehat{\phi}_{GP_T}$ to perform better than $\widehat{\phi}_{GP_E}$. Since $\widehat{\phi}_{GP_E}$ does not require knowledge of the selection probability, it has wider applications, and would be more robust against incorrect information for the selection probability, compared to $\widehat{\phi}_{GP_T}$.

3. Using the selection probability as a covariate (GP_R and GP_S)

Third, consider using the selection probability $\nu(\mathbf{x})$ as an additional covariate x_{d+1}. By doing so, we utilise $\nu(\mathbf{x})$ while only needing to model $\mu(\mathbf{x})$. Since all the $\mathbf{x}_i$, $i = 1, \ldots, d$, simulated by our experiments will range from $-\infty$ to ∞, we let $x_{d+1} = \text{logit}(\nu(\mathbf{x}))$ so that x_{d+1} has about the same range as the other x_j's, where $\text{logit}(a) = \log\left(\frac{a}{1-a}\right)$. Then

$$\mu(\mathbf{x}) = \tilde{\mu}(g_\mu(\mathbf{x})) = \tilde{\mu}\left(g_\mu^*\left(\mathbf{x}, \text{logit}\left(\nu(\mathbf{x})\right)\right)\right) \tag{4.20}$$

Note that g_μ is a function with d arguments, $x_1, \ldots, x_d$, while g_μ^* has $d+1$ arguments, $x_1, \ldots, x_d$, x_{d+1}.

Let

$$g_\mu^* = g_1^* \tag{4.21}$$

where g_1^* has a Gaussian process prior as follows

$$g_1^* \quad \sim \quad \mathcal{GP}_1 = \mathcal{GP}(\sigma_{0,1}^2,\ \lambda_1^2,\ \eta_1^2,\ \ell_1,\ r_1). \tag{4.22}$$

The difference between g_1^* and g_1 is that for g_1^*, the hyperparameters λ_1 and ℓ_1 are $d+1$ dimensional instead of d dimensional.

As discussed in Subsection 2.2.4, the strategy of using the selection probability as a covariate requires knowledge of $\nu(\mathbf{x})$.

- When $\nu(\mathbf{x})$ is only known at the observed $\mathbf{x_1}, \ldots, \mathbf{x_n}$, formulae (3.22) should be used for estimating ϕ.

  In this case, the GP method is denoted by GP_R and the corresponding estimator is denoted by $\widehat{\phi}_{GP_R}$.

- When $\nu(\mathbf{x})$ is known for all $\mathbf{x}$, formula (3.21) can be used instead.

  In this case, the GP method is denoted by GP_S and the corresponding estimator is denoted by $\widehat{\phi}_{GP_S}$.

The above five Gaussian process model based estimators are summarized as follow.

$$\begin{aligned}
\widehat{\phi}_{GP_I}: &\quad \text{ignoring } \nu(\mathbf{x})\\
\widehat{\phi}_{GP_T}: &\quad \text{with a joint dependent prior and true } \nu_1, \nu_2, \ldots, \nu_n\\
\widehat{\phi}_{GP_E}: &\quad \text{with a joint dependent prior and estimated } \nu_1, \nu_2, \ldots, \nu_n\\
\widehat{\phi}_{GP_R}: &\quad \text{with } \nu(\mathbf{x}) \text{ as a covariate known only at } \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\\
\widehat{\phi}_{GP_S}: &\quad \text{with } \nu(\mathbf{x}) \text{ as a covariate known at all } \mathbf{x}.
\end{aligned} \tag{4.23}$$

One difference worth noting between $\widehat{\phi}_{GP_I}$, $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ from $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ is that for $\widehat{\phi}_{GP_I}$, $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$, $\mu(\mathbf{x})$ is based on a single latent function, g_1 or g_1^*, while for $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$, $\mu(\mathbf{x})$ is based on the sum of two independent latent functions, g_1 and g_0. Although this difference may complicate the comparison of these estimators, as long as the hyperparameters are adjustable over a wide range of values, it will have little effect on the issues this thesis addresses.

Choosing hyperparameters and their priors

For the Gaussian process model based estimators, we need to decide how to select the hyperparameters $\sigma_{0,h}^2$, λ_h^2, η_h^2, ℓ_h, r_h, $h = 1, 2, 0$, or their corresponding priors when they need to be adjustable at higher levels. For the estimators $\widehat{\phi}_{GP_I}$, $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$, only the hyperparameters for g_1 or g_1^*, i.e. $\sigma_{0,1}^2$, λ_1^2, η_1^2, ℓ_1, r_1, are involved and the other hyperparameters do not apply.

Constant components ($\sigma^2_{0,h}$)

The hyperparameters for the constant components determine how much the latent functions g_1 (or g_1^*), g_2 and g_0 can shift vertically from zero. These hyperparameters do not need to have a prior at a higher level and are fixed as follow.

$$\sigma^2_{0,1} = 0.5^2,\ \sigma^2_{0,2} = 0.5^2,\ \sigma^2_{0,0} = 0 \tag{4.24}$$

Letting $\sigma^2_{0,0} = 0$ allows the procedure on which $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ are based to be able to model independent $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ when they are indeed independent. Letting $\sigma^2_{0,1} = 0.5^2$ allows a chance of about 5% for the average level of $\mu(\mathbf{x})$ to be as large as $0.84 = \Phi^{-1}(2 \cdot 0.5)$ or as small as $0.16 = \Phi^{-1}(-2 \cdot 0.5)$. Similarly, letting $\sigma^2_{0,2} = 0.5^2$ allows a chance of about 5% for the average level of $\nu(\mathbf{x})$ to be as large as $(1-\zeta) \cdot 0.84 + \zeta$ or as small as $(1-\zeta) \cdot 0.16 + \zeta$. Note that over a restricted range of $\mathbf{x}$, these probabilities would be higher than 5%.

Linear components (λ_h)

The hyperparameters in the linear component determine the slope of the latent function along each dimension. For different covariates x_j's, their corresponding linear component hyperparameters do not need to be the same. Instead, a joint prior can be given to these hyperparameters for different covariates. Using a joint prior allows each linear component hyperparameter to be adjusted individually according to the real situation without ignoring their dependency. A common choice for such joint priors is the multivariate log-normal distributions as follow.

$$\begin{aligned}
\log(\lambda_1) &\sim N_d\left(\begin{pmatrix}\log(0.2)\\ \log(0.2)\\ \vdots \\ \log(0.2)\end{pmatrix}, \begin{pmatrix}0.6 & 0.12 & \cdots & 0.12\\ 0.12 & 0.6 & \cdots & 0.12\\ \vdots & \vdots & \ddots & \vdots\\ 0.12 & 0.12 & \cdots & 0.6\end{pmatrix}\right),\\
\log(\lambda_2) &\sim N_d\left(\begin{pmatrix}\log(0.3)\\ \log(0.3)\\ \vdots \\ \log(0.3)\end{pmatrix}, \begin{pmatrix}0.6 & 0.18 & \cdots & 0.18\\ 0.18 & 0.6 & \cdots & 0.18\\ \vdots & \vdots & \ddots & \vdots\\ 0.18 & 0.18 & \cdots & 0.6\end{pmatrix}\right),\\
\log(\lambda_0) &\sim N_d\left(\begin{pmatrix}\log(0.2)\\ \log(0.2)\\ \vdots \\ \log(0.2)\end{pmatrix}, \begin{pmatrix}0.6 & 0.15 & \cdots & 0.15\\ 0.15 & 0.6 & \cdots & 0.15\\ \vdots & \vdots & \ddots & \vdots\\ 0.15 & 0.15 & \cdots & 0.6\end{pmatrix}\right)
\end{aligned} \tag{4.25}$$

These priors for λ_h's are chosen so that the linear trends of the corresponding latent functions will not be too flat or too steep with high probabilities.

For the case of $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$, λ_1 is $d+1$ dimensional instead of d dimensional.

Overall scaling hyperparameters (η_h)

The overall scaling hyperparameter for the exponential component controls the variance of the corresponding latent function at each $\mathbf{x}$. For the case of $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$, η_1, η_2 and η_0 not only control the variances of $g_\mu = g_1 + g_0$ and $g_\nu = g_2 + g_0$ but also the correlation between them. For better performance, the overall scaling hyperparameters must be adjustable with higher level priors. Log-normal distributions are a common choice for these priors. In the experiments considered, the following log-normal distributions will be used.

$$\log(\eta_1) \sim N\left(\log(0.3), 0.7\right),\ \log(\eta_2) \sim N\left(\log(0.2), 0.7\right),\ \log(\eta_0) \sim N\left(\log(0.3), 0.7\right) \tag{4.26}$$

These priors for η_h's are chosen so that the corresponding μ function and ν function will not be saturated with high probabilities, and the correlation between these two functions will not have a high probability of being too extreme.

Length-scale hyperparameters (ℓ_h)

The length-scale hyperparameters for the exponential component control the correlation between the values of the latent function at different $\mathbf{x}$'s. For a given distance between two $\mathbf{x}$'s, the smaller the length-scale hyperparameters are, the less correlated the values of the latent function at these two $\mathbf{x}$'s are. Therefore, the length-scale hyperparameters determine how wiggly or smooth the corresponding latent function is. Similarly to the linear component hyperparameters, a joint prior can be given to the length-scale hyperparameters for different covariates x_j's, so that each length-scale can be adjusted individually without losing their dependency. The following multivariate log-normal priors are chosen.

$$\log(\ell_1) \sim N_d\left(\begin{pmatrix}\log(2)\\ \log(2)\\ \vdots\\ \log(2)\end{pmatrix}, \begin{pmatrix}0.8 & 0.2 & \cdots & 0.2\\ 0.2 & 0.8 & \cdots & 0.2\\ \vdots & \vdots & \ddots & \vdots\\ 0.2 & 0.2 & \cdots & 0.8\end{pmatrix}\right),$$

$$\log(\ell_2) \sim N_d\left(\begin{pmatrix}\log(2)\\ \log(2)\\ \vdots\\ \log(2)\end{pmatrix}, \begin{pmatrix}0.8 & 0.2 & \cdots & 0.2\\ 0.2 & 0.8 & \cdots & 0.2\\ \vdots & \vdots & \ddots & \vdots\\ 0.2 & 0.2 & \cdots & 0.8\end{pmatrix}\right),$$

$$\log(\ell_0) \sim N_d\left(\begin{pmatrix}\log(1)\\ \log(1)\\ \vdots\\ \log(1)\end{pmatrix}, \begin{pmatrix}0.8 & 0.2 & \cdots & 0.2\\ 0.2 & 0.8 & \cdots & 0.2\\ \vdots & \vdots & \ddots & \vdots\\ 0.2 & 0.2 & \cdots & 0.8\end{pmatrix}\right) \tag{4.27}$$

These priors for ℓ_h's are chosen so that atypical length-scale values will not happen with high probabilities.

For the case of $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$, ℓ_1 is $d+1$ dimensional instead of d dimensional.

Exponents (r_h)

The exponents for the exponential components must satisfy $0 < r_h \leq 2$ for the corresponding covariance matrices to be positive definite. When the exponents equal 2, the corresponding latent functions are analytic. Otherwise, they are non-differential. In practice, r_h's could be made adjustable with higher level priors. However, for the experiments considered in this section, r_h's are all fixed at 2 partly for faster computation.

Jitter

For numerical stability, a jitter equal to 10^{-5} will be added to the diagonal elements of the covariance functions of both g_μ and g_ν. Doing so will help avoid singularity of the covariance matrix after possible round-off errors in numerical computations. Quite small compared to the overall scaling hyperparameters η_h^2, $h = 1, 2, 0$, adding such a jitter will only affect the results to a negligible extent. (Note that using a jitter is not necessary for real-valued variables. For example, when y is a real-valued response, a jitter will be replaced by a noise standard deviation which may be a fixed constant or a parameter to be adapted.)

A limitation

One limitation of the chosen priors for λ_h's and for ℓ_h's needs to be addressed. In the chosen multivariate log-normal priors, the correlations are fixed. In practical situation, when prior information is not sufficient to fix these correlations, they can be made adjustable by the following strategy. Take ℓ_h's for example. Let

$$\begin{aligned} \log(\ell_{h1}), \ldots, \log(\ell_{hd}) &\overset{iid}{\sim} \mathcal{N}(a_{h1}, b_{h1}), \text{ given } a_{h1},\ b_{h1}, \text{ and} \\ a_{h1} &\sim \mathcal{N}(a_{h2}, b_{h2}), \text{ given } a_{h2},\ b_{h2},\ h = 1, 2, 0. \end{aligned} \tag{4.28}$$

Then

$$\text{Cor}\left(\log(\ell_{hi}), \log(\ell_{hj})\right) = \frac{b_{h2}}{b_{h1} + b_{h2}},\ i, j = 1, \ldots, d,\ i \neq j, \text{ given } a_{h2},\ b_{h1},\ b_{h2},\ h = 1, 2, 0. \tag{4.29}$$

If b_{h2}'s are assigned higher level priors, then the correlations can be adjusted through updating b_{h2}'s (with b_{h1}'s either fixed or also assigned higher level priors). For λ_h's, the same strategy can be applied.

4.2.2 Scenario design

This subsection describes in details how the experiments under various scenarios are designed.

Four types of scenarios

Four types of scenarios are designed for generating the functions μ and ν with different degrees of correlation. The four types of scenarios are denoted by *c.gp, gp.c, gp.i, gp.d* and described next.

- Scenario *c.gp*: μ is a constant function and ν is generated using a Gaussian process model.
- Scenario *gp.c*: ν is a constant function and μ is generated using a Gaussian process model.
- Scenario *gp.i*: both μ and ν are generated using a Gaussian process models, independently.
- Scenario *gp.d*: both μ and ν are generated using a joint Gaussian process model, dependently.

Under the *c.gp* or *gp.c* scenario, μ and ν are completely uncorrelated. Under the *gp.i* scenario, μ and ν as two random functions are independent; however, the values of the particular μ and ν that are generated may end up correlated by chance with respect to random $\mathbf{x}$. Under the *gp.d* scenario, μ and ν as two random functions are dependent; therefore, particular μ and ν generated are likely to be more correlated than under the *gp.i* scenario. Note that under the *gp.c* scenario where the selection probability is a constant, the estimators $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ that use the selection probability as a covariate do not apply.

Generating μ and ν given hyperparameters

In order to evaluate the average performance of the estimators under each scenario, 20 pairs of μ and ν functions are generated independently for each scenario. For the constant μ or ν under the *c.gp* or *gp.c* scenario, 20 equally spaced values between 0.2 and 0.8 are selected. To generate the non-constant functions using Gaussian process models with given hyperparameters, the strategy of (2.59)-(2.60) is adopted and described in detail next.

- Step 1. For a given dimensionality d, generate 2000 $\mathbf{x}_0$'s $\overset{iid}{\sim} \mathcal{N}_d(\mathbf{0}, 2I_d)$, where I_d is the d dimensional identity matrix.
- Step 2. With the given hyperparameters, generate values of $g_\mu = g_1 + g_0$ and $g_\nu = g_2 + g_0$ at both $\mathbf{x}_0$'s and $-\mathbf{x}_0$'s, using the Cholesky decomposition of the 8000×8000 covariance matrix of g_μ and g_ν.
- Step 3. Get the values of the μ and ν functions at these $\mathbf{x}_0$'s and $-\mathbf{x}_0$'s by

$$\begin{aligned} \mu(\mathbf{x}) &= \Phi(g_\mu(\mathbf{x})) \\ \nu(\mathbf{x}) &= (1-0.1)\Phi(g_\nu(\mathbf{x})) + 0.1. \end{aligned} \tag{4.30}$$

  The μ function generated this way is used for the *gp.c*, *gp.i* and *gp.d* scenarios; the ν function generated this way is dependent on the generated μ function and used for the *c.gp* and *gp.d* scenarios.
- Step 4. Repeat Steps 2-3 20 times to get 20 independent pairs of dependent μ and ν functions.
- Step 5. Generate $g_\nu = g_2^* + g_0^*$ again from the given hyperparameters at $\mathbf{x}_0$'s and $-\mathbf{x}_0$'s with g_2^* and g_0^* independent of the previous g_2 and g_0.
- Step 6. Get the values of the ν function at $\mathbf{x}_0$'s and $-\mathbf{x}_0$'s from the newly generated g_ν by (4.30). This newly generated ν function is independent of the generated μ function and used for the *gp.i* scenario.
- Step 7. Repeat Steps 5-6 20 times to get 20 ν functions that are independent of the μ functions.

Generating $\mathbf{x}_i$, y_i, and r_i given μ and ν

For each given pair of μ and ν functions and a given sample size n, two sets of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are selected from the previously generated $\mathbf{x}_0$'s. (Note that for different pairs of functions under the same scenario, different sets of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are selected to avoid systematic error due to fixed $\mathbf{x}$'s). Given each set of $\mathbf{x}_1, \ldots, \mathbf{x}_n$, two sets of $y_1, \ldots, y_n$ and $r_1, \ldots, r_n$ are generated according to the corresponding μ values and ν values, respectively. Therefore, under each scenario, there are in total $20 \times 2 \times 2$ datasets. Note that having nested datasets allows for an analysis of variance (ANOVA).

Three sets of hyperparameters

To produce functions of different degrees of correlation and smoothness, three sets of hyperparameters are considered. The three sets of hyperparameters share common constant component hyperparameters, linear component hyperparameters, overall scaling hyperparameters for g_1 and g_2, and exponent hyperparameters as follow.

$$\begin{aligned}
&\sigma_{0,1}^2 = 0.5^2,\ \sigma_{0,2}^2 = 0.5^2,\ \sigma_{0,0}^2 = 0 \\
&\lambda_1 = (0.2, \ldots, 0.2),\ \lambda_2 = (0.3, \ldots, 0.3),\ \lambda_0 = (0.2, \ldots, 0.2) \\
&\eta_1 = 0.2,\ \eta_2 = 0.2 \\
&r_1 = 2,\ r_2 = 2,\ r_0 = 2 \qquad (4.31)
\end{aligned}$$

Three different sets of length-scale hyperparameters ℓ_h's and overall scaling hyperparameters η_0 for g_0 are selected to produce functions of different degrees of correlation and wiggliness (or smoothness) as given by (4.32)-(4.34). The first set has large length-scales and small value of η_0, therefore produces functions with low wiggliness and low correlation. The second set has the same length-scales as the first, but has larger η_0, and therefore produces functions of the same wiggliness but higher correlation. The third set has the same η_0 as the second, but has smaller length-scales, and therefore produces functions with both high wigglieness and high correlation. The three sets of hyperparameters are denoted by *ll, hl, hh*, respectively.

- Hyperparameter set *ll*:

$$\begin{aligned}
\ell_1 &= (2\exp(\sqrt{0.8}), \ldots, 2\exp(\sqrt{0.8})) \\
\ell_2 &= (2\exp(\sqrt{0.8}), \ldots, 2\exp(\sqrt{0.8})) \\
\ell_0 &= (\exp(\sqrt{0.8}), \ldots, \exp(\sqrt{0.8})) \\
\eta_0 &= 0.3 \qquad (4.32)
\end{aligned}$$

- Hyperparameter set *hl*:

$$\begin{aligned}
\ell_1 &= (2\exp(\sqrt{0.8}), \ldots, 2\exp(\sqrt{0.8})) \\
\ell_2 &= (2\exp(\sqrt{0.8}), \ldots, 2\exp(\sqrt{0.8})) \\
\ell_0 &= (\exp(\sqrt{0.8}), \ldots, \exp(\sqrt{0.8})) \\
\eta_0 &= 0.3\exp(1.3\sqrt{0.7}) \qquad (4.33)
\end{aligned}$$

- Hyperparameter set *hh*:

$$\begin{aligned} \ell_1 &= (0.2, \ldots, 0.2) \\ \ell_2 &= (0.2, \ldots, 0.2) \\ \ell_0 &= (0.1, \ldots, 0.1) \\ \eta_0 &= 0.3\exp(1.3\sqrt{0.7}) \end{aligned} \tag{4.34}$$

Note that $\log(\eta_0)$ under the hyperparameter set *ll* equals the prior mean of $\log(\eta_0)$ assigned to the GP estimators; $\log(\eta_0)$ under the hyperparameter set *hl* or *hh* is 1.3 standard deviations bigger than the prior mean of $\log(\eta_0)$ assigned to the GP estimators. Similarly, $\log(\ell)$'s under the hyperparameter set *ll* or *hl* are one standard deviation bigger than the corresponding prior means of $\log(\ell)$'s assigned to the GP estimators; $\log(\ell)$'s under the hyperparameter set *hh* are about 2.5 standard deviations smaller than the corresponding prior means of $\log(\ell)$'s assigned to the GP estimators. Such choices of hyperparameters for generating data guarantee the true hyperparameter values are reachable by the GP methods with reasonably large prior probabilities. In practice, having priors that cover the possible true parameter or hyperparameter values with reasonably large probabilities is the key to the success of a Bayesian method.

Also note that for different covariates $x_j, j = 1, \ldots, d$, the same linear component coefficients and length-scales are used. This is so that the generated functions are indeed of dimension d. If, instead, some x_j's dominated the others, the generated functions would actually resemble lower dimensional functions. Having the same linear component coefficients and length-scales makes all covariate variables equally relevant, which may not be the case in practice. However, our models which the GP estimators are based do not "know" this fact and therefore are valid for general situations.

Dimensionalities and sample sizes

For all scenarios, six dimensionalities ($d = 1,\ 2,\ 3,\ 5,\ 10,\ 20$) and two sample sizes ($n = 20,\ 50$) are considered.

Figure A.1 in Appendix gives two sample pairs of the μ and ν functions generated in one-dimensional space ($d = 1$) under {*hh, gp.d*}, {*hh, gp.i*}, {*hl, gp.d*}, {*hl, gp.i*}, {*ll, gp.d*}, and {*ll, gp.i*}, respectively.

4.2.3 Results and discussion

This subsection compares the Gaussian process (GP) estimators and the non-model based estimators on datasets generated under the various scenarios described in the previous subsection. All the estimators are compared in terms of mean squared error.

To compute the mean squared error, we first need to compute the *true* value of the population mean ϕ. For each μ function generated, the true value of ϕ is estimated by averaging the values of μ at $\mathbf{x}_0$'s and $-\mathbf{x}_0$'s, where $\mathbf{x}_0$'s and $-\mathbf{x}_0$'s are sampled as described in Subsection 4.2.2. (For a constant μ function, ϕ simply equals μ.) For the Horvitz-Thompson estimator $\widehat{\phi}_{HT_3}$ which requires knowledge of ψ, the *true* value of ψ is estimated the same way as for the true value of ϕ. Then, for each pair of the generated μ and ν functions, the mean squared error of each estimator conditional on the given pair of μ and ν is estimated by averaging over the 2×2 sets of $y_1, \ldots, y_n$ and $r_1, \ldots, r_n$. Since there are 20 independent pairs of μ and ν generated under each scenario, for each estimator, there are 20 estimated conditional mean squared errors which are independent of each other under each scenario. Then for any two estimators, a paired t-test can be performed on these estimated conditional mean squared errors under each scenario.

The mean squared errors (MSE) of all the estimators are presented along different dimensionalities and different scenarios in Figures 4.5 - 4.10 for different hyperparameter sets and different sample sizes, where for example, *d5* refers to the dimensionality $d = 5$. The results of paired t-tests on the conditional mean squared errors are given in Figures A.2-A.28 in the Appendix.

Note that none of the datasets simulated in this subsection have the effective sample size $n_{eff} = \sum_{i=1}^{n} r_i$ equal to zero. So the results presented in this subsection will be the same regardless how the $\widehat{\phi}_{naive}$, $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ estimators are defined when $\sum_{i=1}^{n} r_i = 0$.

Non-model based estimators

When the μ and ν functions are independent

When one of the μ and ν functions is a constant, i.e. under the *gp.c* or *c.gp* scenario, there is no potential selection bias. Under these two types of scenarios, the naive estimator $\widehat{\phi}_{naive}$ is the best among the four non-model based estimators (i.e. $\widehat{\phi}_{naive}$, $\widehat{\phi}_{HT_1}$, $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$), under all sets of hyperparameters: *hh*, *ll* and *hl*. The advantage of the naive estimator under these cases simply indicates that when the selection bias is not an issue, the Horvitz-Thompson estimators are subject to larger sampling errors than the naive estimator. (Note that $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ exactly equal $\widehat{\phi}_{naive}$ under the *gp.c* scenario).

Under the *gp.i* scenario, where the two functions are chosen independently, but will have some chance correlation, the selection bias can be an issue. However, the naive estimator is still better than the Horvitz-Thompson estimators (except for a few cases under the hyperparameter sets *hl* and *ll* where $\widehat{\phi}_{HT_2}$ or $\widehat{\phi}_{HT_3}$ is slightly but not significantly better than $\widehat{\phi}_{naive}$). The advantage of the naive estimator over the others under the *gp.i* scenario is bigger under the hyperparameter set *hh* than under

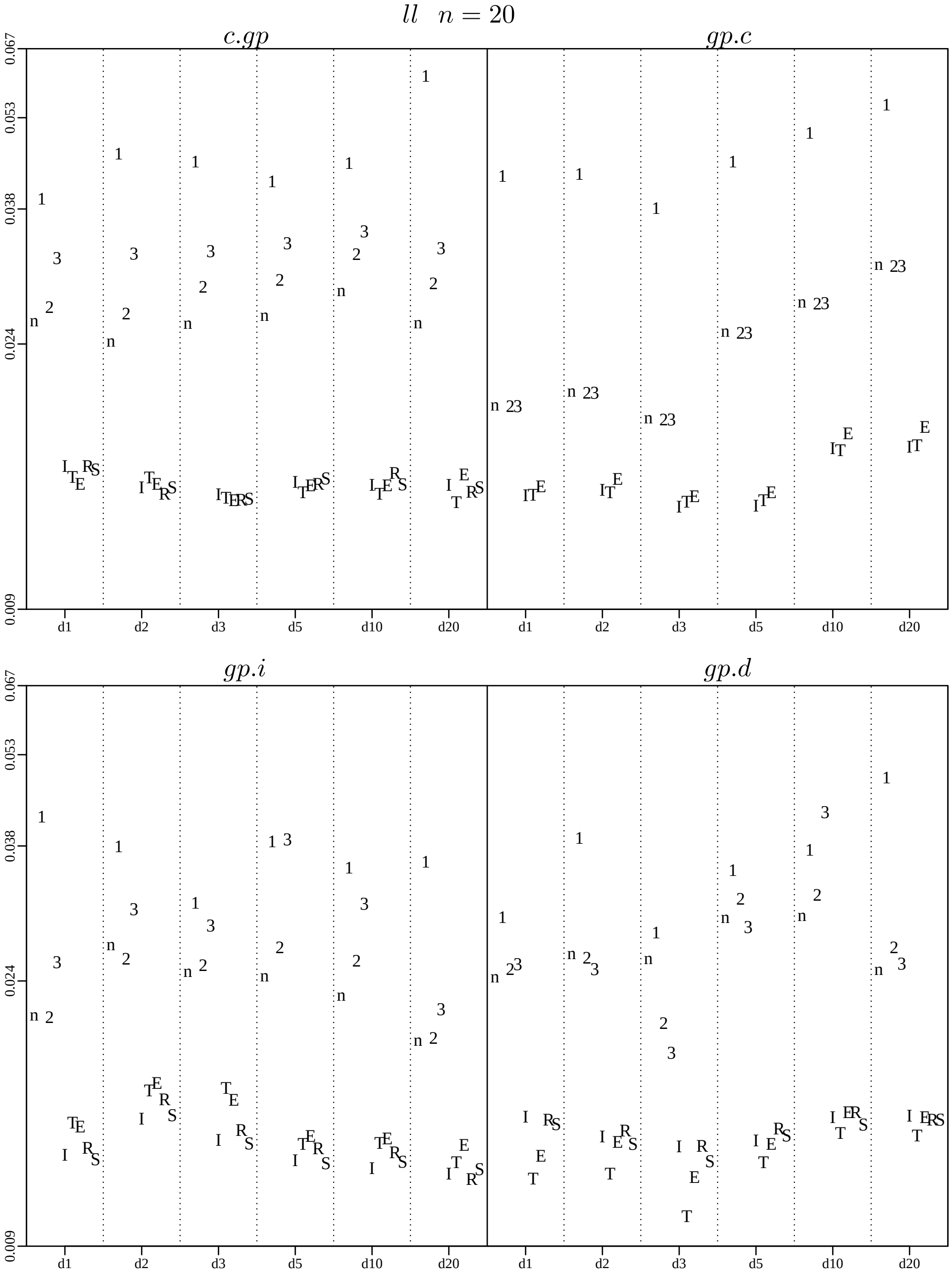


Figure 4.5: Low correlation and less wiggly. $n = 20$. Mean squared errors (MSE) in logarithm. ‘n’: $\widehat{\phi}_{naive}$; ‘1’: $\widehat{\phi}_{HT_1}$; ‘2’: $\widehat{\phi}_{HT_2}$; ‘3’: $\widehat{\phi}_{HT_3}$; ‘I’: $\widehat{\phi}_{GP_I}$; ‘T’:, $\widehat{\phi}_{GP_T}$; ‘E’: $\widehat{\phi}_{GP_E}$; ‘R’: $\widehat{\phi}_{GP_R}$; ‘S’: $\widehat{\phi}_{GP_S}$.

the hyperparameter sets *hl* and *ll*. One possible explanation is that with two independent functions both more wiggly under the hyperparameter set *hh*, it is more likely for the biases within each wiggled subregion to cancel out over the whole range of $\mathbf{x}$ to such a degree that the sampling errors dominate the selection bias. (More specifically, since the two functions are independent, by randomness they tend to be correlated positively in one subregion and negatively in another so that biases within subregions

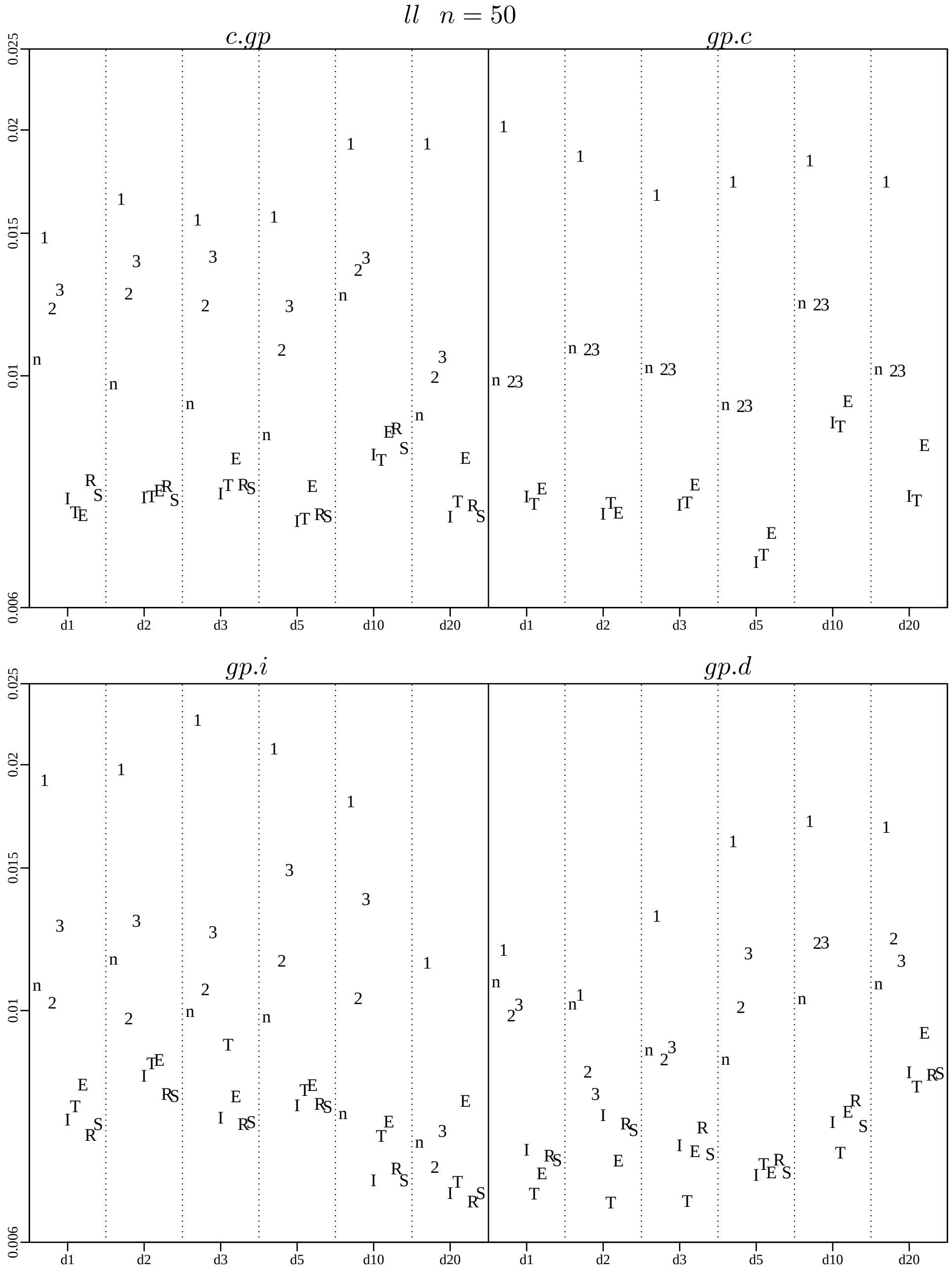


Figure 4.6: Low correlation and less wiggly. $n = 50$. Mean squared errors (MSE) in logarithm. 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$.

have both signs and tend to cancel out over a large number of subregions.)

The Horvitz-Thompson estimator $\widehat{\phi}_{HT_1}$, although unbiased, has, except five cases, been the worst among all the non-model based estimators under the scenarios *c.gp*, *gp.c* and *gp.i* for all hyperparameter sets. In the five cases (*ll*, $n = 20$, *gp.i*, $d = 5$*; hh*, $n = 50$, *c.gp*, $d = 2$*; hl*, $n = 20$, *gp.i*, $d = 3$*; hl*, $n = 20$, *gp.i*, $d = 5$*; hl*, $n = 50$, *c.gp*, $d = 2$) where $\widehat{\phi}_{HT_1}$ is not the worst, it seems better than $\widehat{\phi}_{HT_3}$

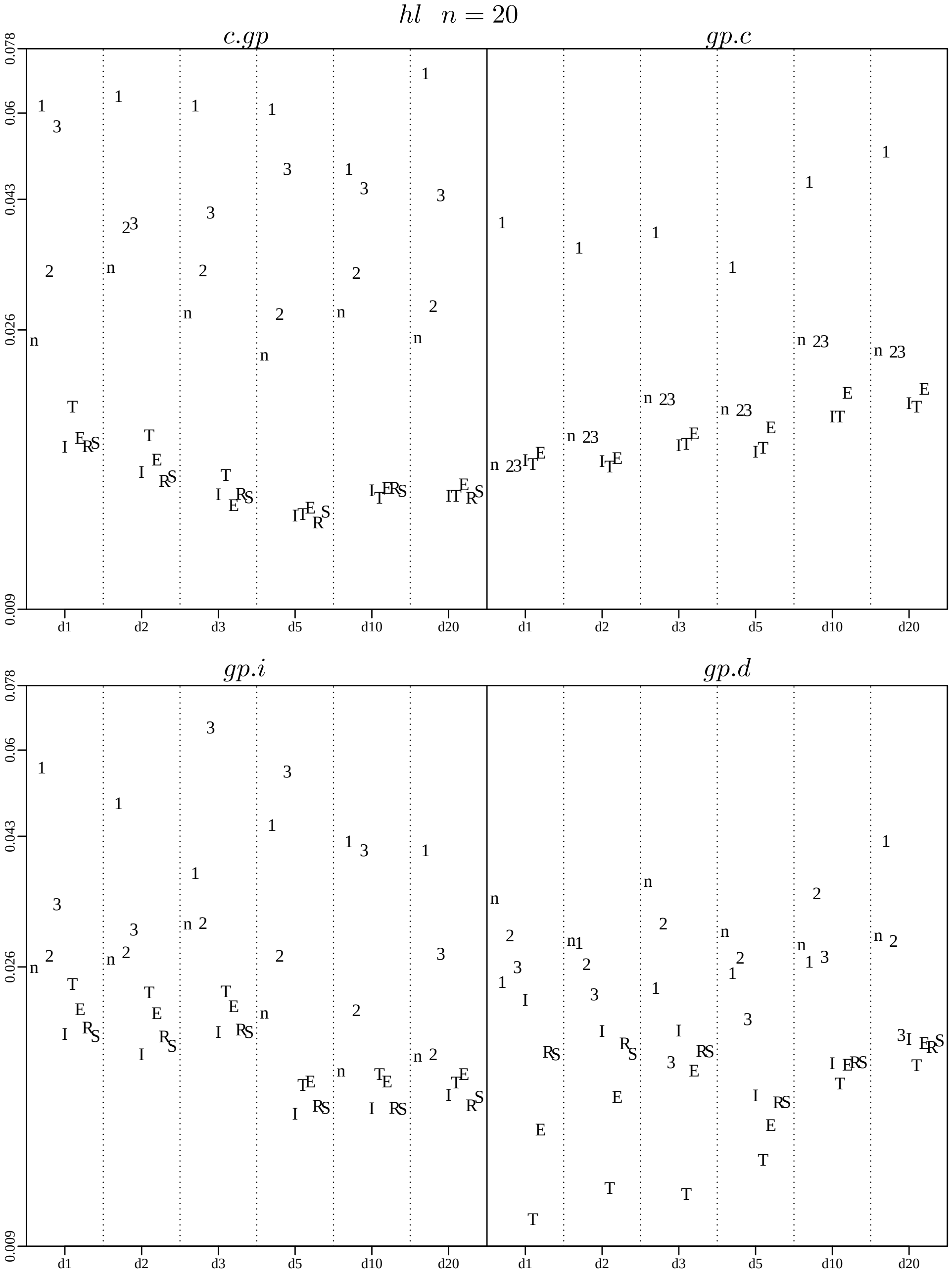


Figure 4.7: High correlation and less wiggly. $n = 20$. Mean squared errors (MSE) in logarithm. 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$.

but not significantly. Recall from (2.41) that the MSE of $\widehat{\phi}_{HT_3}$ is asymptotically smaller than that of $\widehat{\phi}_{HT_1}$ under all cases unless $\psi = 1$ or $\phi = 0$. Therefore, the smaller MSE of $\widehat{\phi}_{HT_1}$ under those five cases may be either due to the sample size n being not large enough or to the inaccurate estimates of the MSE's (which are based on *only 20* independently drawn pairs of functions under each scenario).

$\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ are both equal to the naive estimator under the *gp.c* scenario. However, $\widehat{\phi}_{HT_2}$ is

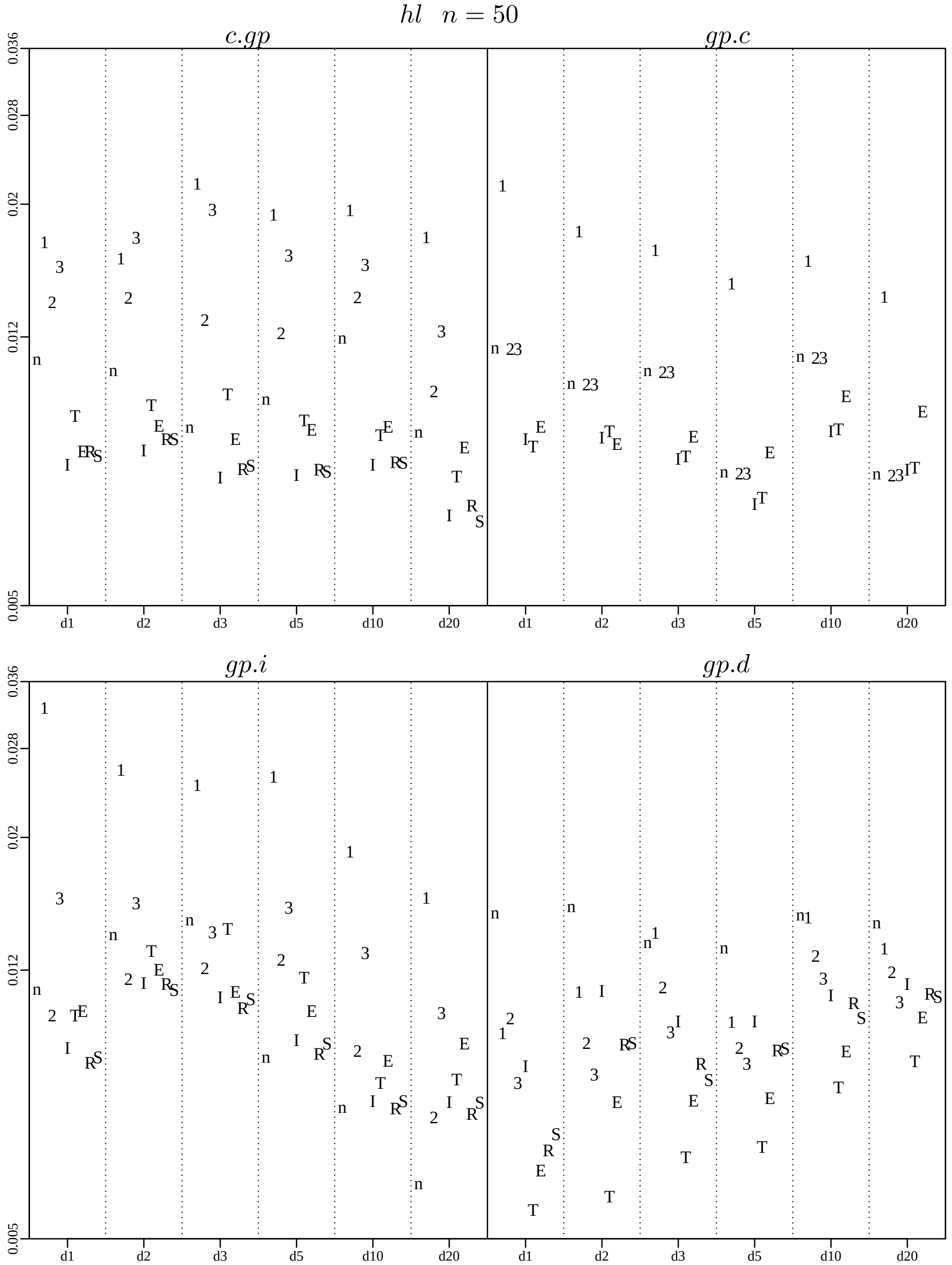


Figure 4.8: High correlation and less wiggly. $n = 50$. Mean squared errors (MSE) in logarithm. 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$.

almost always better than $\widehat{\phi}_{HT_3}$ under the scenarios *c.gp* and *gp.i*. (Where it is not, i.e *hh*, $n = 50$, *gp.i*, $d = 20$, the difference between these two estimators is tiny.) It is not surprising that $\widehat{\phi}_{HT_2}$ does better than $\widehat{\phi}_{HT_3}$ under the *c.gp* scenario due to it being equivariant in the extended sense where y has a constant mean function, as shown in (2.6). The reason why $\widehat{\phi}_{HT_2}$ is better than $\widehat{\phi}_{HT_3}$ under the *gp.i* scenario remains less clear.

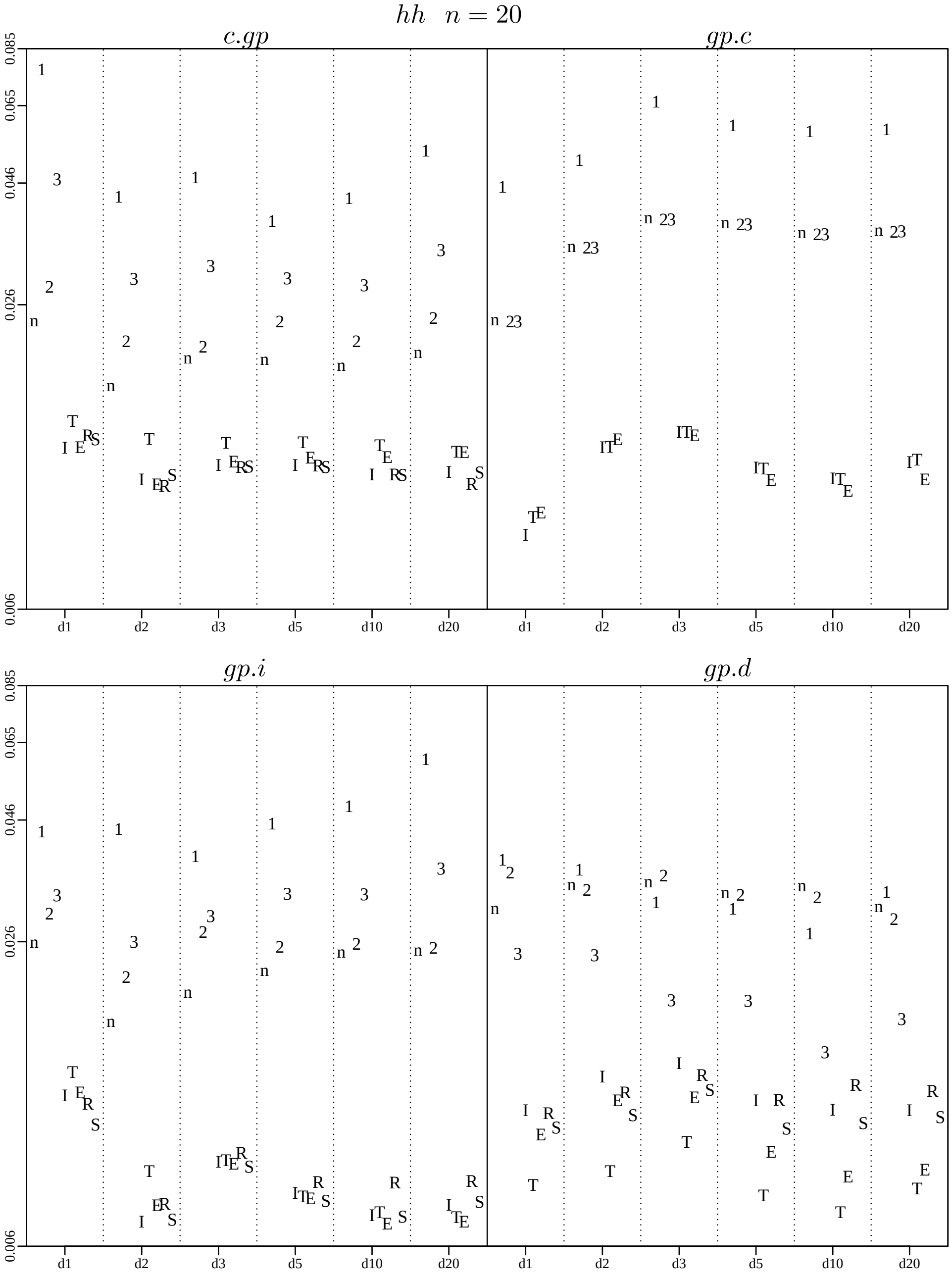


Figure 4.9: High correlation and highly wiggly. $n = 20$. Mean squared errors (MSE) in logarithm. 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$.

When the μ and ν functions are dependent

When the μ and ν functions are dependent, i.e. under the *gp.d* scenario, $\widehat{\phi}_{HT_1}$ eventually gains some advantage over the naive estimator under the hyperparameter set *hh* and when $n = 50$, since now the selection biases are strong and the sampling errors fade away. However, the advantage of $\widehat{\phi}_{HT_1}$ over $\widehat{\phi}_{naive}$ under this scenario is only significant for a couple of cases ($d = 1$ and $d = 10$). When $n = 20$

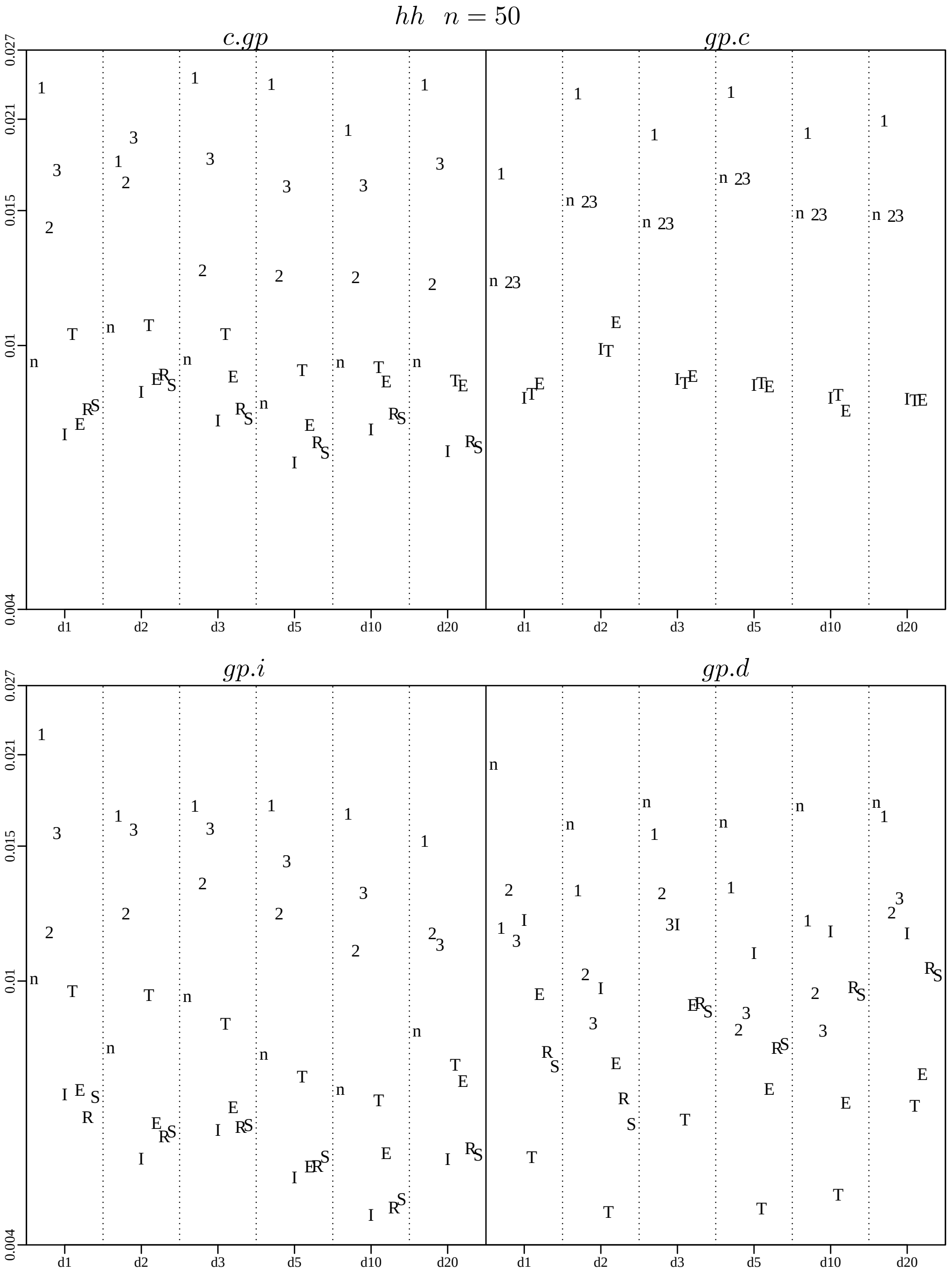


Figure 4.10: High correlation and highly wiggly. $n = 50$. Mean squared errors (MSE) in logarithm. 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$.

under the hyperparameter set *hh*, $\widehat{\phi}_{HT_1}$ is as bad as $\widehat{\phi}_{naive}$. Under the hyperparameter set *hl* where the dependency between the two functions is weaker , $\widehat{\phi}_{HT_1}$ seems better than the naive estimator in most cases, but none of the differences are significant. Under the hyperparameter set *ll* where the dependency between the two functions is the weakest, the naive estimator is actually always better than $\widehat{\phi}_{HT_1}$. (If the dependency between the two functions is even stronger or the sample size is even larger, we would expect to see more advantage of $\widehat{\phi}_{HT_1}$ over the naive estimator when selection bias eventually dominate sampling error.)

The Horvitz-Thompson estimators $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ also start gaining advantages over the naive estimator under the *gp.d* scenario, especially under the hyperparameter set *hh*. Under the hyperparameter set *hh*, $\widehat{\phi}_{HT_3}$ is the best among all the four non-model based estimators when $n = 20$, with the advantages being significant. Under the hyperparameter set *hh*, $\widehat{\phi}_{HT_2}$ performs as badly as $\widehat{\phi}_{HT_1}$ and the naive estimator when $n = 20$; when $n = 50$, $\widehat{\phi}_{HT_2}$ catches up with $\widehat{\phi}_{HT_3}$ and outperforms $\widehat{\phi}_{HT_1}$ and $\widehat{\phi}_{naive}$ significantly, due to its consistency and reduced bias with the larger sample size. Under the hyperparameter set *hl*, $\widehat{\phi}_{HT_3}$ still seems mostly the best among the non-model based estimators, but with no significant advantages. Compared to $\widehat{\phi}_{HT_1}$, $\widehat{\phi}_{HT_2}$ appears to improve more due to a larger sample size under the hyperparameter set *hl*. But the apparent advantages of $\widehat{\phi}_{HT_2}$ over both $\widehat{\phi}_{HT_1}$ and the naive estimator are not significant. Under the hyperparameter set *ll*, $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ are more often better and less often worse than the naive estimator under the *gp.d* scenario than under the other scenarios where the two functions are independent, although the differences between these three estimators are seldom significant. (As argued for $\widehat{\phi}_{HT_1}$, we can also expect to see more significant advantages of $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ over the naive estimator when the dependency between the two functions are stronger or the sample size is larger.)

Summary

According to the above analyses, when selection bias is strong, $\widehat{\phi}_{HT_3}$ tends to outperform both $\widehat{\phi}_{HT_1}$ and the naive estimator. $\widehat{\phi}_{HT_2}$, although not having so much advantage over $\widehat{\phi}_{HT_1}$ or the naive estimator when the sample size is small, catches up with $\widehat{\phi}_{HT_3}$ with a larger sample size. When the two functions are less correlated, although $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ are not better, or may even be worse, than the naive estimator, they outperform $\widehat{\phi}_{HT_1}$ most of the time. The Horvitz-Thompson estimator $\widehat{\phi}_{HT_1}$ has no general advantages over the other non-model based estimators in all the scenarios investigated. The overall poor performance of $\widehat{\phi}_{HT_1}$ is not surprising as its inefficiency has been recognized by many researchers (e.g. Rotnitzky et al., 2012; Scharfstein et al., 1999; Kang and Schafer, 2007).

$\widehat{\phi}_{HT_3}$ seems superior to $\widehat{\phi}_{HT_2}$ under the *gp.d* scenario where selection bias is strong. However, note that under the *gp.d* scenario, $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ as random functions are positively correlated and likely have a positive correlation with respect to $\mathbf{x}$. The advantage of $\widehat{\phi}_{HT_3}$ over $\widehat{\phi}_{HT_2}$ under the *gp.d* scenario where $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ are positively correlated coincides with the results found in Section 4.1. How the performance of $\widehat{\phi}_{HT_3}$ is affected by the sign of the correlation between $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ requires further investigation. (Note that $\widehat{\phi}_{HT_2}$ will not be affected by the sign of the correlation between $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ due to its equivarance under all affine transformations). Nevertheless, $\widehat{\phi}_{HT_3}$ requires knowledge of the marginal selection probability $\psi = \int \nu(\mathbf{x}) dF_{\mathbf{X}}$ and is therefore not as widely applicable as $\widehat{\phi}_{HT_2}$.

Model based estimators

$\widehat{\phi}_{GP_T}$ versus $\widehat{\phi}_{GP_E}$

When the relationship between the μ and ν functions are modeled correctly, or in other words, when the selection probabilities are incorporated correctly, we would expect that $\widehat{\phi}_{GP_T}$ should outperform $\widehat{\phi}_{GP_E}$, since $\widehat{\phi}_{GP_T}$ is based on the true selection probabilities while $\widehat{\phi}_{GP_E}$ is based on estimated ones. Under the *gp.d* scenario, where the two functions are generated using dependent Gaussian process priors that are similar to the adaptable Gaussian process priors assigned for $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$, $\widehat{\phi}_{GP_T}$ indeed outperforms $\widehat{\phi}_{GP_E}$ except one case (*ll*, $n = 50$, *gp.d*, $d = 5$) where all the GP estimators are about the same. Under the hyperparameter set *hh*, the advantages of $\widehat{\phi}_{GP_T}$ over $\widehat{\phi}_{GP_E}$ are the strongest, often marginally significant and sometimes even significant with p-values < 0.01. Under the hyperparameter set *hl*, $\widehat{\phi}_{GP_T}$ outperforms $\widehat{\phi}_{GP_E}$ with marginal significance half of the time. Under the hyperparameter set *ll* where selection bias is the weakest, $\widehat{\phi}_{GP_T}$ is still better than $\widehat{\phi}_{GP_E}$ except when $d = 5$ and $n = 50$, but the advantages of $\widehat{\phi}_{GP_T}$ over $\widehat{\phi}_{GP_E}$ are not large.

When the μ and ν functions are independent, $\widehat{\phi}_{GP_T}$ seems worse than $\widehat{\phi}_{GP_E}$ under the scenarios *c.gp* and *gp.i*, especially under the hyperparameter set *hh* when $n = 50$. Under the *gp.c* scenario where the selection probability is a constant, the differences between $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ are rather small. Actually, it is not clear what we would expect for the differences between $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ under these scenarios, since models that lack dependence of μ and ν have only small probabilities under the priors assigned for both $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$.

$\widehat{\phi}_{GP_R}$ versus $\widehat{\phi}_{GP_S}$

First note that $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ are not applicable under the scenario *gp.c* where the selection probability is a constant. Otherwise, the differences between $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ are rather small under all scenarios. Although we would expect that $\widehat{\phi}_{GP_S}$ should be better than $\widehat{\phi}_{GP_R}$, since $\widehat{\phi}_{GP_S}$ is based on a larger set of $\mathbf{x}$ sampled from the distribution of $\mathbf{x}$, it is not clear how much better $\widehat{\phi}_{GP_S}$ is. If the advantage of $\widehat{\phi}_{GP_S}$ over $\widehat{\phi}_{GP_R}$ is small as in the scenarios considered, then practically when a larger set of $\mathbf{x}$ is not available and $\widehat{\phi}_{GP_S}$ is thus not applicable, we would not be too concerned whether the results by $\widehat{\phi}_{GP_R}$ would be slightly better if $\widehat{\phi}_{GP_S}$ were used instead.

$\widehat{\phi}_{GP_T}$ versus $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$

Although $\widehat{\phi}_{GP_T}$, $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ all use the true selection probabilities, they take different approaches. Recall that $\widehat{\phi}_{GP_T}$ incorporates the selection probability by modeling the μ function conditional on the true selection probabilities using dependent priors for the μ and ν functions, while $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ only model the μ function, but with the selection probability ν as an additional covariate.

Under the *gp.d* scenario where the two functions are generated dependently using joint Gaussian process priors, $\widehat{\phi}_{GP_T}$ has its model closer to the true model than $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ do. Therefore, we would expect that $\widehat{\phi}_{GP_T}$ is better than $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ under the *gp.d* scenario. Indeed, under the *gp.d* scenario, $\widehat{\phi}_{GP_T}$ is better than $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ except one case (*ll,* $n = 50$*, gp.d,* $d = 5$) where all the GP estimators are about the same. The advantage of $\widehat{\phi}_{GP_T}$ is the strongest under the hyperparameter set *hh* where selection bias is the strongest, and the weakest under the hyperparameter set *ll* where selection bias is the weakest.

Under the scenarios *c.gp* and *gp.i* where the two functions are independent, $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ seem generally better than $\widehat{\phi}_{GP_T}$, with a few cases where the advantages of $\widehat{\phi}_{GP_R}$ and/or $\widehat{\phi}_{GP_S}$ over $\widehat{\phi}_{GP_T}$ are (marginally) significant. Note that $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ are not applicable under the scenario *gp.c* where the selection probability is a constant.

$\widehat{\phi}_{GP_E}$ *versus* $\widehat{\phi}_{GP_R}$ *and* $\widehat{\phi}_{GP_S}$

Under the *gp.d* scenario, $\widehat{\phi}_{GP_E}$ (like $\widehat{\phi}_{GP_T}$) has its model closer to the true model than $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ do. But, $\widehat{\phi}_{GP_E}$ is based on the estimated selection probabilities while $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ use the true selection probabilities. Therefore, it is not obvious whether $\widehat{\phi}_{GP_E}$ would do better or worse than $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ under the *gp.d* scenario. Under the hyperparameter set *ll*, there are no significant differences from $\widehat{\phi}_{GP_E}$ to $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$. Under the hyperparameter set *hl*, $\widehat{\phi}_{GP_E}$ seems generally better than $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$, but the differences are mostly not significant. Under the hyperparameter set *hh*, $\widehat{\phi}_{GP_E}$ perform better than $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ for larger dimensionalities d, possibly because in the higher dimensional spaces, the generated functions are more complex and therefore having the right model is more important than having the right selection probabilities.

Under the scenarios *c.gp* and *gp.i*, $\widehat{\phi}_{GP_E}$ (like $\widehat{\phi}_{GP_T}$) seems generally worse than $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$, with a few cases where the advantages of $\widehat{\phi}_{GP_R}$ and/or $\widehat{\phi}_{GP_S}$ over $\widehat{\phi}_{GP_E}$ are (marginally) significant. Note that $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ are not applicable under the scenario *gp.c* where the selection probability is a constant.

$\widehat{\phi}_{GP_I}$ *versus* $\widehat{\phi}_{GP_T}$, $\widehat{\phi}_{GP_E}$, $\widehat{\phi}_{GP_R}$ *and* $\widehat{\phi}_{GP_S}$

Theoretically, we would expect that $\widehat{\phi}_{GP_I}$ should perform worse than the other GP estimators when the μ and ν functions are dependent and better when the two functions are independent. Under the scenario *gp.d* where selection bias is the strongest, $\widehat{\phi}_{GP_I}$ is indeed the worst among all the GP estimators except where $\widehat{\phi}_{GP_I}$ is pretty close to $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$. For *hh* when $n = 20$ and $d = 10$ or 20, $\widehat{\phi}_{GP_I}$ seems significantly better than $\widehat{\phi}_{GP_R}$ but with an advantage much less than the advantages of $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ over $\widehat{\phi}_{GP_R}$.

Under the scenarios *c.gp* and *gp.i*, the differences between $\widehat{\phi}_{GP_I}$ and $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ are rather small except two cases (*hh,* $n = 20$*, gp.i,* $d = 10$*; hh,* $n = 20$*, gp.i,* $d = 20$) where $\widehat{\phi}_{GP_I}$ is (marginally) significantly better than $\widehat{\phi}_{GP_R}$. Under the scenarios *c.gp* and *gp.i*, $\widehat{\phi}_{GP_E}$ is also often close to $\widehat{\phi}_{GP_I}$. Where it is not, $\widehat{\phi}_{GP_E}$ is only slightly worse than $\widehat{\phi}_{GP_I}$. Since, as discussed earlier, under the scenarios *c.gp* and *gp.i*, $\widehat{\phi}_{GP_T}$ is generally worse than $\widehat{\phi}_{GP_E}$ (although the differences are not dramatic), it is not surprising that the advantages of $\widehat{\phi}_{GP_I}$ over $\widehat{\phi}_{GP_T}$ under these scenarios are more obvious than the advantages of $\widehat{\phi}_{GP_I}$ over $\widehat{\phi}_{GP_E}$. However, the advantages of $\widehat{\phi}_{GP_I}$ over $\widehat{\phi}_{GP_T}$ under these scenarios are not very large too, and often not significant. Under the scenario *gp.c* where $\widehat{\phi}_{GP_R}$ and $\widehat{\phi}_{GP_S}$ are not applicable, $\widehat{\phi}_{GP_I}$, $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ are rather close each other.

Although we would expect $\widehat{\phi}_{GP_I}$ to perform better under the scenarios where the μ and ν functions are independent, it often does not have obvious advantages over the other GP estimators. When it does, the advantages are not large. Although under the priors assigned for the other GP estimators, independent μ and ν functions only occur with small probabilities, the other GP estimators are able to model the μ function comparably well when it is independent of the ν function.

Model based v.s. Non-model based estimators

Under the various scenarios considered, all the model based estimators generally perform better than all the non-model based estimators, often with large and significant advantages, whether selection bias is strong or not. In particular, $\widehat{\phi}_{GP_I}$ is often significantly better than the Horvitz-Thompson estimators when selection bias is not strong (or not present). When selection bias is the strongest (i.e. under *hh* and *gp.d*), compared to $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$, $\widehat{\phi}_{GP_I}$ does comparably well when $n = 50$ and significantly better when $n = 20$. This is contrary to what one might expect from the arguments by Robins and Ritov (1997) and by Ritov et al. (2013) that when the sample size is small and the μ function is complex, any Bayesian method that fails to consider the selection probability will not do as well as the Horvitz-Thompson estimator $\widehat{\phi}_{HT_1}$. (Note that, $\widehat{\phi}_{HT_1}$ is even worse than either of $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ or both of them in all scenarios considered.)

Summary

Based on the above analyses, whether selection bias is strong or not, having an appropriate model for μ is important for improving the efficiency for estimating the population mean ϕ. A method like Horvitz-Thompson that totally ignores the covariates, $\mathbf{x}$, can be very inefficient, even if it is unbiased or consistent. When the μ function is dependent on the selection probability, a method like $\widehat{\phi}_{GP_I}$, although consistent by having a correct model for μ, may also not perform well enough with limited sample sizes. Therefore, it is best to have a method which not only corrects selection bias but also exploits the covariate information as much as possible. Some have argued (e.g. Robins and Ritov, 1997; Kang and Schafer, 2007) that in complex situations, where the dependency between the μ and the ν functions is complex, it is impossible to have a model for μ that is flexible enough to effectively capture the relationship between $\mathbf{x}$ and $\mu(\mathbf{x})$ with a limited sample size. However, the results analyzed in this subsection demonstrate that Gaussian process model based methods (even with the selection probability being ignored) can efficiently model complex functions with better performance than those non-model based methods most of time. These results also demonstrate that Gaussian process models can be implemented effectively enough for their benefits to be realized in practice.

4.3 An example due to Kang and Schafer

This section studies an example from Kang and Schafer (2007) as described next. Suppose that a covariate vector $\mathbf{z} = (z_1, z_2, z_3, z_4)$ is distributed as $N(\mathbf{0}, \mathbf{I})$ where $\mathbf{I}$ is the 4×4 identity matrix. Given $\mathbf{z}$, the real-valued response variable y is determined by,

$$y = \mu_0(\mathbf{z}) + \epsilon = 210 + 27.4z_1 + 13.7z_2 + 13.7z_3 + 13.7z_4 + \epsilon, \tag{4.35}$$

where $\epsilon \sim N(0, 1)$. The response variable y is observed if $r = 1$, with the selection probability function being

$$\nu_0(\mathbf{z}) = \Pr(r = 1|\mathbf{z}) = \text{expit}(-z_1 + 0.5z_2 - 0.25z_3 - 0.1z_4) \tag{4.36}$$

Instead of observing the z_j's, suppose it is the following covariates that are observed

$$\begin{aligned} x_1 &= \exp(z_1/2) \\ x_2 &= z_2/(1+\exp(z_1)) + 10 \\ x_3 &= (z_1 z_3/25 + 0.6)^3 \\ x_4 &= (z_2 + z_4 + 20)^2 \end{aligned} \tag{4.37}$$

Denote $\mathbf{x} = (x_1, x_2, x_3, x_4)$. Since $\Pr(z_2 + z_4 < -20) \approx 0$, the mapping between these two covariate vectors $\mathbf{z}$ and $\mathbf{x}$ is practically one-to-one. Therefore, if the true relationships were known, y could be predicted from $\mathbf{x}$ as well as it could be from $\mathbf{z}$. However, with x_j's being the covariates, the functions $\mu(\mathbf{x}) = \text{E}[y|\mathbf{x}]$ and $\nu(\mathbf{x}) = \Pr(r = 1|\mathbf{x})$ are more complicated than $\mu_0(\mathbf{z})$ and $\nu_0(\mathbf{z})$. Since we pretend we observed the wrong covariate vector $\mathbf{x}$, we assume that the distribution of $\mathbf{x}$ is unknown.

Note that in this artificial example, the selection probability ν is not bounded away from zero in the original paper by Kang and Schafer. As discussed earlier, in practice, it is essential to have the selection probability bounded away from zero. Otherwise, the survey is not sufficiently well designed, and any inference problem based on it would be inherently difficult. In this section, I will consider both the cases where the selection probability is not bounded or is bounded away from zero by 0.05.

4.3.1 Setup

For this example, the ordinary least squares (OLS) estimator based on the linear regression model (without interaction terms) will be included for comparison, in addition to the estimators considered in Section 4.2. Since we do not know the distribution of $\mathbf{x}$, $\phi = \mathrm{E}[\mu(\mathbf{x})]$ will be estimated using only the observed $\mathbf{x}$'s as in (3.22). In such a case, the estimator $\widehat{\phi}_{GP_S}$ does not apply.

Since y is real-valued, for the Gaussian process estimators that use the latent function g_μ, the link function from g_μ to μ is simply the identity function. Then, y is modeled as

$$y = \mu(\mathbf{x}) + \epsilon = g_\mu(\mathbf{x}) + \epsilon \tag{4.38}$$

where the predictor function g_μ has a prior based on Gaussian process model and the noise term ϵ has a normal distribution $\mathcal{N}(0, \delta^2)$ and is independent of the predictor g_μ. When $\mu(\mathbf{x}) = g_\mu(\mathbf{x})$, the latent vector $\mathbf{g}_\mu^{(n)}$, i.e. the values of g_μ at $\mathbf{x}_1, \ldots, \mathbf{x}_n$, no longer needs to be updated. Therefore, for the estimators $\widehat{\phi}_{GP_I}$, $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_R}$, no latent vector will be updated, while for $\widehat{\phi}_{GP_E}$, only $\mathbf{g}_\nu^{(n)}$, i.e. the values of g_ν at $\mathbf{x}_1, \ldots, \mathbf{x}_n$, will be updated. (Whether a latent vector is updated or not substantially affects the computing time, as will be discussed in Section 4.4.)

For the Gaussian process estimators, the same priors for g_μ and g_ν from Section 4.2 will be used. In addition to model g_μ and g_ν, we also need to model the noise standard deviation δ. The following log-normal distribution is selected for δ,

$$\log(\delta) \sim \mathcal{N}(\log(2.4), 0.8) \tag{4.39}$$

Such a prior for δ is chosen so that it covers a range of values of the noise standard deviation with reasonably large probabilities, including the true value of 1.

Since the x_j's under this example differ dramatically in scale from the x_j's generated in Section 4.2, we need to transform them so that they have about the same range as those in Section 4.2, since the same priors will be used. The x_j's are transformed by

$$\begin{aligned} &x_1 \to \sqrt{2}(x_1 - 1)/0.6, && x_2 \to \sqrt{2}(x_2 - 10)/0.5 \\ &x_3 \to \sqrt{2}(x_3 - 0.2)/0.04, && x_4 \to \sqrt{2}(x_4 - 400)/55 \end{aligned} \tag{4.40}$$

These transformations of x_j's are what one might do after looking at the summary statistics or the scatterplots of the observed x_j's. Similarly, we also need to subtract 200 from y and then divide the difference by 40 so that it has about the same range as the latent function g_μ generated in Section 4.2. This transformation of y is also what one might do after looking at the summary statistics or the scatterplot of the observed y's, and therefore does not build in knowledge of the true mean of y. (Note that transforming the observed data with the priors for the hyperparameters kept the same is equivalent to transforming the priors for the hyperparameters with the observed data kept the same.) Consequently, the prior for the noise standard deviation becomes

$$\log(\delta) \sim \mathcal{N}(\log(0.06), 0.8) \tag{4.41}$$

For convenience, the transformed x_j's are also used for the OLS estimators and the transformed y is also used for both the OLS estimator and the Horvitz-Thompson estimators. For the Horvitz-Thompson estimators $\widehat{\phi}_{HT_1}$ and $\widehat{\phi}_{HT_3}$ which are non-equivariant under the transfomation: $y \to y + c$, $c \neq 0$, the results would be different if the untransformed y is used, as will be discussed more later in this section.

Robins et al. (2007) have pointed out that under the original selection mechanism of this example, the selection bias is not very strong and therefore it is not surprising that the ordinary least squares (OLS) estimator performed well or even better than some of the doubly robust estimators considered in Kang and Schafer (2007). Robins et al. (2007) have also showed that if the selection mechanism is reversed, i.e. y being observed with the selection probability function $1 - \nu(\mathbf{x})$, the performance of the OLS estimator is no longer acceptable. Therefore, in this section all the estimators will be studied under both the original and the reversed selection mechanisms. When considering the bounded selection probability, the selection probability equals $0.95\nu(\mathbf{x}) + 0.05$ under the original selection and equals $0.95(1 - \nu(\mathbf{x})) + 0.05$ under the reversed selection.

In the original paper, Kang and Schafer considered two sample sizes, $n = 200$ and $n = 1000$, with $m = 1000$ simulated datasets for each. However, due to the limited amount of time, only datasets of sample size $n = 200$ will be studied. Also to save time, the Gaussian process estimators will be studied only with $m = 100$ simulated datasets. The frequentist estimators that are fast to compute will be studied with $m = 1000$ simulated datasets.

For the estimators $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ which are based on Gaussian process models that assign only positive correlation for the functions g_μ and g_ν and therefore favors $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ with positive correlation (w.r.t $\mathbf{x}$), the sign of y needs to be reversed, since the true correlation between $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ (w.r.t $\mathbf{x}$) is negative (-0.61). Under the reversed selection mechanism, y is kept with its original sign, since the correlation between $\mu(\mathbf{x})$ and the reversed selection probability $1 - \nu(\mathbf{x})$ is positive. (Generalisation of $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ so that the correlation between g_μ and g_ν is adjustable will be discussed in the last chapter.)

For the estimators $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$, ν will not be bounded away from zero when being linked from the corresponding latent function g_ν, whether the true selection probability is bounded or not. This will, however, make the estimator $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ slightly different from what they are in Section 4.2. Robustness of $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ against whether ν is bounded or not will be discussed later in this section. Note that $\widehat{\phi}_{GP_I}$ and $\widehat{\phi}_{GP_R}$ will not be affected, since they do not have a model for ν.

For the Horvitz-Thompson estimators, true selection probabilities at the observed covariates are used, typical in survey problems where selection probabilities are often available. For the estimator $\widehat{\phi}_{HT_3}$, it is also the true marginal selection probability ψ that is used. If ψ is hard to obtain in practice, $\widehat{\phi}_{HT_3}$ will not be applicable.

4.3.2 Results

First note that none of the datasets simulated in this section have the effective sample size $n_{eff} = \sum_{i=1}^n r_i$ equal to zero. So the results presented in this subsection will be the same regardless how the $\widehat{\phi}_{naive}$, $\widehat{\phi}_{HT_2}$ and $\widehat{\phi}_{HT_3}$ estimators are defined when $\sum_{i=1}^n r_i = 0$.

When the selection probability is not bounded

We first look at the results when the selection probability is not bounded. The results of the frequentist estimators obtained from $m = 1000$ datasets are given in Table 4.1. Table 4.2 gives the results for the Gaussian process estimators and two of the frequentist estimators based on $m = 100$ datasets. All the results are presented based on the original scale of y.

Obviously, as shown in Table 4.1, $\widehat{\phi}_{naive}$ must be severely biased and has an extremely large MSE compared to the other estimators under both the original and reversed selection mechanisms. Under the original selection mechanism, all comparisons between the frequentist estimators are significant according to the paired t-tests except for $\widehat{\phi}_{HT_1}$ and $\widehat{\phi}_{HT_3}$. $\widehat{\phi}_{HT_2}$ seems better than $\widehat{\phi}_{HT_1}$ and $\widehat{\phi}_{HT_3}$ under both selection mechanisms, but not significantly when the selection is reversed. (Note that from the result in Section 2.1, we may expect that $\widehat{\phi}_{HT_3}$ is better than $\widehat{\phi}_{HT_1}$ under both selection mechanisms, which seems true but not significantly.)

The OLS estimator $\widehat{\phi}_{OLS}$ does substantially better than all the other frequentist estimators under the original selection mechanism. This is no surprise, as Robins et al. (2007) point out that selection bias is not very strong under the original selection mechanism. Unlike the naive estimator, $\widehat{\phi}_{OLS}$, although also ignoring the selection probability, does consider the covariates x_j's, although not in a

		MSE		p-value			
				$\widehat{\phi}_{naive}$	$\widehat{\phi}_{HT_1}$	$\widehat{\phi}_{HT_2}$	$\widehat{\phi}_{HT_3}$
A. Original Selection	$\widehat{\phi}_{naive}$	110.9	(2.3)				
	$\widehat{\phi}_{HT_1}$	29.5	(1.9)	0.000			
	$\widehat{\phi}_{HT_2}$	22.6	(1.2)	0.000	0.000		
	$\widehat{\phi}_{HT_3}$	29.2	(1.9)	0.000	0.572	0.000	
	$\widehat{\phi}_{OLS}$	11.9	(0.5)	0.000	0.000	0.000	0.000
B. Reversed Selection	$\widehat{\phi}_{naive}$	110.5	(2.2)				
	$\widehat{\phi}_{HT_1}$	33.9	(12.9)	0.000			
	$\widehat{\phi}_{HT_2}$	26.5	(3.5)	0.000	0.447		
	$\widehat{\phi}_{HT_3}$	30.7	(9.8)	0.000	0.314	0.523	
	$\widehat{\phi}_{OLS}$	31.9	(1.0)	0.000	0.877	0.135	0.902

Table 4.1: Mean squared errors (MSE) of $\widehat{\phi}_{naive}$, $\widehat{\phi}_{HT_1}$, $\widehat{\phi}_{HT_2}$, $\widehat{\phi}_{HT_3}$ and $\widehat{\phi}_{OLS}$ (with standard errors in brackets) and p-values of paired t-tests on the squared errors, based on $\boldsymbol{m} = \mathbf{1000}$ datasets of sample size $n = 200$. The selection probability is **not bounded away** from zero. Note: a p-value of "0.000" means "< 0.0005".

very effective way. Therefore, $\widehat{\phi}_{OLS}$ is able to remove part of selection bias and even does better than the HT estimators which are unbiased or consistent but ignore the covariates totally. When the selection is reversed and selection bias is strong, $\widehat{\phi}_{OLS}$ no longer outperforms the HT estimators. But the differences between $\widehat{\phi}_{OLS}$ and the HT estimators are not significant.

Note that the results for the OLS estimator in Table 4.1 obtained with $m = 1000$ datasets are similar to those by Kang and Schafer (2007) and by the discussants of Kang and Schafer (2007).

		MSE		p-value				
				$\widehat{\phi}_{HT_2}$	$\widehat{\phi}_{OLS}$	$\widehat{\phi}_{GP_I}$	$\widehat{\phi}_{GP_T}$	$\widehat{\phi}_{GP_E}$
A. Original Selection	$\widehat{\phi}_{HT_2}$	21.9	(3.3)					
	$\widehat{\phi}_{OLS}$	11.0	(1.4)	0.003				
	$\widehat{\phi}_{GP_I}$	7.5	(0.9)	0.000	0.000			
	$\widehat{\phi}_{GP_T}$	7.0	(0.8)	0.000	0.000	0.101		
	$\widehat{\phi}_{GP_E}$	7.6	(1.0)	0.000	0.007	0.866	0.294	
	$\widehat{\phi}_{GP_R}$	5.9	(0.7)	0.000	0.000	0.035	0.050	0.029
B. Reversed Selection	$\widehat{\phi}_{HT_2}$	21.1	(4.0)					
	$\widehat{\phi}_{OLS}$	29.9	(2.6)	0.081				
	$\widehat{\phi}_{GP_I}$	10.1	(1.2)	0.007	0.000			
	$\widehat{\phi}_{GP_T}$	8.0	(1.0)	0.001	0.000	0.000		
	$\widehat{\phi}_{GP_E}$	8.1	(1.0)	0.001	0.000	0.000	0.877	
	$\widehat{\phi}_{GP_R}$	5.7	(0.7)	0.000	0.000	0.000	0.002	0.002

Table 4.2: Mean squared errors (MSE) of $\widehat{\phi}_{HT_2}$, $\widehat{\phi}_{OLS}$, $\widehat{\phi}_{GP_I}$, $\widehat{\phi}_{GP_T}$, $\widehat{\phi}_{GP_E}$ and $\widehat{\phi}_{GP_R}$ (with standard errors in brackets) and p-values of paired t-tests on the squared errors, based on $\boldsymbol{m} = \mathbf{100}$ datasets of sample size $n = 200$. The selection probability is **not bounded away** from zero. Note: a p-value of "0.000" means "< 0.0005".

Table 4.2 gives the results of the Gaussian process estimators with comparison to both $\widehat{\phi}_{OLS}$ and

$\widehat{\phi}_{HT_2}$ (the best among the HT estimators according to Table 4.1). Obviously, all the GP estimators perform better than both $\widehat{\phi}_{OLS}$ and $\widehat{\phi}_{HT_2}$ under either selection mechanism. The superority of the GP estimators is because they, whether ignoring the selection probability or not, can model the complex function $\mu(\mathbf{x})$ more effectively, compared to $\widehat{\phi}_{OLS}$ which is based on a linear regression model and $\widehat{\phi}_{HT_2}$ which has no model at all.

Among the GP estimators, $\widehat{\phi}_{GP_R}$ is best under both selection mechanisms. Under the original selection, $\widehat{\phi}_{GP_T}$ seems better than $\widehat{\phi}_{GP_E}$ but not significantly, and is only better than $\widehat{\phi}_{GP_I}$ with a marginal significance. The difference between $\widehat{\phi}_{GP_E}$ and $\widehat{\phi}_{GP_I}$ is rather small under the original selection. When selection bias is strong under the reversed selection, all the differences between the GP estimators are significant, except for $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$, the difference between which is almost invisible. Unlike $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$, $\widehat{\phi}_{GP_R}$ does not assume a link function between the latent function g_ν and the selection probability ν. (It actually does not model ν or have g_ν at all.) Both $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ impose a probit link function between g_ν and ν which may not match the truth well. This difference of $\widehat{\phi}_{GP_R}$ from $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ may explain why $\widehat{\phi}_{GP_R}$ does better than $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$, although they have all exploited the selection probability in some way. As discussed in earlier sections, $\widehat{\phi}_{GP_T}$ is expected to be better than $\widehat{\phi}_{GP_E}$ if the selection probabilities are correct *and* the model for the dependency between the functions μ and ν is also correct. In this example, however, we are not sure how the μ and ν functions are actually related. If the Gaussian process priors which $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ are based on do not have a high probability for the true relationship between μ and ν, $\widehat{\phi}_{GP_T}$ is not necessarily better than $\widehat{\phi}_{GP_E}$ as seen in Table 4.2. $\widehat{\phi}_{GP_E}$, although about the same as $\widehat{\phi}_{GP_I}$ under the original selection, indeed does better when the selection is reversed. That is, when selection bias is strong, even with a highly flexible model for y, incorporating the selection bias in some appropriate way can help achieve better results. Note that $\widehat{\phi}_{GP_E}$ is the only estimator that does not require knowledge of ν but still exploits the selection probability through a joint model for μ and ν.

When the selection probability is bounded

Results with the selection probability bounded away from zero by 0.05 are presented in Tables 4.3 and 4.4, where the selection probability equals $0.95\nu(\mathbf{x}) + 0.05$ under the original selection and equals $0.95(1 - \nu(\mathbf{x})) + 0.05$ under the reversed selection.

Compared to the results in Tables 4.1 and 4.2, the Horvitz-Thompson estimators have improved quite a bit under both selection mechanisms. Particularly, all the HT methods are now significantly better than the OLS method under the reversed selection, although still much worse than the OLS method under the original selection. The substantial improvement of the HT estimators when the selection probability is bounded away from zero is not surprising, since they are well known to be sensitive

		MSE		p-value			
				$\widehat{\phi}_{naive}$	$\widehat{\phi}_{HT_1}$	$\widehat{\phi}_{HT_2}$	$\widehat{\phi}_{HT_3}$
A. Original Selection	$\widehat{\phi}_{naive}$	92.7	(2.0)				
	$\widehat{\phi}_{HT_1}$	23.0	(1.2)	0.000			
	$\widehat{\phi}_{HT_2}$	18.6	(0.9)	0.000	0.000		
	$\widehat{\phi}_{HT_3}$	22.5	(1.1)	0.000	0.078	0.000	
	$\widehat{\phi}_{OLS}$	11.5	(0.5)	0.000	0.000	0.000	0.000
B. Reversed Selection	$\widehat{\phi}_{naive}$	94.5	(2.0)				
	$\widehat{\phi}_{HT_1}$	16.0	(0.7)	0.000			
	$\widehat{\phi}_{HT_2}$	18.3	(0.8)	0.000	0.000		
	$\widehat{\phi}_{HT_3}$	15.5	(0.7)	0.000	0.005	0.000	
	$\widehat{\phi}_{OLS}$	26.3	(0.9)	0.000	0.000	0.000	0.000

Table 4.3: Mean squared errors (MSE) of $\widehat{\phi}_{naive}$, $\widehat{\phi}_{HT_1}$, $\widehat{\phi}_{HT_2}$, $\widehat{\phi}_{HT_3}$ and $\widehat{\phi}_{OLS}$ (with standard errors in brackets) and p-values of paired t-tests on the squared errors, based on $\boldsymbol{m} = \mathbf{1000}$ datasets of sample size $n = 200$. The selection probability is **bounded away** from zero by 0.05. Note: a p-value of "0.000" means "< 0.0005".

		MSE		p-value						
				$\widehat{\phi}_{HT_1}$	$\widehat{\phi}_{HT_2}$	$\widehat{\phi}_{HT_3}$	$\widehat{\phi}_{OLS}$	$\widehat{\phi}_{GP_I}$	$\widehat{\phi}_{GP_T}$	$\widehat{\phi}_{GP_E}$
A. Original Selection	$\widehat{\phi}_{HT_1}$	23.0	(3.5)							
	$\widehat{\phi}_{HT_2}$	18.1	(2.5)	0.002						
	$\widehat{\phi}_{HT_3}$	23.6	(3.7)	0.484	0.000					
	$\widehat{\phi}_{OLS}$	11.7	(1.4)	0.004	0.030	0.004				
	$\widehat{\phi}_{GP_I}$	7.1	(0.8)	0.000	0.000	0.000	0.000			
	$\widehat{\phi}_{GP_T}$	6.7	(0.8)	0.000	0.000	0.000	0.000	0.117		
	$\widehat{\phi}_{GP_E}$	7.0	(0.9)	0.000	0.000	0.000	0.001	0.856	0.518	
	$\widehat{\phi}_{GP_R}$	6.0	(0.7)	0.000	0.000	0.000	0.000	0.061	0.111	0.062
B. Reversed Selection	$\widehat{\phi}_{HT_1}$	13.3	(2.1)							
	$\widehat{\phi}_{HT_2}$	15.5	(2.4)	0.008						
	$\widehat{\phi}_{HT_3}$	13.0	(2.1)	0.421	0.000					
	$\widehat{\phi}_{OLS}$	23.9	(2.4)	0.000	0.003	0.000				
	$\widehat{\phi}_{GP_I}$	8.7	(1.2)	0.026	0.002	0.037	0.000			
	$\widehat{\phi}_{GP_T}$	7.2	(1.0)	0.002	0.000	0.004	0.000	0.000		
	$\widehat{\phi}_{GP_E}$	7.2	(1.0)	0.003	0.000	0.004	0.000	0.000	0.934	
	$\widehat{\phi}_{GP_R}$	5.6	(0.6)	0.000	0.000	0.000	0.000	0.001	0.014	0.013

Table 4.4: Mean squared errors (MSE) of $\widehat{\phi}_{HT_1}$, $\widehat{\phi}_{HT_2}$, $\widehat{\phi}_{HT_3}$, $\widehat{\phi}_{OLS}$, $\widehat{\phi}_{GP_I}$, $\widehat{\phi}_{GP_T}$, $\widehat{\phi}_{GP_E}$ and $\widehat{\phi}_{GP_R}$ (with standard errors in brackets) and p-values of paired t-tests on the squared errors, based on $\boldsymbol{m} = \mathbf{100}$ datasets of sample size $n = 200$. The selection probability is **bounded away** from zero by 0.05. Note: a p-value of "0.000" means "< 0.0005".

to extremely small selection probabilities, and are incapable of extrapolating into regions where no observations are available, since they totally ignore $\mathbf{x}$.

The GP estimators have also improved compared to the results in Table 4.2, presumably due to both the availability of observations in the previously empty regions and the slightly increased effective sample sizes. (The average effective sample size over 100 datasets has increased from 99.4 to 104.4 under the original selection and from 100.6 to 104.5 under the reversed selection.) Similar to the results in Table 4.2, all the GP methods are substantially better than both the HT methods and the OLS method under both selection mechanisms. $\widehat{\phi}_{GP_R}$ also remains the best among all the GP methods under both selection mechanisms. The advantage of $\widehat{\phi}_{GP_E}$ over $\widehat{\phi}_{GP_I}$ is still more signifincant under the reversed selection where selection bias is strong.

Note that not getting any observations in regions with extremely small selection probabilities imposes an inherently difficult problem for all methods. But this problem is less severe for model-based methods for which extrapolation into those empty regions is possible, depending on how good the models are. This is one of the reasons why the GP methods are less sensitive to extremely small selection probabilities than the HT methods. Another reason why the HT methods are more sensitive to extremely small selection probabilities is inherent to all methods based on inverse probability weighting where the weights can be extremely large due to the extremely small selection probabilities.

Comparison to some doubly robust methods

The results of some doubly robust (DR) methods by several authors are summarized in Tables 4.5-4.7. These DR methods are based on estimated selection probabilities using the linear logistic regression either on the original covariates z_j's or on the transformed covariates x_j's, as shown in Tables 4.5-4.7. Note that results by these authors are based on datasets generated using unbounded selection probabilities.

		MSE	
		fit ν with $\mathbf{z}$	fit ν with $\mathbf{x}$
A. Original Selection	KS_{BC}	10.8	166.9
	KS_{WLR}	8.3	14.7
	KS_R	8.6	12.3
	KS_{SRR_R}	22.0	1.8×10^{10}

Table 4.5: Results by Kang and Schafer (2007) with $m = 1000$ datasets. BC: bias-corrected linear regression; WLR: linear regression with inverse probability weighted coefficients; R: linear regression using selection probability based covariates; SRR_R: using the inverse selection probability as a covariate as proposed by Scharfstein et al. (1999). For more details, see Kang and Schafer (2007).

Since the true selection probability is indeed a linear expit function of the original covariates z's, I assume that their estimated selection probabilities based on z's are very close to the true values. Even so, it may not be completely fair to compare their results to the results by GP_T and GP_R which use the true selection probabilities. Therefore, I will only compare their results to those by GP_E and GP_I, where GP_E estimates the selection probabilities using the transformed covariates x's and GP_I totally ignores the selection probabilities. Please note that since these authors did not provide the standard errors of their estimated MSE, and since all their results are based on $m = 1000$ datasets and mine are

		MSE	
		fit ν with $\mathbf{z}$	fit ν with $\mathbf{x}$
A. Original Selection	$RSLR_{DR(\hat{\pi},\hat{m}_{REG})}$	12.12	169.91
	$RSLR_{DR(\hat{\pi},\hat{m}_{WLS})}$	9.24	14.74
	$RSLR_{DR(\hat{\pi},\hat{m}_{DR-IPW-NR})}$	7.40	14.90
	$RSLR_{B-DR(\hat{\pi},\hat{m}_{REG})}$	11.83	54.65
	$RSLR_{B-DR(\hat{\pi}_{EXT},\hat{m}_{REG})}$	9.69	16.82
B. Reversed Selection	$RSLR_{DR(\hat{\pi},\hat{m}_{REG})}$	13.11	19.90
	$RSLR_{DR(\hat{\pi},\hat{m}_{WLS})}$	9.02	18.24
	$RSLR_{DR(\hat{\pi},\hat{m}_{DR-IPW-NR})}$	7.76	17.90
	$RSLR_{B-DR(\hat{\pi},\hat{m}_{REG})}$	11.76	19.69
	$RSLR_{B-DR(\hat{\pi}_{EXT},\hat{m}_{REG})}$	11.12	19.55

Table 4.6: Results by Robins et al. (2007) with $m = 1000$ datasets. $\hat{\pi}$: the estimated ν or the ν-model; $\hat{m}$: the estimated μ or the y-model, "B-DR": DR robust methods that guarantee that the estimated ϕ fall into the parameter space of ϕ: $\hat{\pi}_{EXT}$: the selection probabilities are estimated using an extended linear logistic regression with an additional user-supplied covariate $h(\mathbf{z})$ or $h(\mathbf{x})$. For more details, see Robins et al. (2007).

		MSE	
		fit ν with $\mathbf{z}$	fit ν with $\mathbf{x}$
A. Original Selection	$TAN_{AIPW_{fix}}$	11.8	158.8
	TAN_{WLS}	8.9	15.3
	$TAN_{REG_{tilde}}$	7.5	12.0
	$TAN_{REG_{hat}}$	7.9	13.5
	$TAN_{REG^{(m)}_{tilde}}$	7.5	12.7
	$TAN_{REG^{(m)}_{hat}}$	7.0	13.5

Table 4.7: Results by Tan (2007) with $m = 1000$ datasets. For more details, see Tan (2007).

based on $m = 100$ datasets, I assume that their results are "accurate" compared to mine. The standard errors of the estimated MSE by GP_I and GP_E are given in Table 4.2.

Comparing results in Table 4.2 and Tables 4.5-4.7, GP_I and GP_E clearly do much better than all these DR methods when the selection probabilities are estimated using $\mathbf{x}$. (Note that GP_E is based on estimated selection probabilities using $\mathbf{x}$ and GP_I totally ignores the selection probabilities.) Even when compared to the results by the DR methods based on the true model for ν, i.e. using $\mathbf{z}$, GP_I and GP_E still do better than most of the DR methods and do comparably well compared to the best of these DR methods. The inefficiency of these DR methods when both models are wrong (i.e. both models are based on $\mathbf{x}$) confirms that only being "doubly" robust is not enough and we need highly flexible models that are robust against various situations.

4.3.3 Discussion

Non-equivariance of $\widehat{\phi}_{HT_1}$ and $\widehat{\phi}_{HT_3}$

Since $\widehat{\phi}_{HT_1}$ and $\widehat{\phi}_{HT_3}$ are non-quivariant under the transformation: $y \to y + c$, where $c \neq 0$, the results of $\widehat{\phi}_{HT_1}$ and $\widehat{\phi}_{HT_3}$ will be different if based on the original y's. Table 4.8 gives the results of $\widehat{\phi}_{HT_1}$ and $\widehat{\phi}_{HT_3}$ using the untransformed y's, which dramatically differ from the results using the transformed y's as in Tables 4.1 and 4.3. Whether the results are better or worse under different transformations, non-equivariance is just undesirable.

	Unbounded Selection Probability		Bounded Selection Probability	
	Original Selection	Reversed Selection	Original Selection	Reversed Selection
$\widehat{\phi}_{HT_1}$	543.0 (28.9)	359.6 (23.0)	414.6 (19.7)	276.6 (11.5)
$\widehat{\phi}_{HT_3}$	308.8 (20.7)	132.9 (9.2)	199.6 (10.3)	74.2 (3.3)

Table 4.8: Mean squared errors (MSE) of $\widehat{\phi}_{HT_1}$ and $\widehat{\phi}_{HT_3}$ (with standard errors in brackets) based on the original y's with $m = 1000$ datasets.

Robustness of GP_T and GP_E against whether ν is assumed bounded or not

In practice, it is common that the selection probability is bounded away from zero. However, it is not always clear where the bound is. When using a method, like GP_T and GP_E, that assumes a model on the selection probability ν, it is desired that the results do not vary a lot whether the model assumes ν is bounded or not. Therefore, I also consider the variants of GP_T and GP_E that assume the selection probability is bounded away from zero by 0.05 and denote them by GP_{Tb} and GP_{Eb}, respectively. For the datasets generated with selection probabilities bounded away from zero by 0.05, we would expect that GP_{Tb} and GP_{Eb} will do better than GP_T and GP_E.

The results of GP_{Tb} and GP_{Eb} for the datasets generated using bounded selection probabilities are presented in Table 4.9 along with the results of GP_T and GP_E. As shown in Table 4.9, the difference between GP_E and GP_{Eb} is not significant under both selection mechanisms. The difference between GP_{Tb} and GP_T is not significant, either, under the reversed selection. Under the original selection, GP_{Tb} indeed seems significantly better than GP_T, but the difference between GP_{Tb} and GP_T is rather small, particularly smaller than the differences from $\widehat{\phi}_{GP_T}$ to $\widehat{\phi}_{GP_E}$ and $\widehat{\phi}_{GP_R}$ as in Table 4.4.

Robustness of GP_T and GP_R against the transformation of ν

Interestingly, I also ran into some variants of the estimators $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_R}$ due to a mistake. Recall that the link function from g_ν to ν assumed for $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ is the probit function. Therefore, for $\widehat{\phi}_{GP_T}$, it is $g_{\nu,i} = \Phi^{-1}(\nu(\mathbf{x}_i))$, $i = 1, \ldots, 200$, that should be used as the "known" g_ν. However,

		MSE		p-value		
				$\widehat{\phi}_{GP_T}$	$\widehat{\phi}_{GP_{Tb}}$	$\widehat{\phi}_{GP_E}$
A. Original Selection	$\widehat{\phi}_{GP_T}$	6.7	(0.8)			
	$\widehat{\phi}_{GP_{Tb}}$	6.5	(0.8)	0.044		
	$\widehat{\phi}_{GP_E}$	7.0	(0.9)	0.518	0.293	
	$\widehat{\phi}_{GP_{Eb}}$	7.1	(0.9)	0.468	0.263	0.855
B. Reversed Selection	$\widehat{\phi}_{GP_T}$	7.2	(1.0)			
	$\widehat{\phi}_{GP_{Tb}}$	7.2	(1.0)	0.471		
	$\widehat{\phi}_{GP_E}$	7.2	(1.0)	0.934	0.736	
	$\widehat{\phi}_{GP_{Eb}}$	7.0	(1.0)	0.504	0.699	0.266

Table 4.9: Mean squared errors (MSE) of $\widehat{\phi}_{GP_T}$, $\widehat{\phi}_{GP_{Tb}}$, $\widehat{\phi}_{GP_E}$, and $\widehat{\phi}_{GP_{Eb}}$ (with standard errors in brackets) and p-values of paired t-tests on the squared errors, based on $m = 100$ datasets of sample size $n = 200$. The selection probability is **bounded away** from zero by 0.05. GP_T and GP_E assume ν is not bounded away from zero; GP_{Tb} and GP_{Eb} assume ν is bounded away from zero by 0.05. Note: a p-value of "0.000" means "< 0.0005".

I originally used $g_{\nu,i} = \text{logit}(\nu(\mathbf{x}_i))$, $i = 1, \ldots, 200$, instead. Having different $g_{\nu,i}$'s would alter the prior relationship between μ and the fixed ν, since the prior relationship between g_μ and g_ν is fixed, thereby making the comparison between $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_E}$ more complicated. After having corrected this mistake, I found that the result is very different, a bit surprisingly. The difference between these two $\widehat{\phi}_{GP_T}$'s is presumably due to the sensitivity of the prior for g_μ conditional on $g_{\nu,i}$'s (which $\widehat{\phi}_{GP_T}$ is based on) against the scaling of $g_{\nu,i}$'s. This has prompted me to wonder if similar issues may occur to $\widehat{\phi}_{GP_R}$, if a different transformation of ν is used as the additional covariate. Therefore, I also considered two variants of $\widehat{\phi}_{GP_R}$ with the additional covariate $x_5 = \Phi^{-1}(\nu(\mathbf{x}))$ or $x_5 = 6\nu(\mathbf{x}) - 3$, respectively. These two variants of $\widehat{\phi}_{GP_R}$ are denoted by $\widehat{\phi}_{GP_{R2}}$ and $\widehat{\phi}_{GP_{R3}}$, respectively. The variant of $\widehat{\phi}_{GP_T}$ with $g_{\nu,i} = \text{logit}(\nu(\mathbf{x}_i))$, $i = 1, \ldots, 200$, is denoted by $\widehat{\phi}_{GP_{T2}}$.

To compare these different versions of $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_R}$, I consider both situations where the true selection probability is bounded away from zero or not. For datasets generated using bounded selection probabilities, I also consider the respective variants of $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_{T2}}$ which assume that ν is bounded away from zero by 0.05. Similar to $\widehat{\phi}_{GP_{Tb}}$, this variant of $\widehat{\phi}_{GP_{T2}}$ is denoted by $\widehat{\phi}_{GP_{T2b}}$.

Table 4.10 gives the results for these different versions of $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_R}$ with datasets generated using unbounded selection probabilities. Table 4.11 gives the results with datasets generated using bounded selection probabilities.

According to Table 4.10, $\widehat{\phi}_{GP_{T2}}$ does differ from $\widehat{\phi}_{GP_T}$ significantly under the reserved selection. Under the original selection where selection bias is not strong, it is conceivable that $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_{T2}}$ may be less sensitive to the scaling of $g_{\nu,i}$'s and therefore differ less. The three versions of $\widehat{\phi}_{GP_R}$'s are, however, close to each other under either selection mechanism.

According to Table 4.11, when the selection probability is bounded away from zero, the differences

		MSE	p-value			
			$\widehat{\phi}_{GP_T}$	$\widehat{\phi}_{GP_{T2}}$	$\widehat{\phi}_{GP_R}$	$\widehat{\phi}_{GP_{R2}}$
A. Original Selection	$\widehat{\phi}_{GP_T}$	7.0 (0.8)				
	$\widehat{\phi}_{GP_{T2}}$	6.5 (0.7)	0.150			
	$\widehat{\phi}_{GP_R}$	5.9 (0.7)	0.050	0.185		
	$\widehat{\phi}_{GP_{R2}}$	6.0 (0.7)	0.077	0.278	0.140	
	$\widehat{\phi}_{GP_{R3}}$	5.9 (0.7)	0.026	0.057	0.888	0.536
B. Reversed Selection	$\widehat{\phi}_{GP_T}$	8.0 (1.0)				
	$\widehat{\phi}_{GP_{T2}}$	6.8 (0.8)	0.0002			
	$\widehat{\phi}_{GP_R}$	5.7 (0.7)	0.0025	0.0416		
	$\widehat{\phi}_{GP_{R2}}$	5.7 (0.7)	0.0018	0.0297	0.450	
	$\widehat{\phi}_{GP_{R3}}$	5.9 (0.7)	0.0004	0.0143	0.593	0.425

Table 4.10: Mean squared errors (MSE) of $\widehat{\phi}_{GP_T}$, $\widehat{\phi}_{GP_{T2}}$, $\widehat{\phi}_{GP_R}$, $\widehat{\phi}_{GP_{R2}}$ and $\widehat{\phi}_{GP_{R3}}$ (with standard errors in brackets) and p-values of paired t-tests on the squared errors, based on $m = 100$ datasets of sample size $n = 200$. The selection probability is **not bounded away** from zero. Note: a p-value of "0.000" means "< 0.0005".

		MSE	p-value						
			$\widehat{\phi}_{GP_T}$	$\widehat{\phi}_{GP_{Tb}}$	$\widehat{\phi}_{GP_{T2}}$	$\widehat{\phi}_{GP_{T2b}}$	$\widehat{\phi}_{GP_R}$	$\widehat{\phi}_{GP_{R2}}$	$\widehat{\phi}_{GP_{R3}}$
A. Original Selection	$\widehat{\phi}_{GP_T}$	6.7 (0.8)							
	$\widehat{\phi}_{GP_{Tb}}$	6.5 (0.8)	0.044						
	$\widehat{\phi}_{GP_{T2}}$	6.0 (0.7)	0.006	0.045					
	$\widehat{\phi}_{GP_{T2b}}$	6.2 (0.7)	0.035	0.149	0.391				
	$\widehat{\phi}_{GP_R}$	6.0 (0.7)	0.111	0.230	0.913	0.581			
	$\widehat{\phi}_{GP_{R2}}$	6.0 (0.7)	0.122	0.252	0.983	0.633	0.361		
	$\widehat{\phi}_{GP_{R3}}$	6.0 (0.7)	0.103	0.226	0.989	0.593	0.776	0.939	
B. Reversed Selection	$\widehat{\phi}_{GP_T}$	7.2 (1.0)							
	$\widehat{\phi}_{GP_{Tb}}$	7.2 (1.0)	0.471						
	$\widehat{\phi}_{GP_{T2}}$	6.7 (0.9)	0.068	0.114					
	$\widehat{\phi}_{GP_{T2b}}$	6.2 (0.8)	0.001	0.002	0.088				
	$\widehat{\phi}_{GP_R}$	5.6 (0.6)	0.014	0.017	0.026	0.163			
	$\widehat{\phi}_{GP_{R2}}$	5.6 (0.6)	0.012	0.015	0.021	0.146	0.653		
	$\widehat{\phi}_{GP_{R3}}$	5.6 (0.6)	0.006	0.007	0.009	0.100	0.927	0.976	

Table 4.11: Mean squared errors (MSE) of $\widehat{\phi}_{GP_T}$, $\widehat{\phi}_{GP_{Tb}}$, $\widehat{\phi}_{GP_{T2}}$, $\widehat{\phi}_{GP_{T2b}}$, $\widehat{\phi}_{GP_R}$, $\widehat{\phi}_{GP_{R2}}$ and $\widehat{\phi}_{GP_{R3}}$ (with standard errors in brackets) and p-values of paired t-tests on the squared errors, based on $m = 100$ datasets of sample size $n = 200$. The selection probability is **bounded away** from zero by 0.05. GP_T and GP_{T2} assume that ν is not bounded away from zero; GP_{Tb} and GP_{T2b} assume that ν is bounded away from zero by 0.05. Note: a p-value of "0.000" means "< 0.0005".

between the different versions of $\widehat{\phi}_{GP_R}$ are, again, rather close to each other, under both selection mechanisms. The difference between $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_{T2}}$ is significant under the original selection and marginally significant under the reversed selection. The difference between $\widehat{\phi}_{GP_{T2}}$ and $\widehat{\phi}_{GP_{T2b}}$ is not significant under the original selection, but significant under the reversed selection.

The sensitivity (or robustness) of $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_R}$ against the scaling of $g_{\nu,i}$'s or the transformation of ν could be clarified further by additional investigations.

Estimated noise standard deviations (δ)

Since the mapping between the original covariates z_j's and x_j's is practically one-to-one, we expect that modeling on x_j's should be as good as on z_j's, if the model is good enough. A perfect model would figure out nearly the exact relationship between y and all the covariates x_j's and thus have the estimated noise standard deviation δ close to its true value. A less perfect model tends to miss some of the important effects and thus produce a larger estimate for the noise standard deviation.

		Estimated noise standard deviation (δ) (true $\delta = 1$)			
		Min	Median	Mean	Max
A. Original Selection	on 1000 datasets				
	OLS	10.4	14.7	14.7	22.9
	on 100 datasets				
	OLS	10.9	14.9	14.9	22.9
	GP_I	0.6	5.5	5.3	9.8
	GP_T	1.0	3.6	3.6	7.6
	GP_{Tb}	1.0	3.5	3.6	7.6
	GP_{T2}	1.3	2.3	2.5	6.6
	GP_{T2b}	1.3	2.0	2.1	5.9
	GP_E	0.6	4.3	4.3	9.7
	GP_{Eb}	0.6	4.3	4.3	9.9
	GP_R	0.8	1.0	1.0	1.2
	GP_{R2}	0.8	1.0	1.0	1.2
	GP_{R3}	0.9	1.1	1.1	1.3
B. Reversed Selection	on 1000 datasets				
	OLS	9.4	14.6	14.6	22.7
	on 100 datasets				
	OLS	10.9	14.5	14.6	22.7
	GP_I	1.5	5.6	5.6	9.0
	GP_T	1.6	3.8	3.9	6.9
	GP_{Tb}	1.2	3.6	3.7	8.0
	GP_{T2}	1.3	2.3	2.5	6.3
	GP_{T2b}	1.2	2.1	2.2	6.4
	GP_E	1.0	4.1	4.1	8.1
	GP_{Eb}	1.1	3.9	4.1	8.5
	GP_R	0.8	1.0	1.1	1.2
	GP_{R2}	0.8	1.0	1.0	1.2
	GP_{R3}	0.8	1.0	1.0	1.3

Table 4.12: Minimum, median, mean and maximum of the estimated noise standard deviations by each method on $m = 1000$ or $m = 100$ datasets. The selection probability is **bounded away** from zero by 0.05.

Table 4.12 gives the estimated noise standard deviations by each method on datasets generated using bounded selection probabilities. Clearly, by the OLS method, the estimated noise standard deviations are much larger than those by the GP methods, indicating that the OLS method is far from figuring out the true relationship between y and x_j's. By the GP_I method, also ignoring the selection probability, the estimated noise standard deviations are much smaller than those by the OLS method, indicating that the GP_I method, although still much less than perfect, can fit the relationship between y and x_j's substantially better than the OLS method. All the other GP methods have the estimated standard deviations much smaller than those by GP_I. Particularly, all the three versions of GP_R have the estimated noise standard deviations pretty close to the true value of 1.

As may be noted in Table 4.12, by the methods GP_E and GP_R, the estimated noise standard deviations are sometimes smaller than the true value, 1. This might seem that the data has been overfitted. However, these estimated noise standard deviations are subject to two sources of random errors. First, they are based on MCMC samples which may have an effective sample as small as 20 due to the stopping rule applied (see Subsection 3.2.5). Therefore, even when the posterior mean of δ is well above 1, the estimated δ may occassionally be smaller than 1 by chance. Second, the particular observed dataset may just be less variable than typical due to randomness. In such a case, a good model tends to have the estimated noise standard deviation less than the true value of δ. In particular, the posterior mean of δ by a GP method may be smaller than the true value of 1.

Overfitting by the GP methods, however, might also be possible. Recall that the exponential parts of the covariance functions used for the GP methods as in (2.50) and (2.66) are stationary, and therefore favor functions with the same properties over different regions in the covariate vector space. For a function with different degrees of wiggliness over different regions, GP methods based on such covariance functions may produce large estimated length-scales due to the smoother parts of the function, therefore underfitting the more wiggly parts; they may also produce small estimated length-scales due to the more wiggly parts of the function, therefore overfitting the less wiggly parts. More about the stationarity of the GP covariance functions is discussed in the last chapter.

4.4 Computing time by Gaussian process estimators

The computing time taken by each Gaussian process method for processing one dataset is determined by the time taken for each MCMC iteration and the number of MCMC iterations required. The time taken per iteration depends on the efficiency of the MCMC sampling schemes used (e.g. Elliptical slice sampling and univariate slice sampling). For the sampling schemes used, the time taken per iteration depends on the dimensionality d of the covariate vector and the length n of the latent vector(s). The dimensionality d of the covariate vector determines how many hyperparameters need to be updated and the length n of the latent vector(s) determines the time taken for computing the Cholesky decomposition of the covariance matrix of the latent vector(s), which is proportional to n^3. Computing the covariance matrix of the latent vector(s) takes time proportional to dn^2.

The number of MCMC iterations required depends on the required effective sample size from the

MCMC iterations and how fast the MCMC iterations mix (see Subsection 3.2.5). Given the required effective sample size of the MCMC iterations, the slower the MCMC iterations mix, the larger the autocorrelation times of the MCMC iterations are and the larger number of iterations are required. The speed at which the MCMC iterations mix is mainly attributable to the length n of the latent vector(s), when n is large compared to d. That is why the average number of iterations required by GP_E per dataset is much larger for the Kang and Schafer example (where the length of the latent vector is $n = 200$) than for the experiments considered in Section 4.2 (where the length of the concatenated latent vector is $2 \times n = 2 \times 20$ or 2×50). Similarly, the average numbers of iterations required by GP_I, GP_T and GP_R per dataset for the Kang and Schafer example are much less than those required for the experiments considered in Section 4.2, since for the Kang and Schafer example where y is real-valued, no latent vector needs to be updated by GP_I, GP_T and GP_R.

Take the Kang and Schafer example for illustration. The average times for 1000 iterations by GP_I and GP_R are around 10 minutes on a processor running at about 3GHz. The average times for 1000 iterations by GP_T and GP_E are about 45 minutes using the same processor and the same version of R programming language. The reasons why GP_T and GP_E take longer time per iteration are 1) about two times more hyperparameters need to be updated for GP_T and GP_E than for GP_I and GP_R, and 2) the covariance matrix of the latent vector(s) that needs to be updated for GP_T and GP_E is $2n \times 2n$ instead of $n \times n$ for GP_I and GP_R. The average numbers of iterations required per dataset by GP_I, GP_T and GP_R are about 1000-1500, while the average number of iterations per dataset required by GP_E is about 12000-15000. Therefore, the average total times taken per dataset by GP_I and GP_R are less than 15 minutes, the average total time taken per dataset by GP_T is about one hour or less, and the average total time taken per dataset by GP_E is about 10 hours.

The total time (10 hours) taken per dataset by GP_E may seem unacceptable in certain practical situations. However, there is large space for improvement in the time taken by GP_E. The current computing program written for this thesis is not optimal. First, there may be redundancy in computing or updating things that are not needed. Second, in terms of computation, the dimensionality of the covariance matrix of the latent vector(s) can be reduced from $2n \times 2n$ to $(n_{eff} + n) \times (n_{eff} + n)$ in the case of GP_T and GP_E, or from $n \times n$ to $n_{eff} \times n_{eff}$ in the case of GP_I and GP_R, where n_{eff} is the effective sample size of the observed dataset. These two types of improvement will help reduce the computing time taken per iteration by all these estimators. The dominating factor that influences the time taken per dataset by GP_E compared to GP_T is the number of iterations required, which is determined by the efficiency of sampling the latent vector. In the current computing program, the latent vector is updated 5 times in between updating each hyperparameter. If instead, the latent vector is updated more times (e.g. 10 or 20) in between updating hyperparameters, the latent vector will be sampled more efficiently over iterations with reduced autocorrelation time. Note that updating the latent vector takes a minor amount of time compared to updating the hyperparameters. That is, increasing the number of times updating the latent vector per iteration will not substantially increase the total time taken per iteration, but may largely reduce the number of iterations required per dataset. Therefore, the time taken per dataset by GP_E can conceivably be reduced greatly, for example, from 10 hours to a few hours or even less.

Chapter 5

Conclusion

Both simulation studies and the analysis of the Kang and Schafer example in this thesis show that the Gaussian process approaches that use the selection probability are able to not only correct selection bias effectively, but also control the sampling errors well, and therefore can often provide more efficient estimates than the methods compared that are not based on Gaussian process models, in both simple and complex situations. Even the Gaussian process approach that ignores the selection probability often, though not always, performs well when some selection bias is present.

Particularly, a method like the Horvitz-Thompson estimator that totally ignores the covariates can be very inefficient, even if it is unbiased or consistent. A method like the Gaussian process estimator $\widehat{\phi}_{GP_I}$ that has a highly flexible model for the response, but does not employ the selection probability explicitly, may still do reasonably well when selection bias is not strong. When the response function depends on the selection probability in a complex manner, a method like $\widehat{\phi}_{GP_I}$ that has a flexible model for the response, but does not exploit the selection probability more explicitly, may no longer do well with a limited sample size. Therefore, it is best to have a method which not only corrects selection bias explicitly but also exploits the covariate information to a maximum extent without overfitting the data. The Gaussian process estimators $\widehat{\phi}_{GP_T}$, $\widehat{\phi}_{GP_E}$ and $\widehat{\phi}_{GP_R}$ not only have a flexible model for the response, but also employ the selection probability in a rather flexible manner, unlike many popular approaches in the literature that employ the selection probability only in the form of its inverse. Methods like $\widehat{\phi}_{GP_T}$, $\widehat{\phi}_{GP_E}$ and $\widehat{\phi}_{GP_R}$, although not perfect, can conceivably deal with a large number of complex problems sufficiently well, when selection bias is an issue.

In addition to demonstrating the strength of the Gaussian process methods considered, this thesis shows that these methods can be implemented efficiently enough to realize their benefits in practice. They, therefore, should be brought to broader attention, and help promote use of Bayesian hierarchical models in general for dealing with selection bias in complex situations.

The Gaussian process approaches considered in this thesis are, however, not without flaws. Particularly, the exponential parts of the Gaussian process covariance functions used in this thesis as in (2.46), (2.47), (2.50) or (2.66) are stationary, and therefore favor functions with the same degrees of smoothness over different subregions of the covariate vector space. With such covariance functions, functions that have different properties in different regions of the covariate vector space only have very small probabilities under the corresponding Gaussian process priors. Therefore, with finite sample sizes, models based on these Gaussian process priors may not do well for estimating such functions. Developing approaches to correcting selection bias using Gaussian process models with non-stationary covariance functions or using other Bayesian hierarchical models would be interesting for future research.

In addition, the strategy given by (2.59) only allows positive correlations between the latent functions g_μ and g_ν. When the correlation between $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ with respect to $\mathbf{x}$ is known to be negative as for the Kang and Schafer example, the response variable can be reversed so that this strategy still works fine. However, in practice, it is often not known if the correlation between $\mu(\mathbf{x})$ and $\nu(\mathbf{x})$ (w.r.t $\mathbf{x}$) is positive or negative. The strategy by (2.59) can be generalized such that the correlation between the latent functions g_μ and g_ν is adjustable. One type of generalisation is as follows. Consider

$$g_\mu = \alpha g_0 + g_1 \text{ and } g_\nu = \beta g_0 + g_2 \tag{5.1}$$

where g_1, g_2, g_0 are the same as in (2.60) and α and β are hyperparameters which have priors over $(-\infty, \infty)$. Then for any $\mathbf{x}_1, \ldots, \mathbf{x}_n$, the covariance matrix for $\begin{pmatrix} \mathbf{g}_\mu^{(n)} \\ \mathbf{g}_\nu^{(n)} \end{pmatrix}$ where

$$\mathbf{g}_\mu^{(n)} = \begin{pmatrix} g_\mu(\mathbf{x}_1) \\ g_\mu(\mathbf{x}_2) \\ \vdots \\ g_\mu(\mathbf{x}_n) \end{pmatrix} \quad \text{and} \quad \mathbf{g}_\nu^{(n)} = \begin{pmatrix} g_\nu(\mathbf{x}_1) \\ g_\nu(\mathbf{x}_2) \\ \vdots \\ g_\nu(\mathbf{x}_n) \end{pmatrix}, \tag{5.2}$$

is

$$\mathrm{Cov}\begin{pmatrix} \mathbf{g}_\mu^{(n)} \\ \mathbf{g}_\nu^{(n)} \end{pmatrix} = \begin{pmatrix} K_1 + \alpha^2 K_0 & \alpha\beta K_0 \\ \alpha\beta K_0 & K_2 + \beta^2 K_0 \end{pmatrix} \tag{5.3}$$

where K_h, $i = 1, 2, 0$, are the same as in (2.63). With α and β adjustable over $(-\infty, \infty)$, the correlation between $\mathbf{g}_\mu^{(n)}$ and $\mathbf{g}_\nu^{(n)}$ is also adjustable over $(-1, 1)$.

Despite the extensive experimental studies in this thesis, certain aspects of the Gaussian process estimators studied remain unclear. In particular, as indicated by the Kang and Schafer example in Section 4.3, the type of estimators that use the selection probability as a covariate (i.e. $\widehat{\phi}_{GP_R}$, $\widehat{\phi}_{GP_{R2}}$ and $\widehat{\phi}_{GP_{R3}}$) seem more robust against how the selection probability function has been transformed when used as a covariate, compared to the type of estimators (i.e. $\widehat{\phi}_{GP_T}$ and $\widehat{\phi}_{GP_{Tb}}$) that assign a joint prior for g_μ and g_ν. Further studies may determine whether it is generally true that the $\widehat{\phi}_{GP_R}$ estimator is more robust than the $\widehat{\phi}_{GP_T}$ estimator against the transformation of the selection probability.

If the $\widehat{\phi}_{GP_R}$ estimator is indeed more robust than the $\widehat{\phi}_{GP_T}$ estimator, then we can expect that a modified version of $\widehat{\phi}_{GP_R}$ that uses estimated selection probabilities would also be more robust than $\widehat{\phi}_{GP_E}$ (the version of $\widehat{\phi}_{GP_T}$ that uses the estimated ν). Particularly, we may consider estimating the selection probability using Gaussian process models. To do so, two strategies could be considered. First, the selection probability could be estimated, and then used as a fixed covariate over MCMC iterations for modeling the response. Second, we could model the response and the selection probability simultaneously. That is, the selection probability is sampled at each MCMC iteration. In this case, the hyperparameters for the response and the hyperparameters for the selection probability can either be independent or share common priors.

Appendix A

Figures

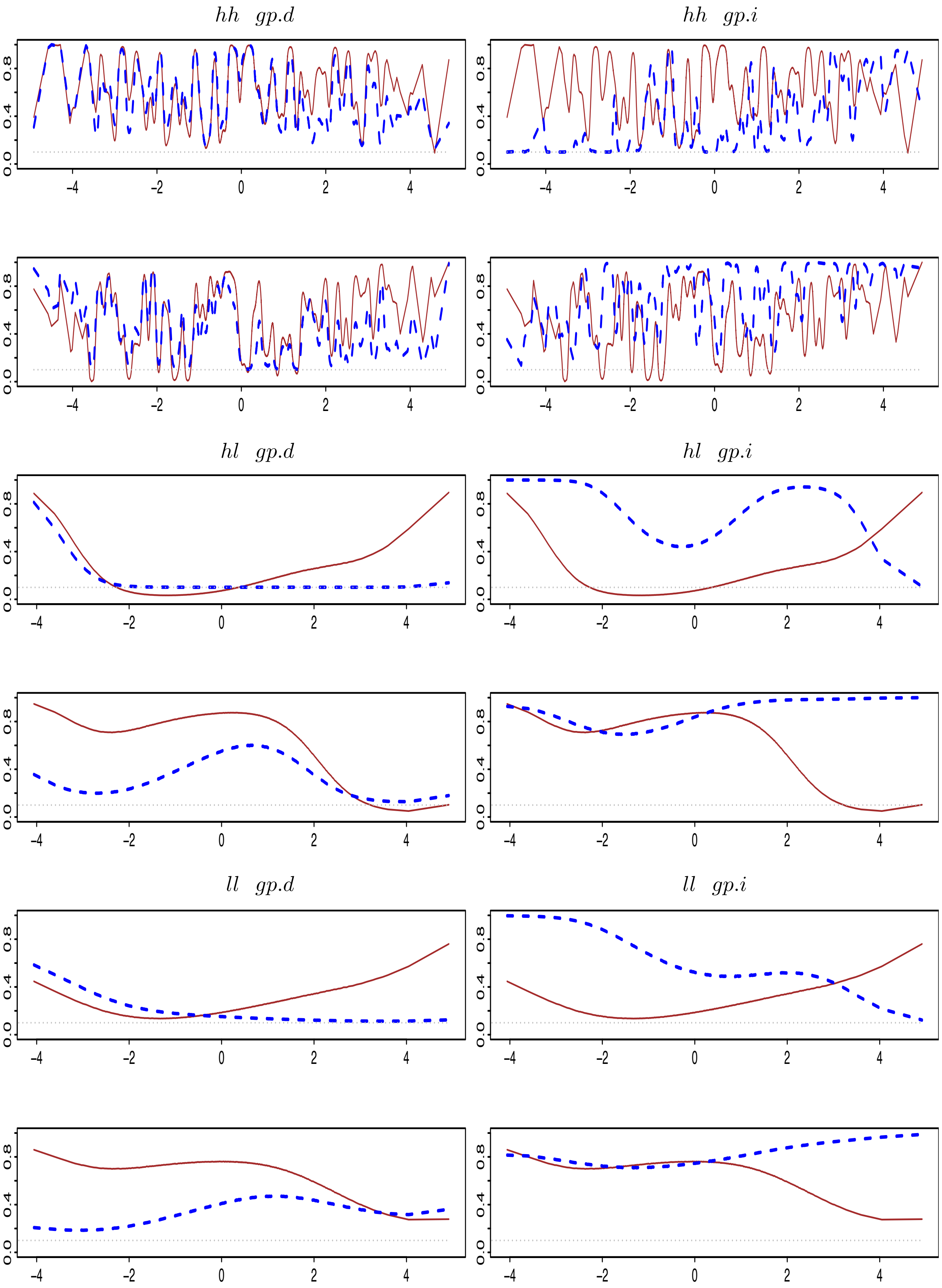


Figure A.1: Two sample pairs of functions (red solid: μ and blue dash: ν) in one-dimensional space ($d = 1$) under $\{hh,\ gp.d\}$, $\{hh,\ gp.i\}$, $\{hl,\ gp.d\}$, $\{hl,\ gp.i\}$, $\{ll,\ gp.d\}$, and $\{ll,\ gp.i\}$, respectively.

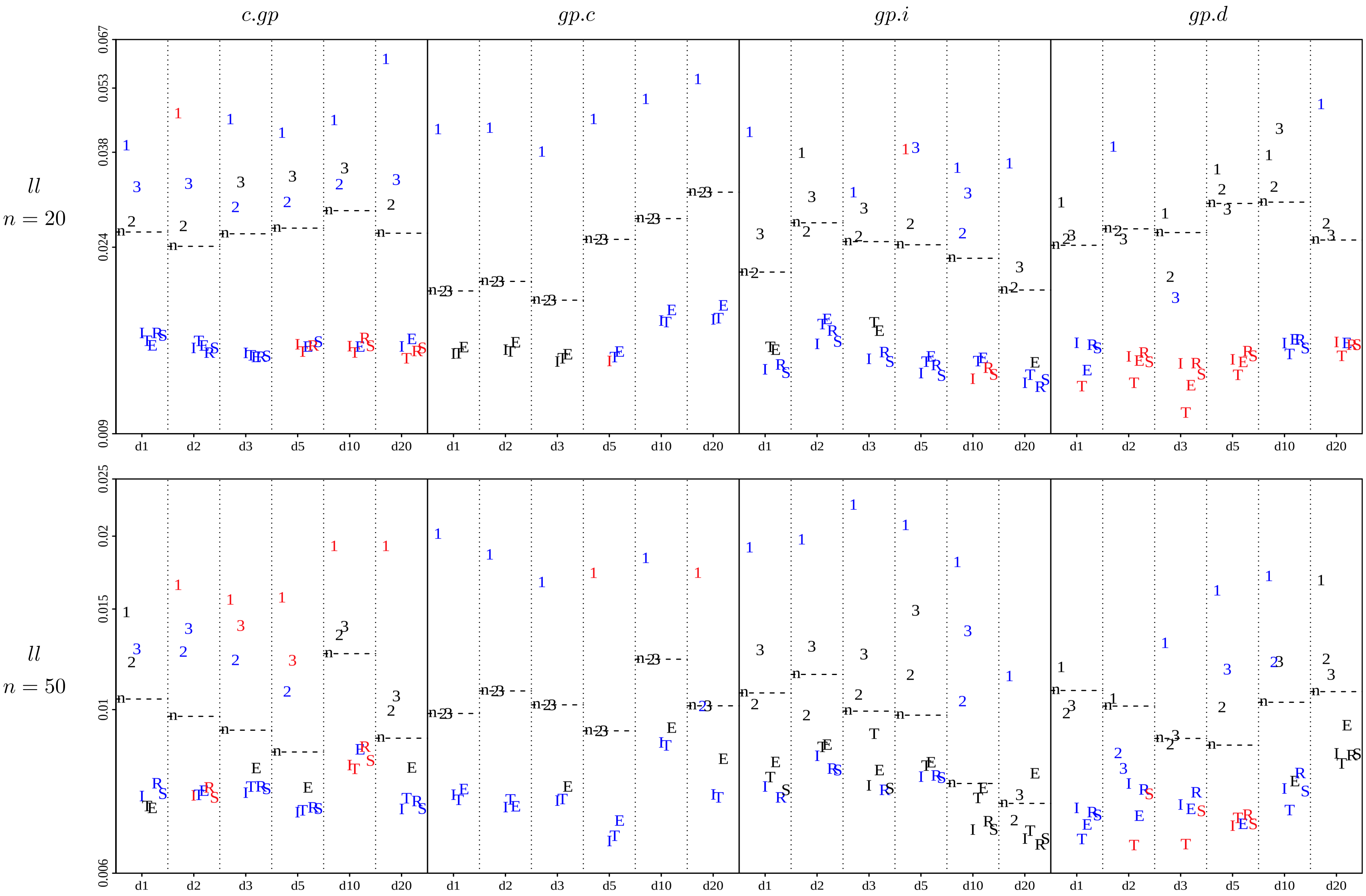


Figure A.2: Low correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{naive}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: 0.01 ≤ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

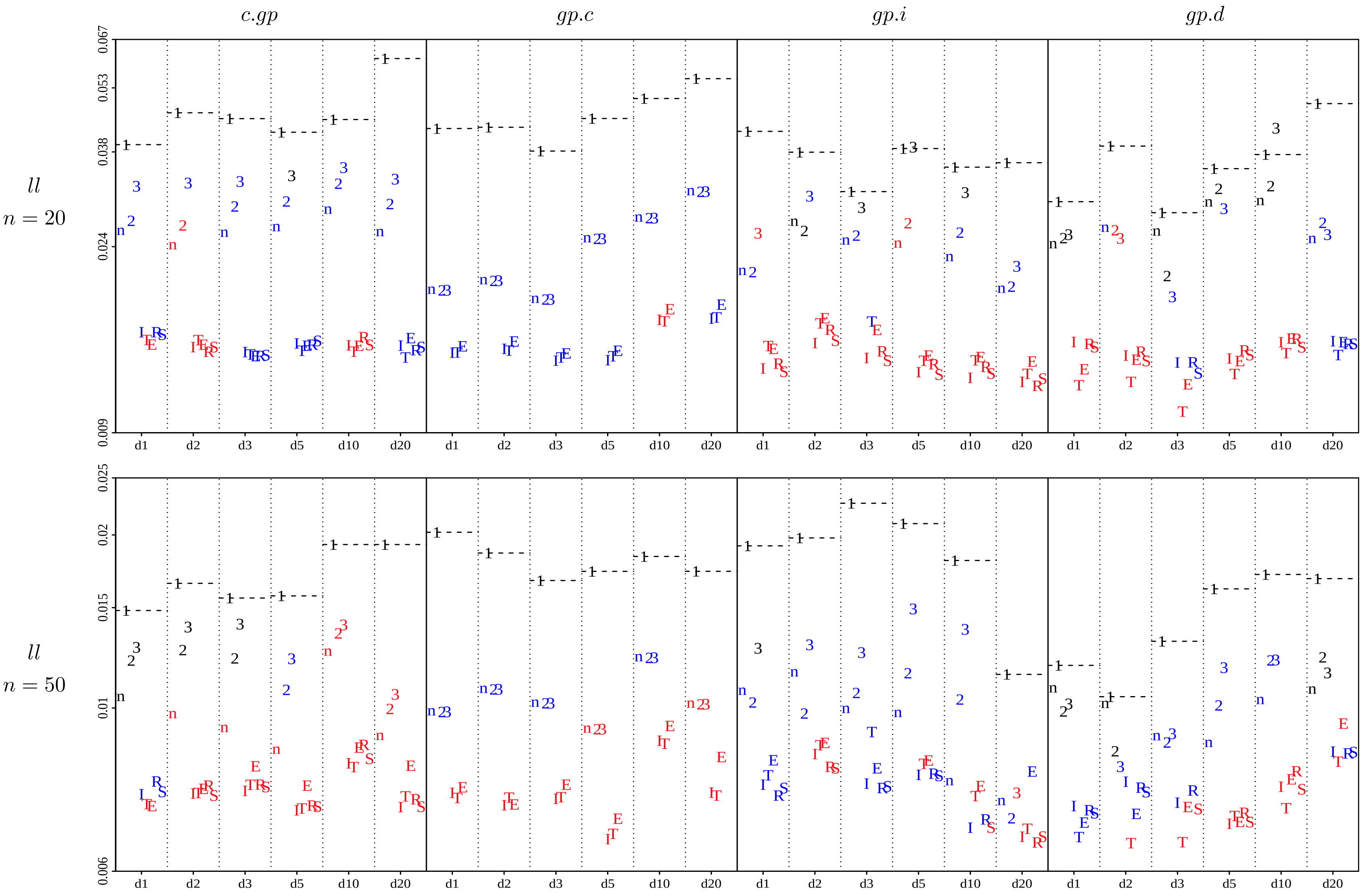


Figure A.3: Low correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{HT_1}$ (dashed). ‘n’: $\widehat{\phi}_{naive}$; ‘1’: $\widehat{\phi}_{HT_1}$; ‘2’: $\widehat{\phi}_{HT_2}$; ‘3’: $\widehat{\phi}_{HT_3}$; ‘I’: $\widehat{\phi}_{GP_I}$; ‘T’:, $\widehat{\phi}_{GP_T}$; ‘E’: $\widehat{\phi}_{GP_E}$; ‘R’: $\widehat{\phi}_{GP_R}$; ‘S’: $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

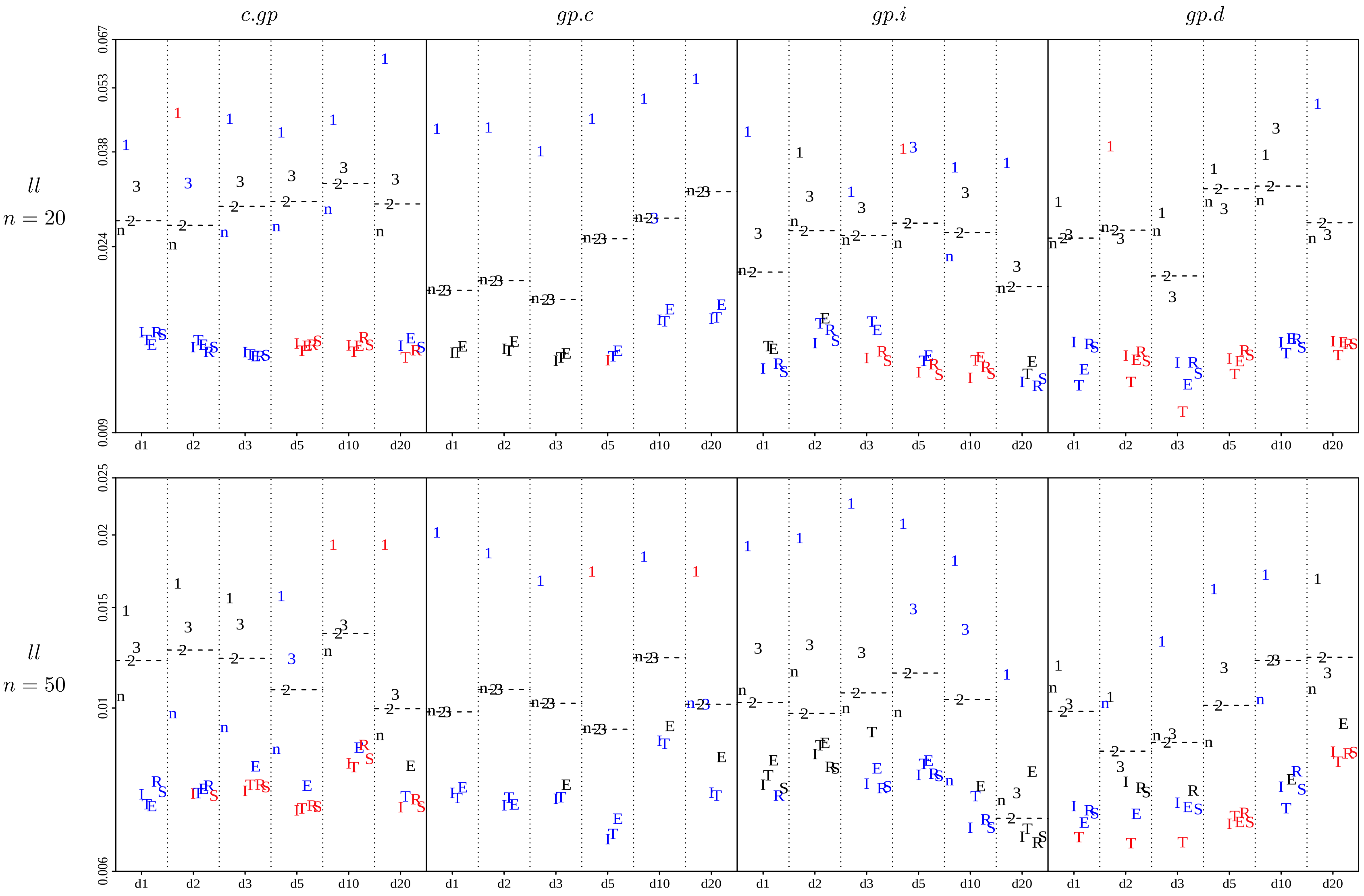


Figure A.4: Low correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{HT_2}$ (dashed). ‘n’: $\widehat{\phi}_{naive}$; ‘1’: $\widehat{\phi}_{HT_1}$; ‘2’: $\widehat{\phi}_{HT_2}$; ‘3’: $\widehat{\phi}_{HT_3}$; ‘I’: $\widehat{\phi}_{GP_I}$; ‘T’:, $\widehat{\phi}_{GP_T}$; ‘E’: $\widehat{\phi}_{GP_E}$; ‘R’: $\widehat{\phi}_{GP_R}$; ‘S’: $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

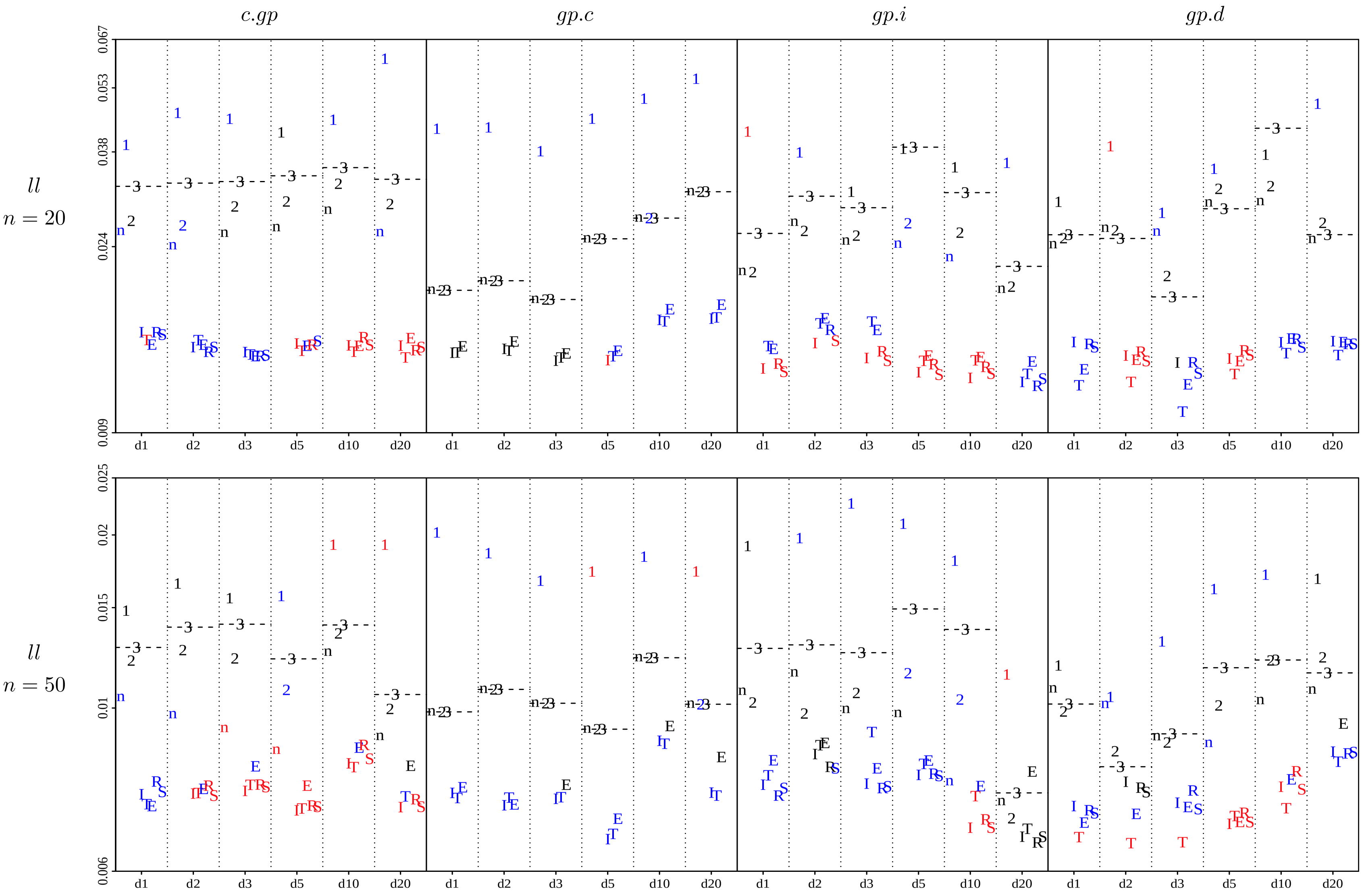


Figure A.5: Low correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{HT_3}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

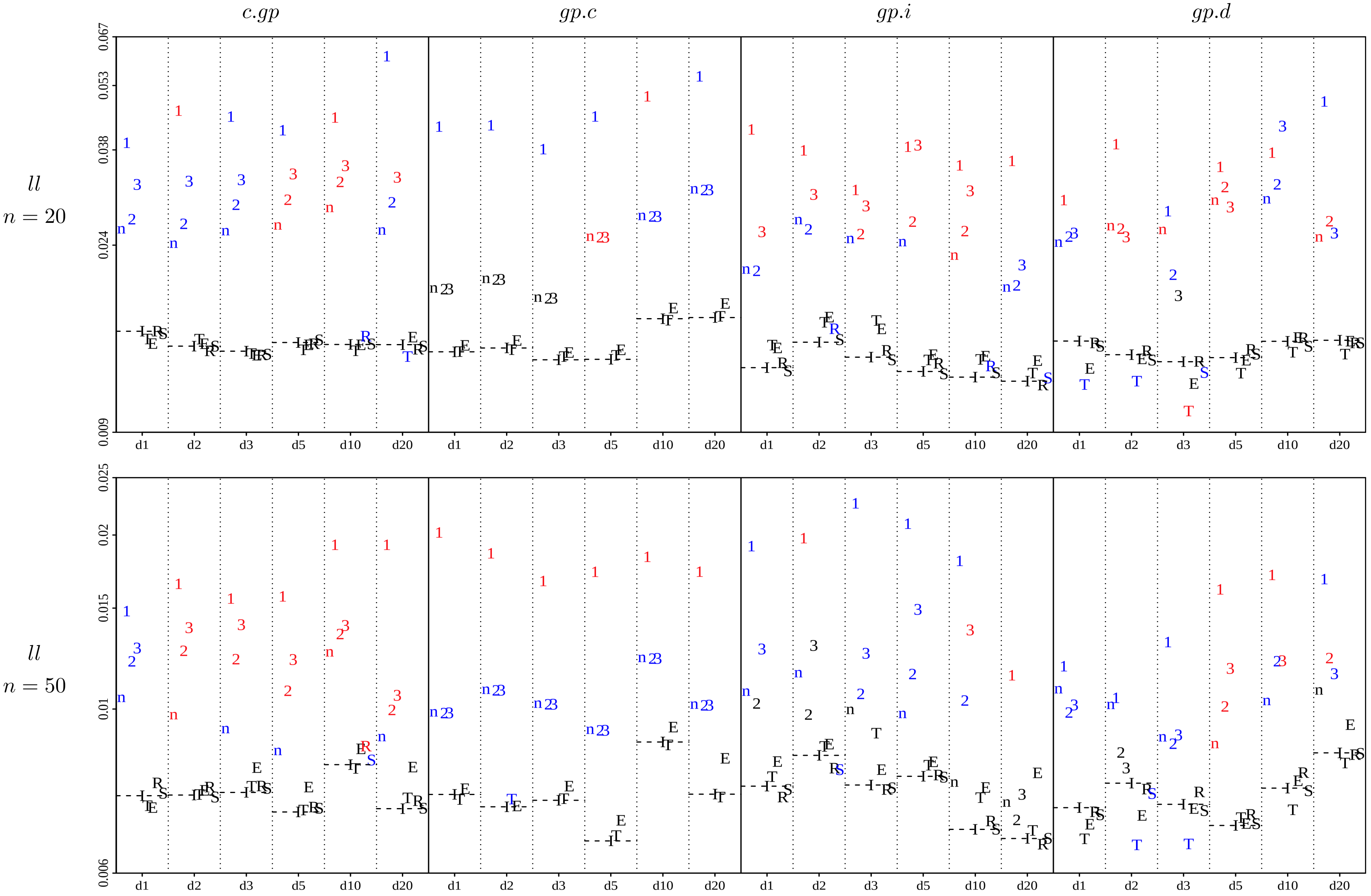


Figure A.6: Low correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_I}$ (dashed). ‘n’: $\widehat{\phi}_{naive}$; ‘1’: $\widehat{\phi}_{HT_1}$; ‘2’: $\widehat{\phi}_{HT_2}$; ‘3’: $\widehat{\phi}_{HT_3}$; ‘I’: $\widehat{\phi}_{GP_I}$; ‘T’:, $\widehat{\phi}_{GP_T}$; ‘E’: $\widehat{\phi}_{GP_E}$; ‘R’: $\widehat{\phi}_{GP_R}$; ‘S’: $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

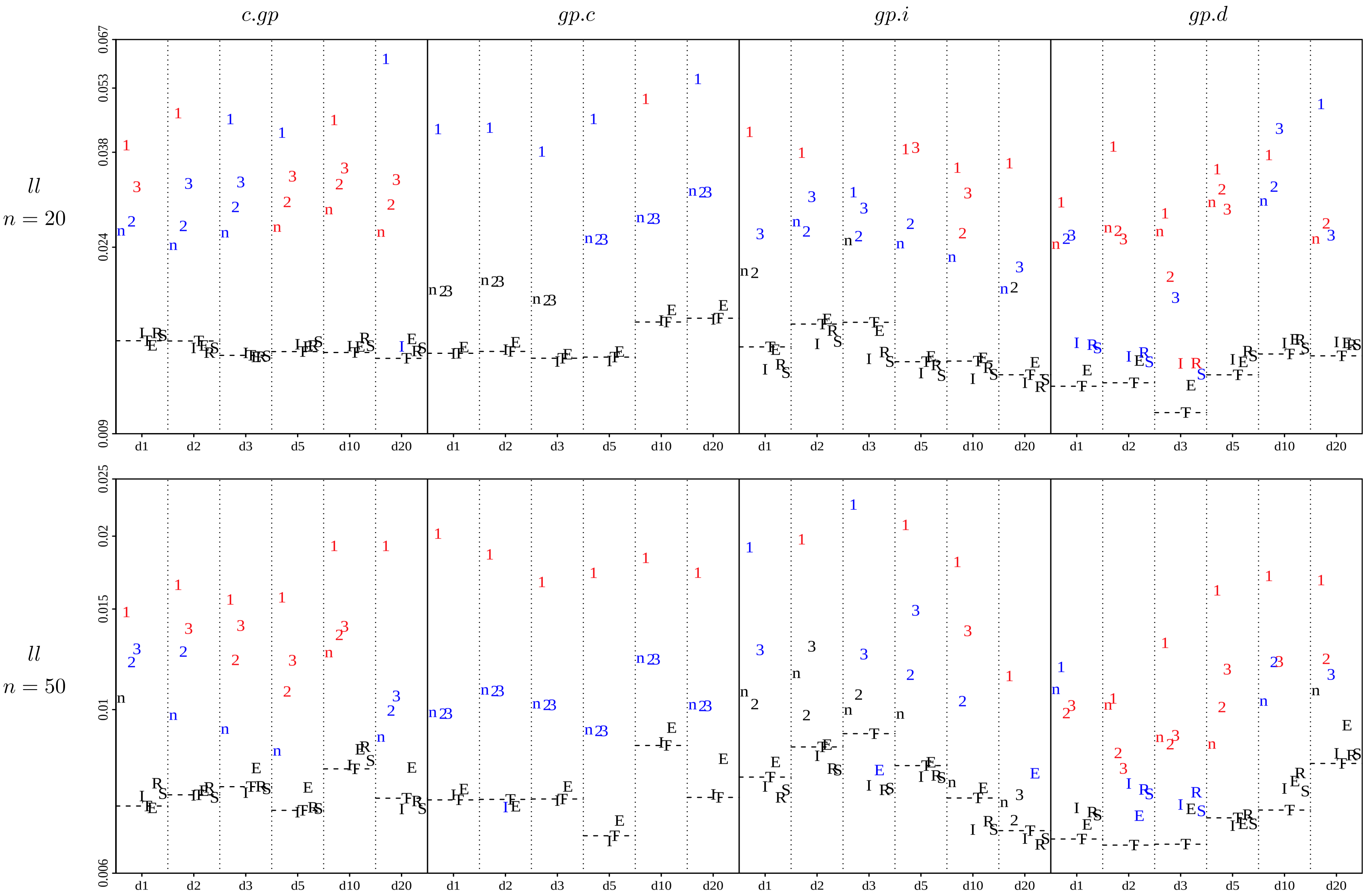


Figure A.7: Low correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_T}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

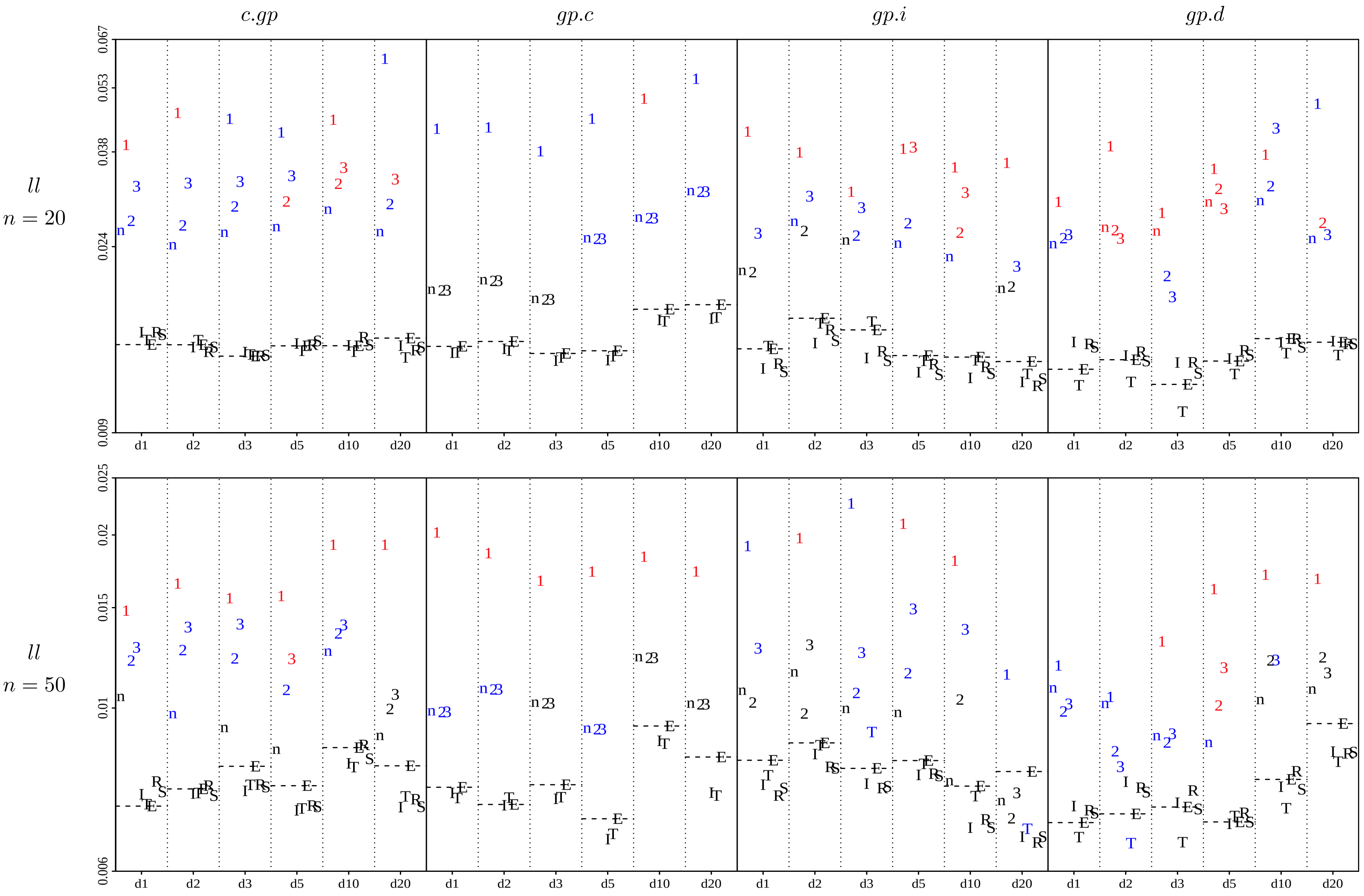


Figure A.8: Low correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_E}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

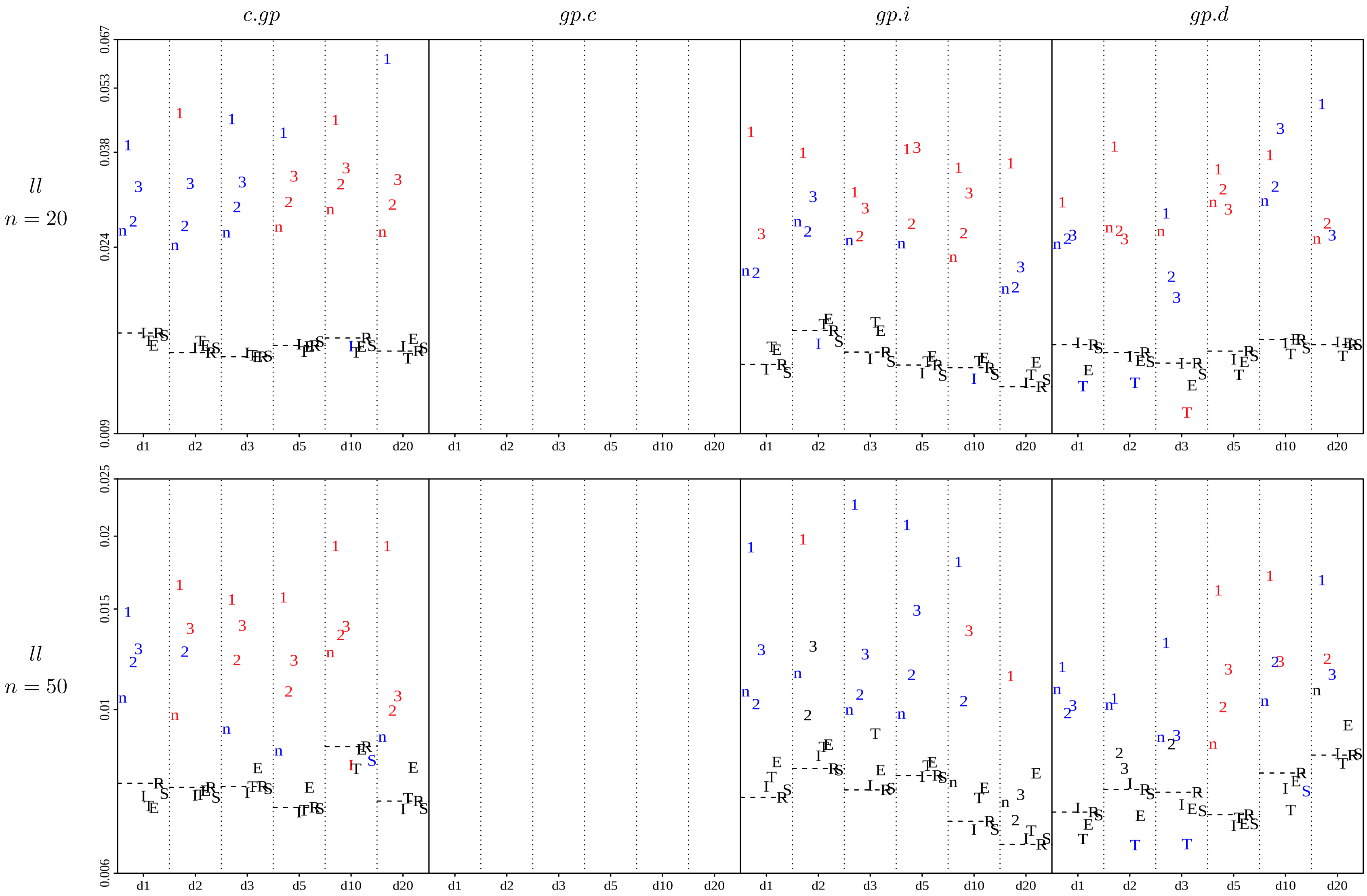


Figure A.9: Low correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_R}$ (dashed). ‘n’: $\widehat{\phi}_{naive}$; ‘1’: $\widehat{\phi}_{HT_1}$; ‘2’: $\widehat{\phi}_{HT_2}$; ‘3’: $\widehat{\phi}_{HT_3}$; ‘I’: $\widehat{\phi}_{GP_I}$; ‘T’:, $\widehat{\phi}_{GP_T}$; ‘E’: $\widehat{\phi}_{GP_E}$; ‘R’: $\widehat{\phi}_{GP_R}$; ‘S’: $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

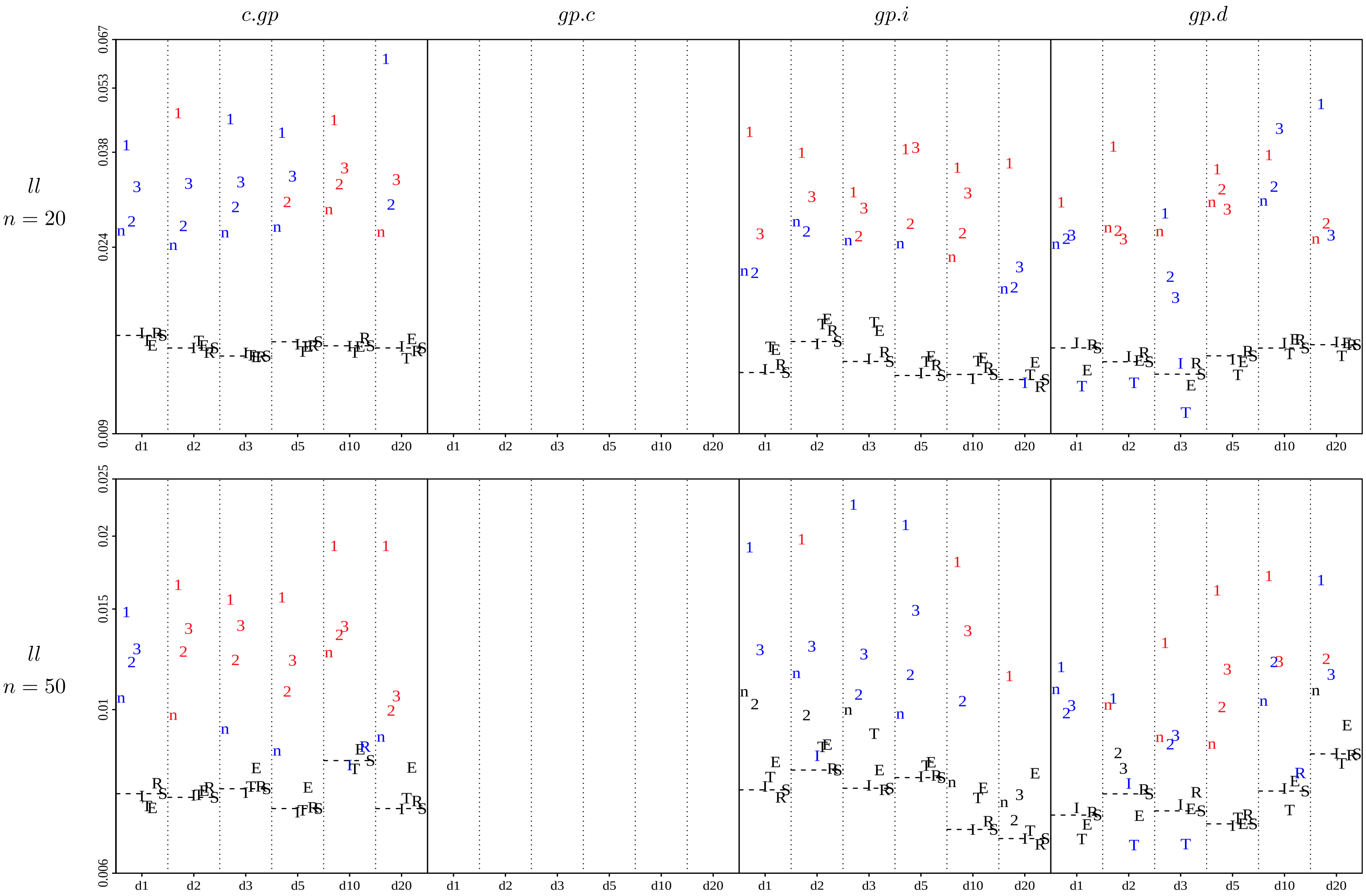


Figure A.10: Low correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_S}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

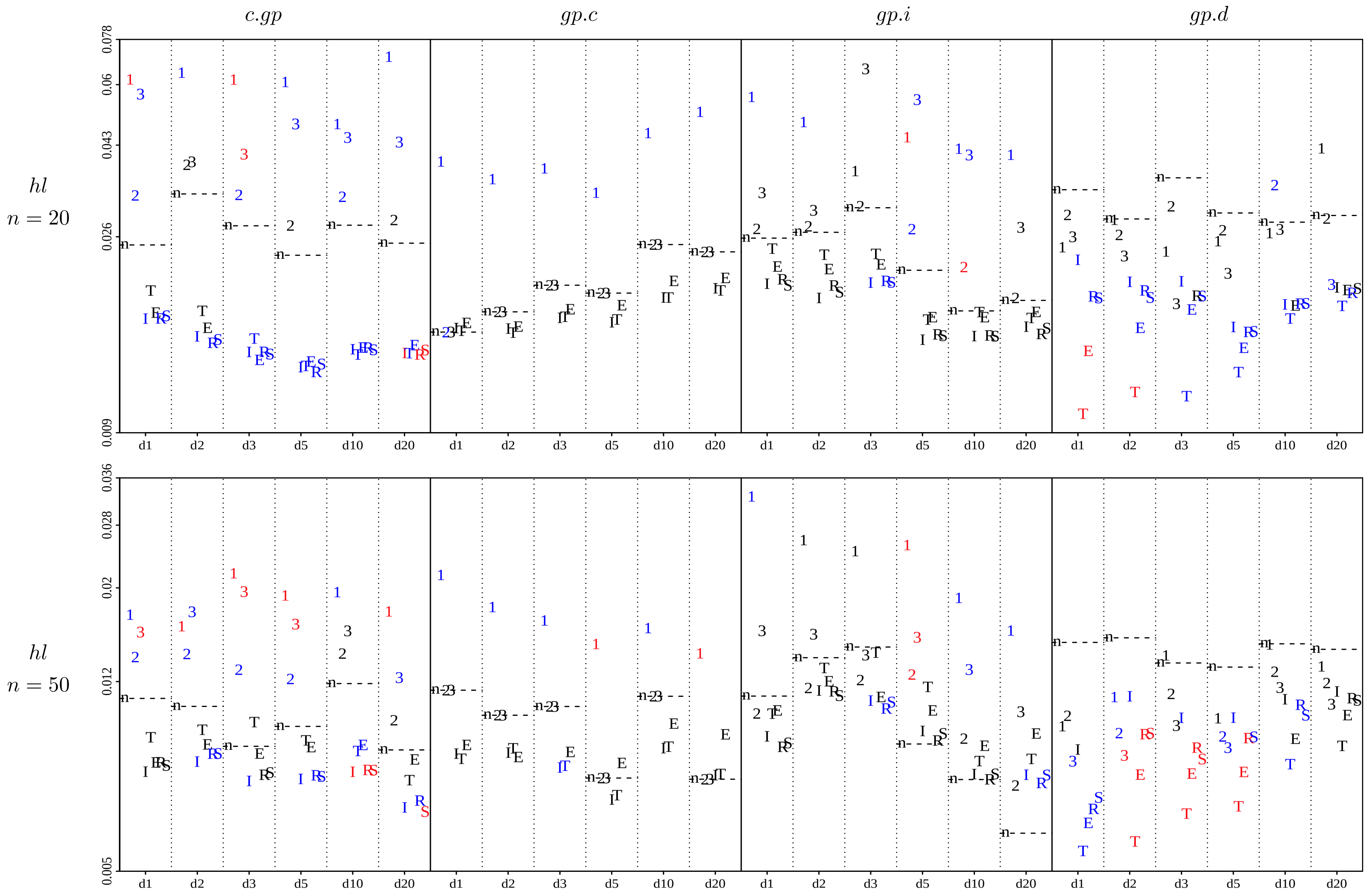


Figure A.11: High correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{naive}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

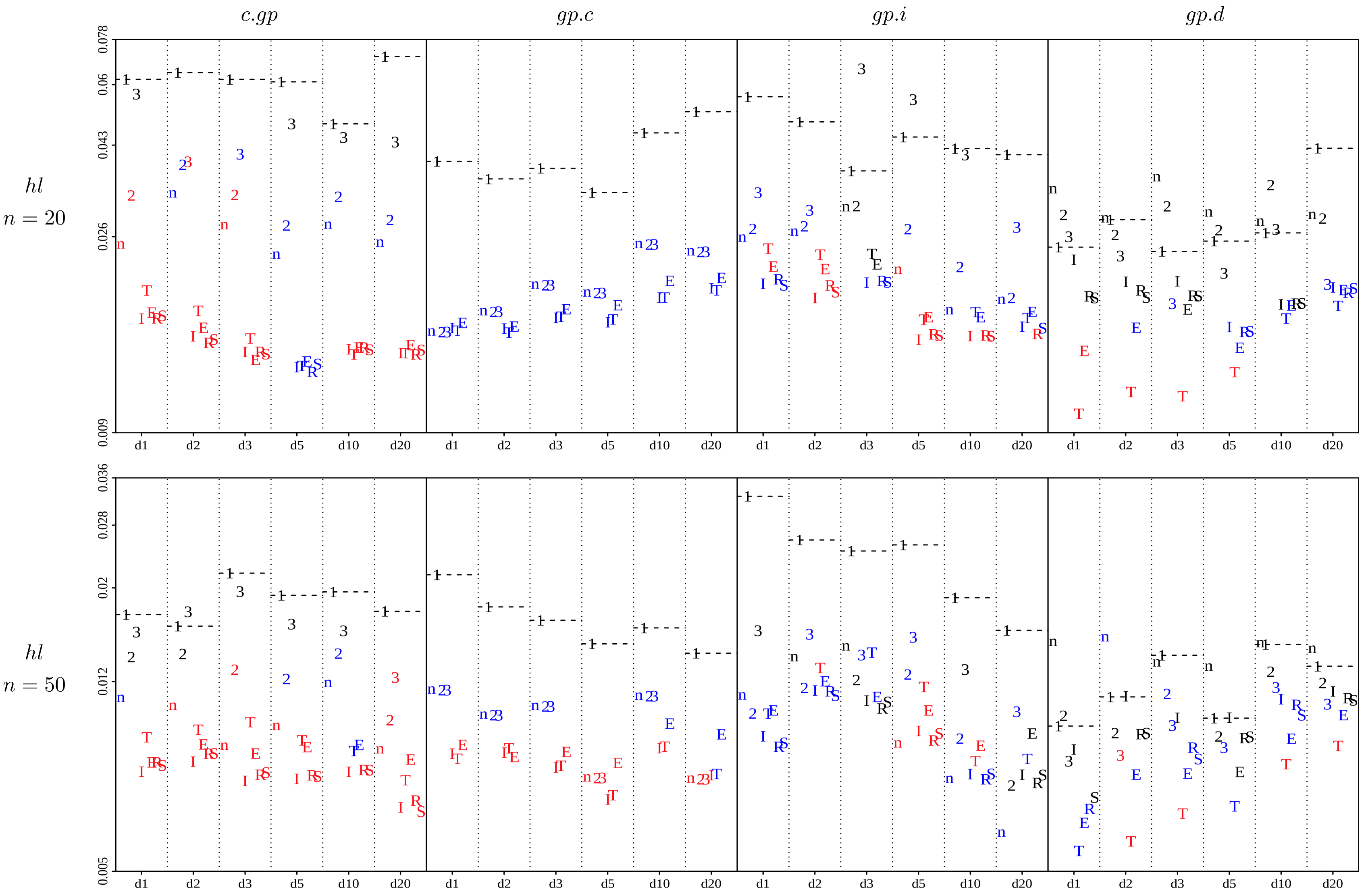


Figure A.12: High correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{HT_1}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

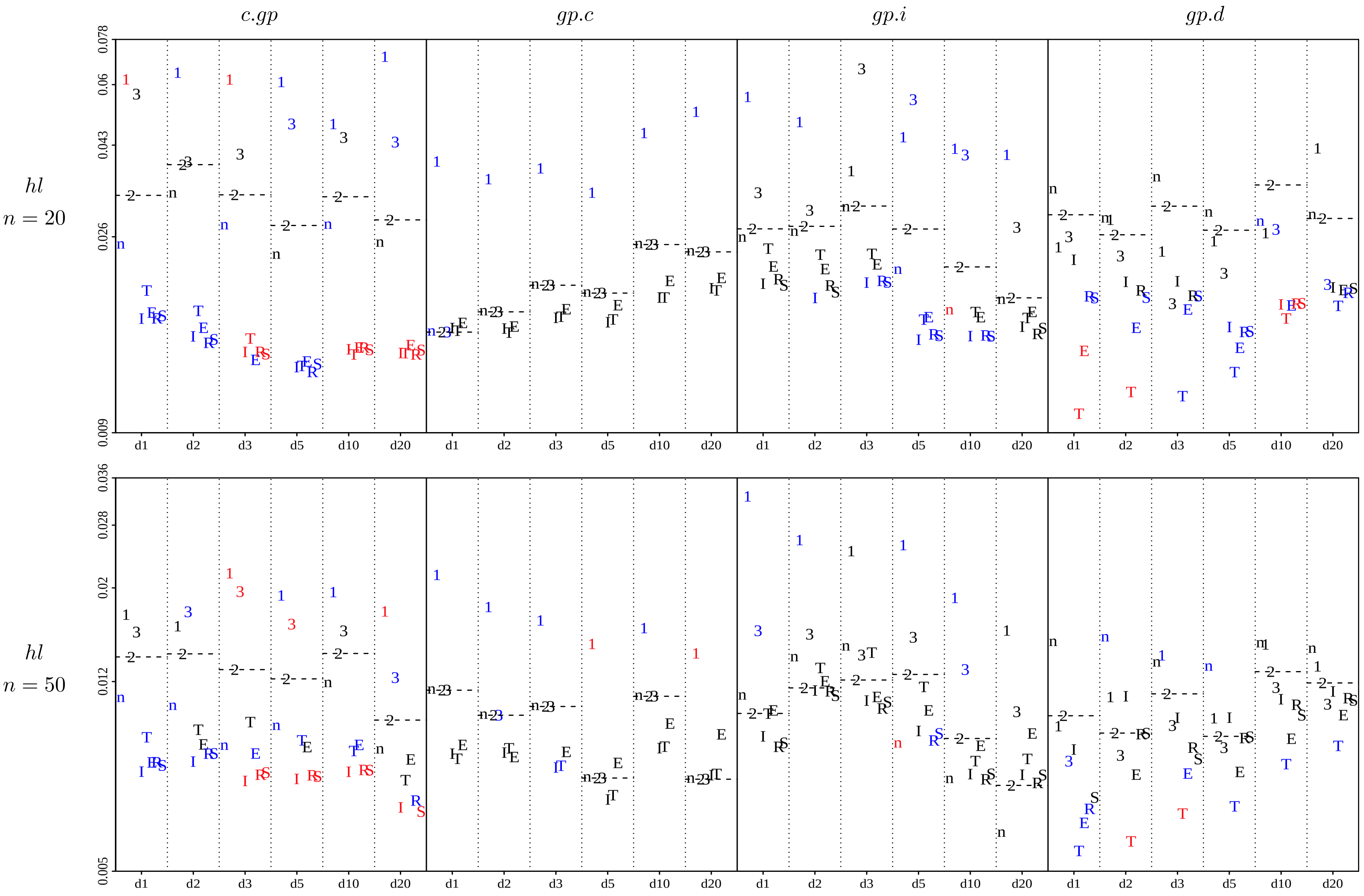


Figure A.13: High correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{HT_2}$ (dashed). ‘n’: $\widehat{\phi}_{naive}$; ‘1’: $\widehat{\phi}_{HT_1}$; ‘2’: $\widehat{\phi}_{HT_2}$; ‘3’: $\widehat{\phi}_{HT_3}$; ‘I’: $\widehat{\phi}_{GP_I}$; ‘T’:, $\widehat{\phi}_{GP_T}$; ‘E’: $\widehat{\phi}_{GP_E}$; ‘R’: $\widehat{\phi}_{GP_R}$; ‘S’: $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

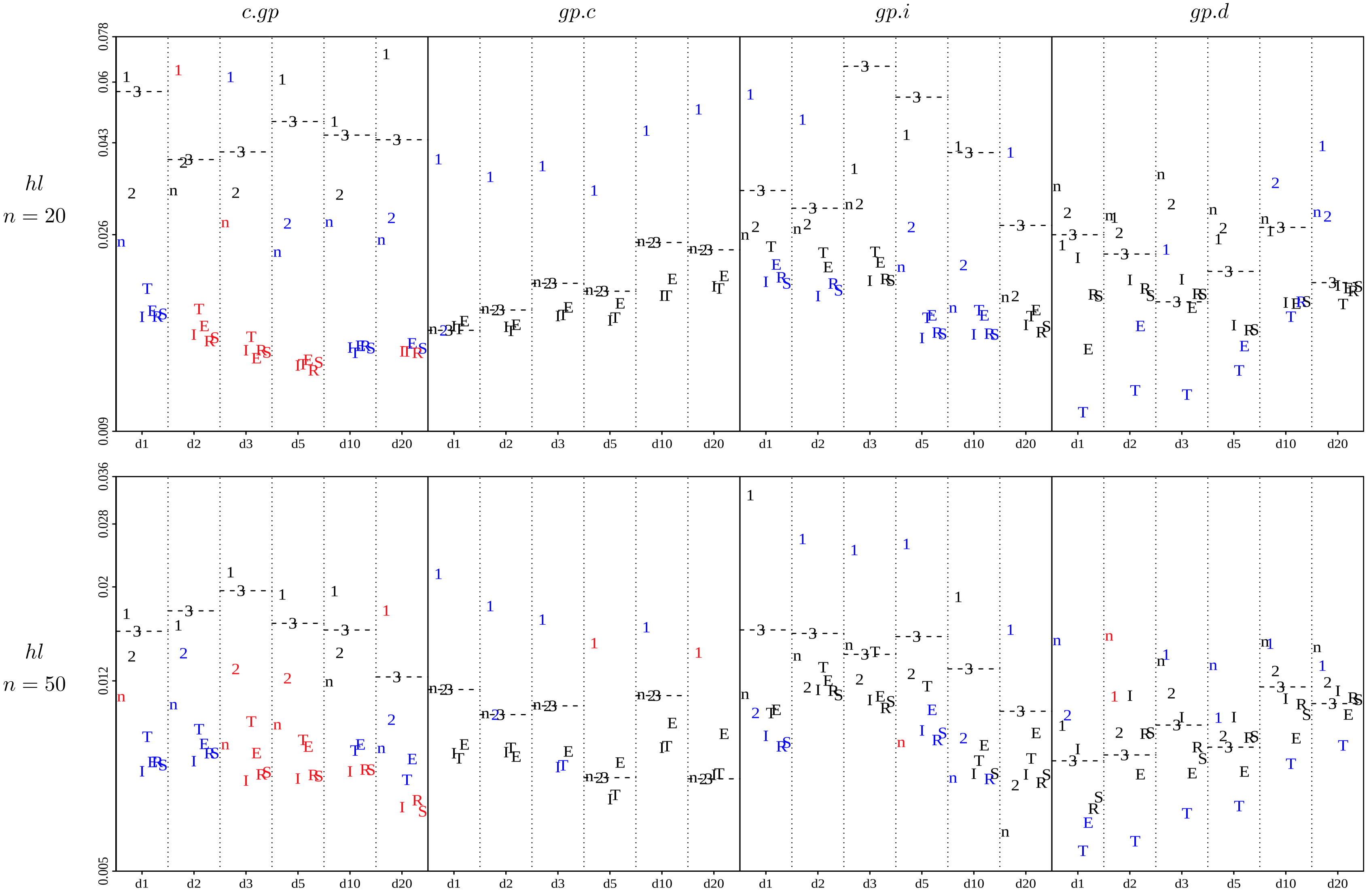


Figure A.14: High correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{HT_3}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

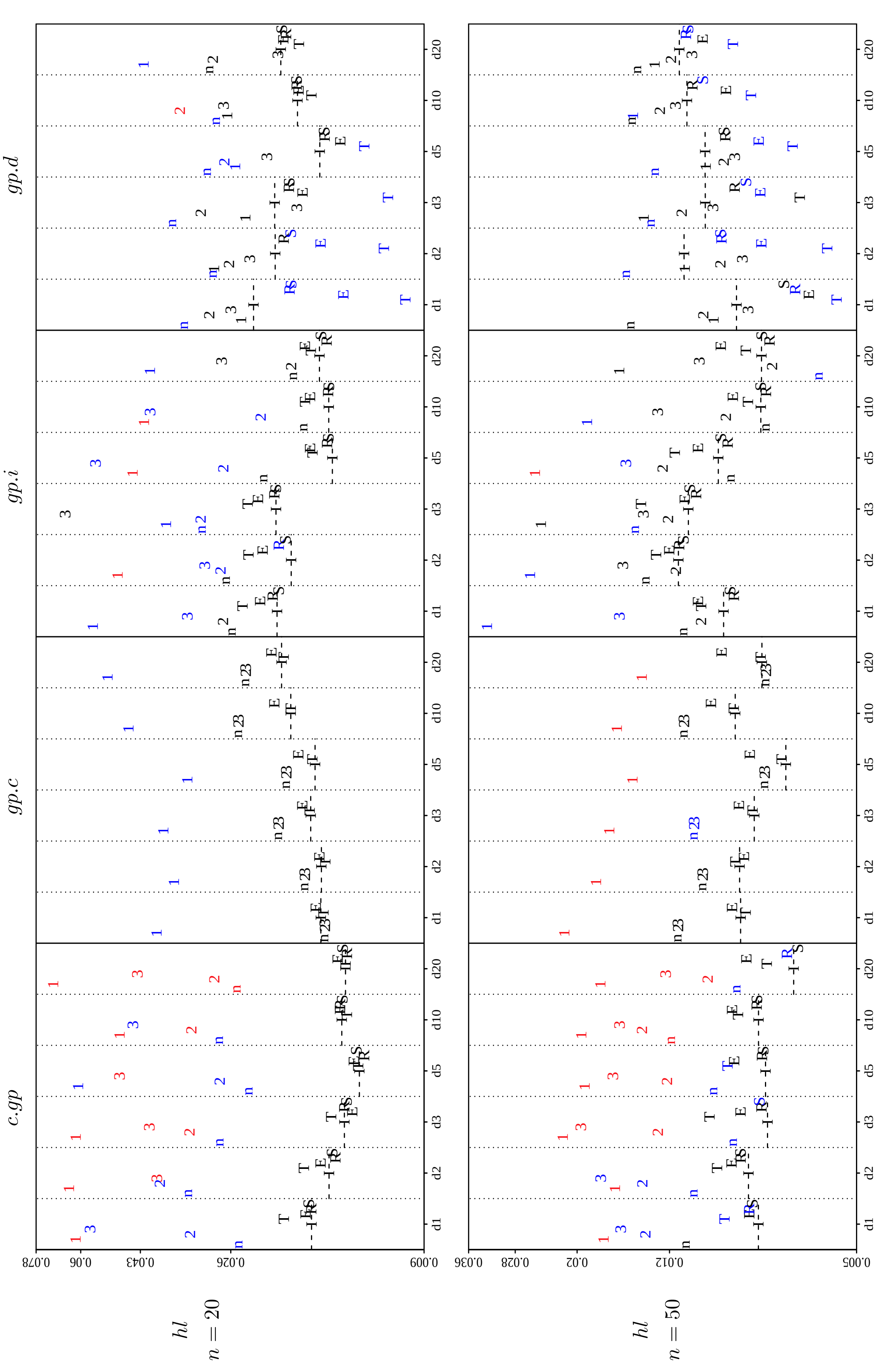


Figure A.15: High correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_I}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

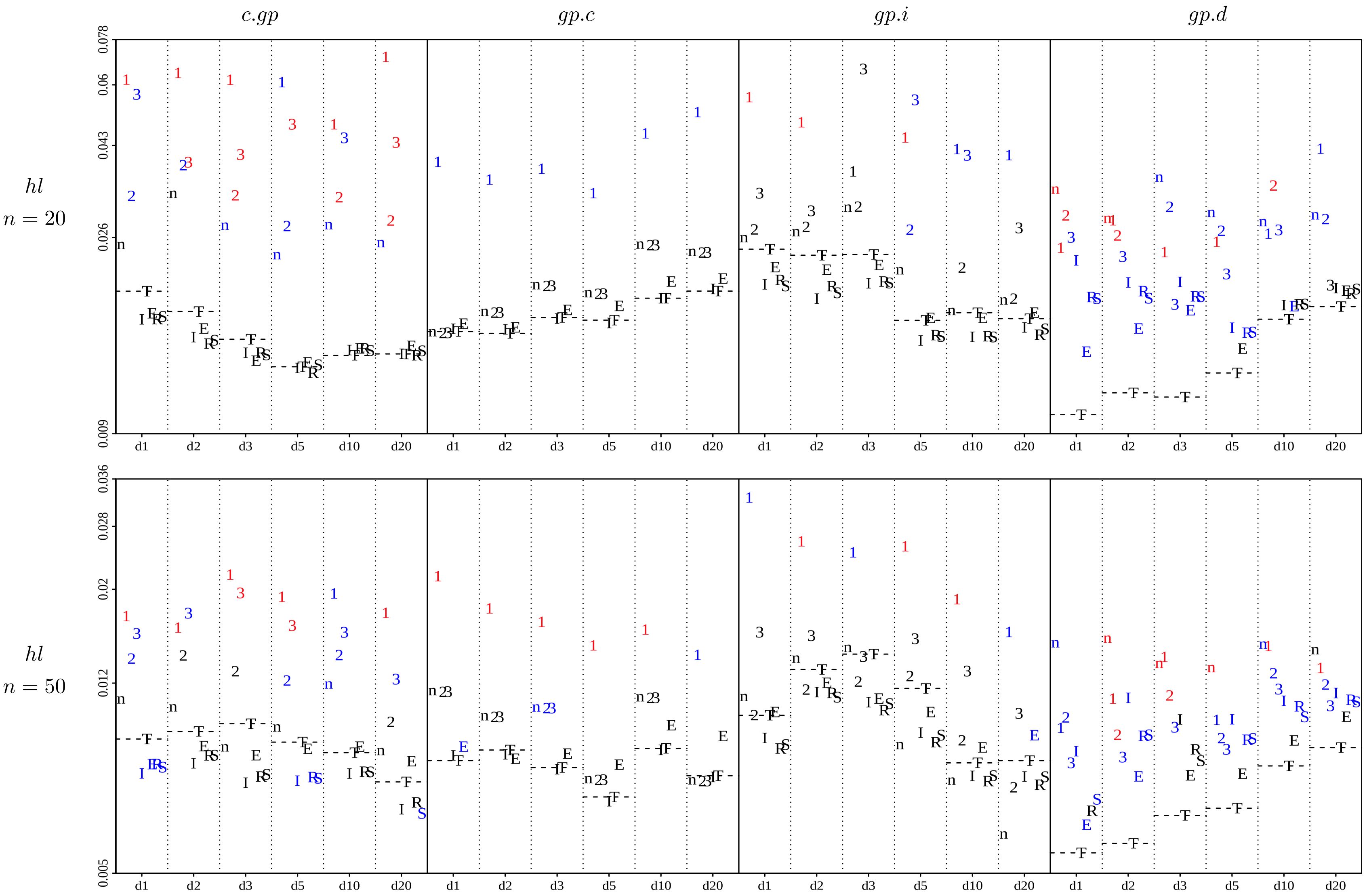


Figure A.16: High correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_T}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

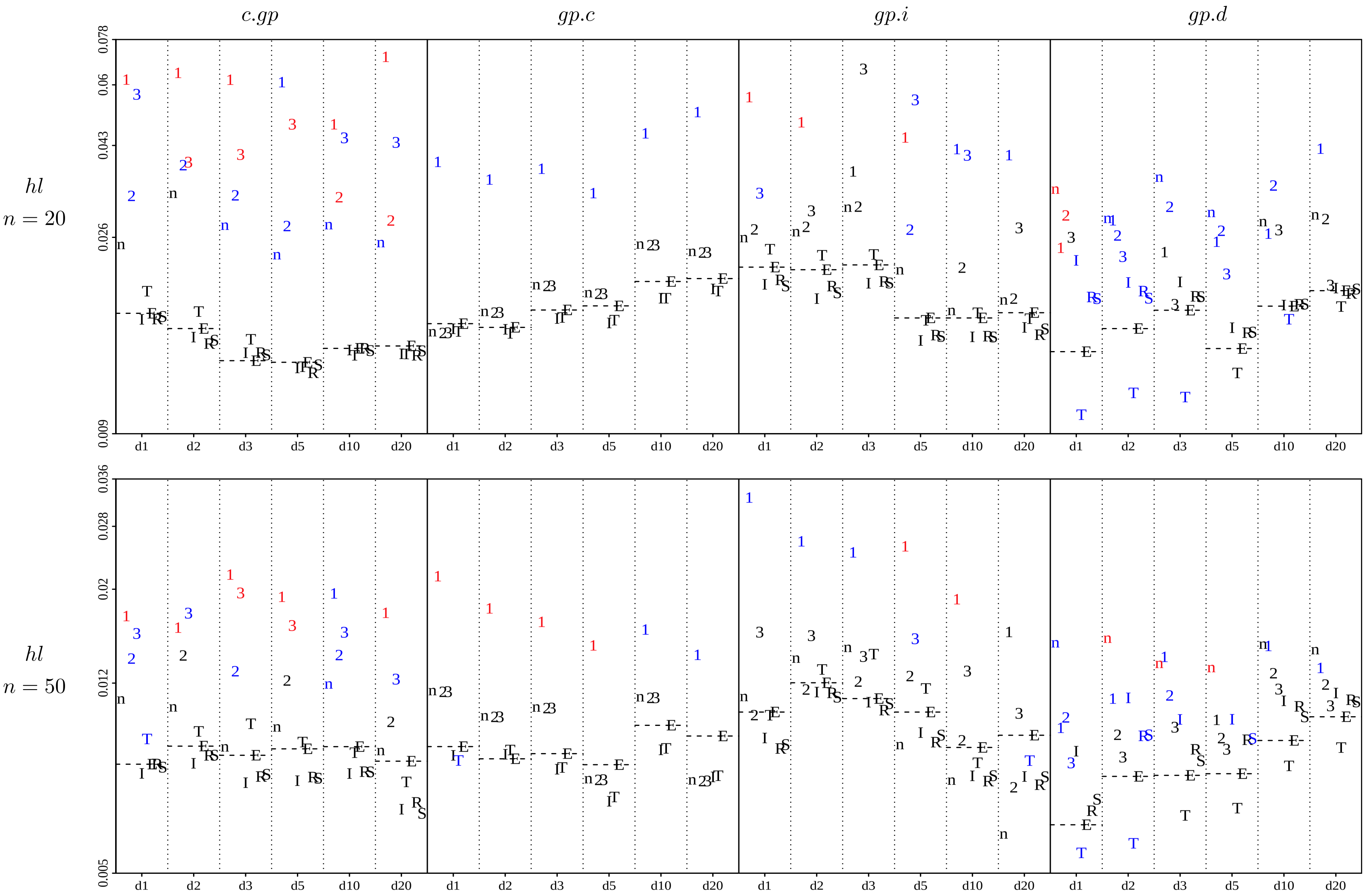


Figure A.17: High correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_E}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

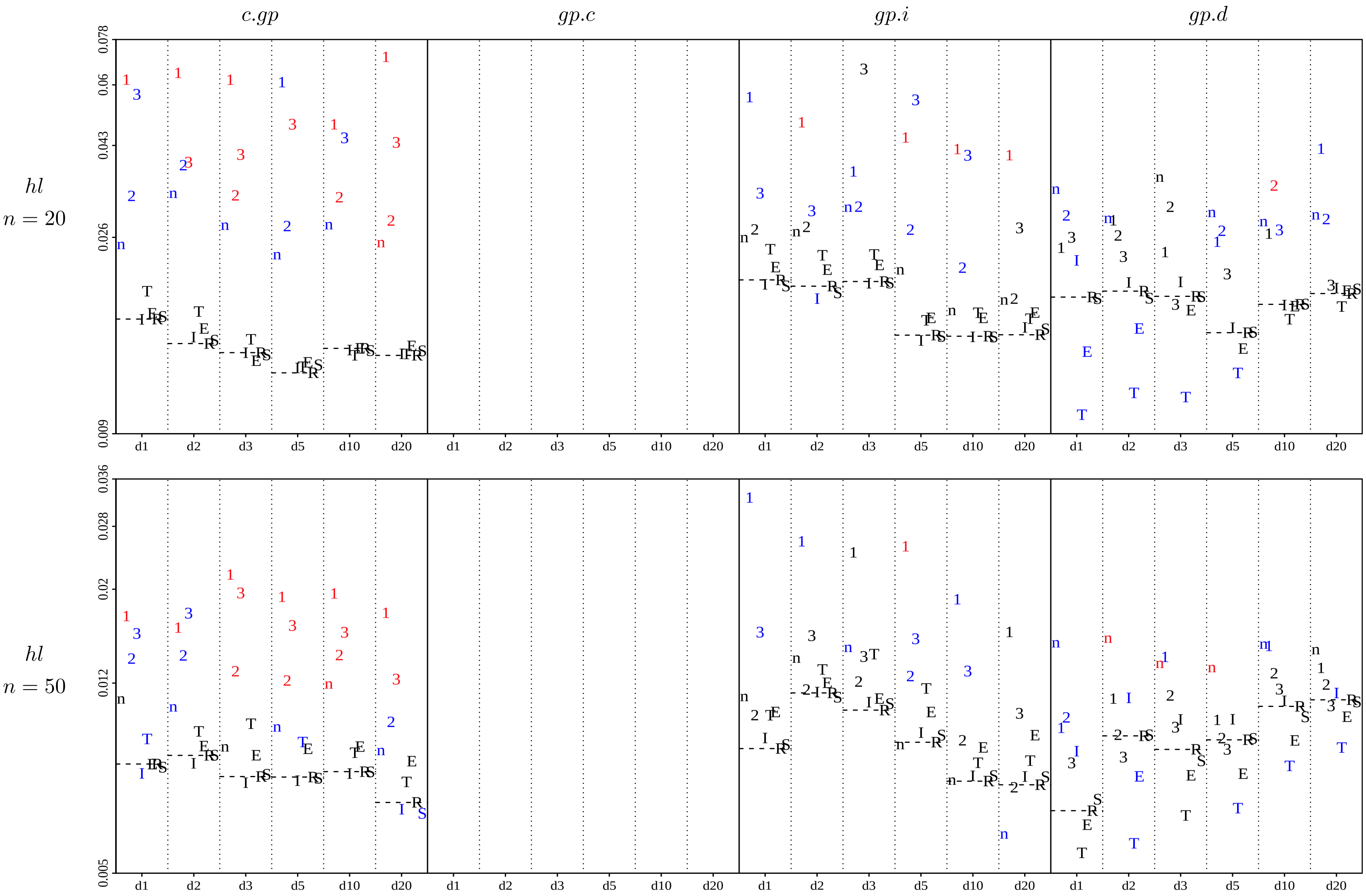


Figure A.18: High correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_R}$ (dashed). ‘n’: $\widehat{\phi}_{naive}$; ‘1’: $\widehat{\phi}_{HT_1}$; ‘2’: $\widehat{\phi}_{HT_2}$; ‘3’: $\widehat{\phi}_{HT_3}$; ‘I’: $\widehat{\phi}_{GP_I}$; ‘T’:, $\widehat{\phi}_{GP_T}$; ‘E’: $\widehat{\phi}_{GP_E}$; ‘R’: $\widehat{\phi}_{GP_R}$; ‘S’: $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

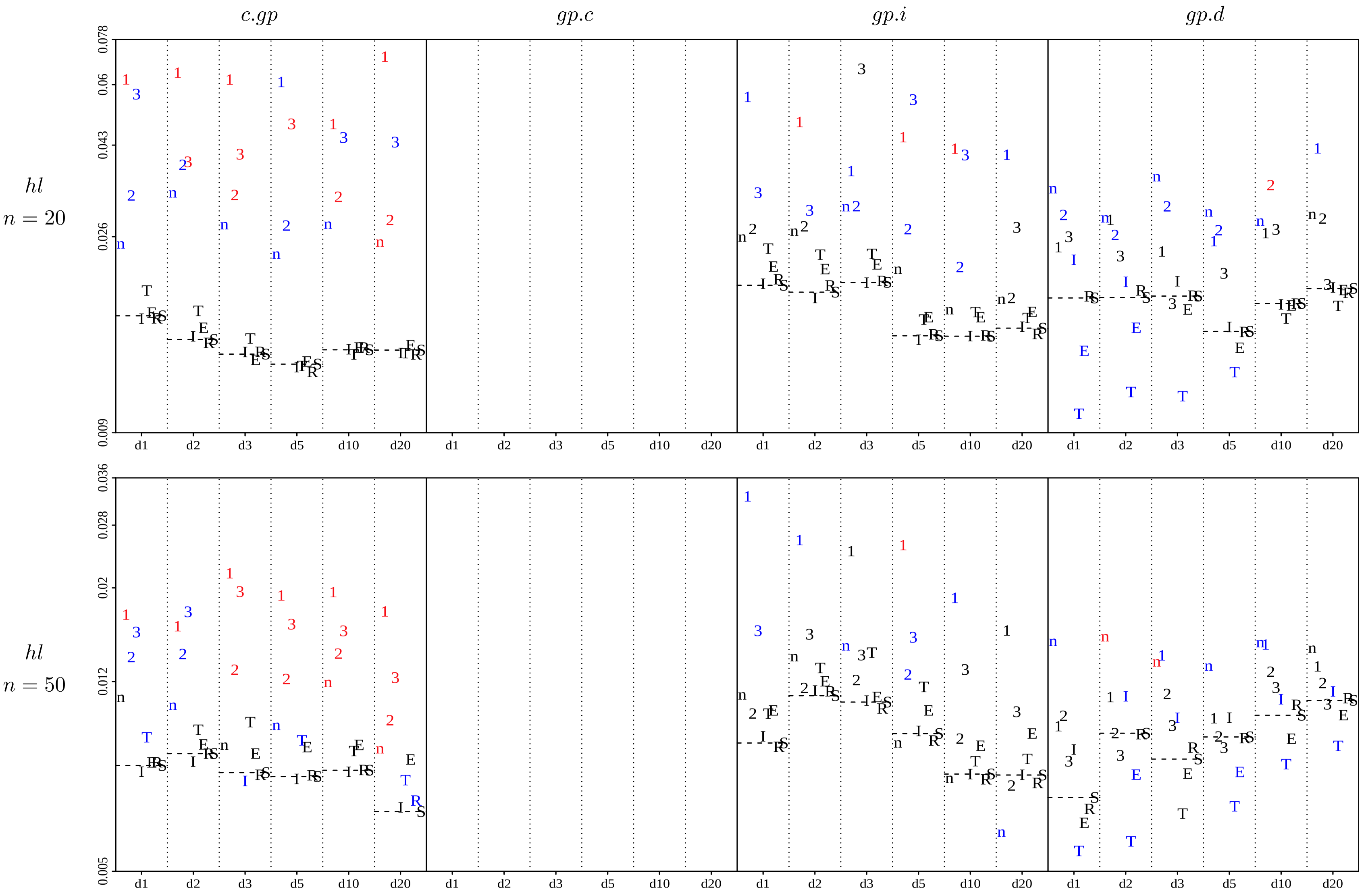


Figure A.19: High correlation and less wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_S}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

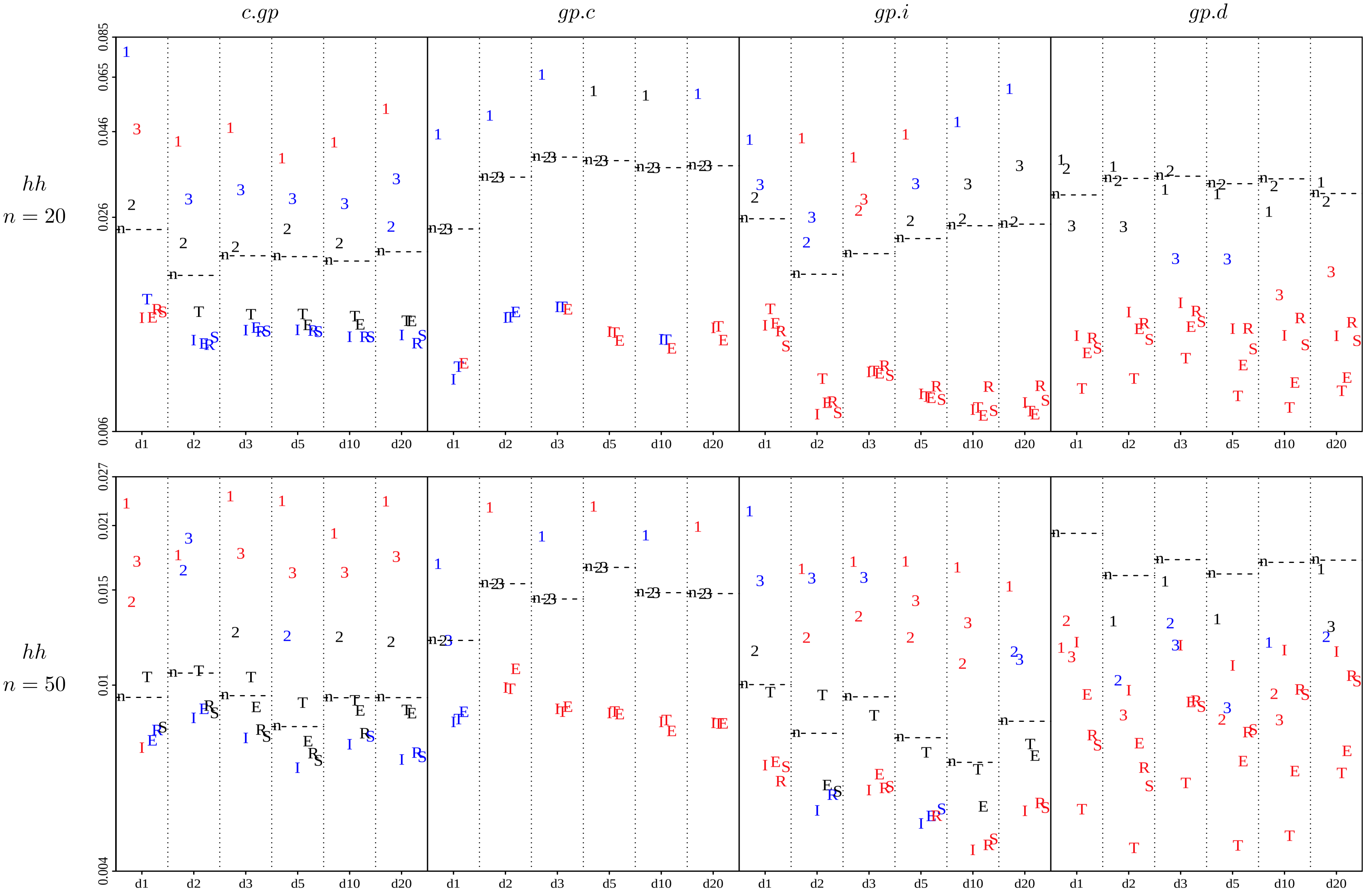


Figure A.20: High correlation and highly wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{naive}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

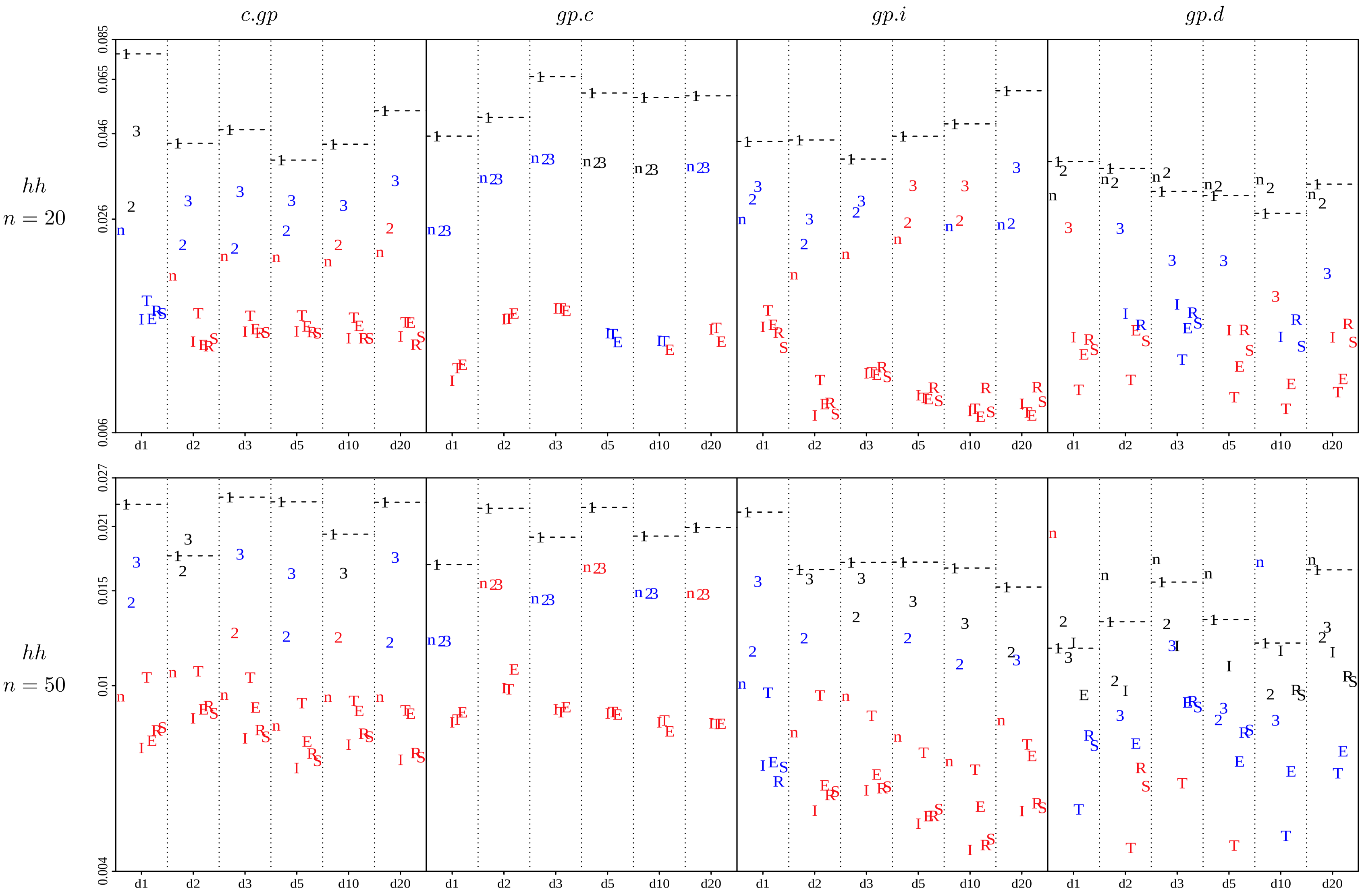


Figure A.21: High correlation and highly wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{HT_1}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

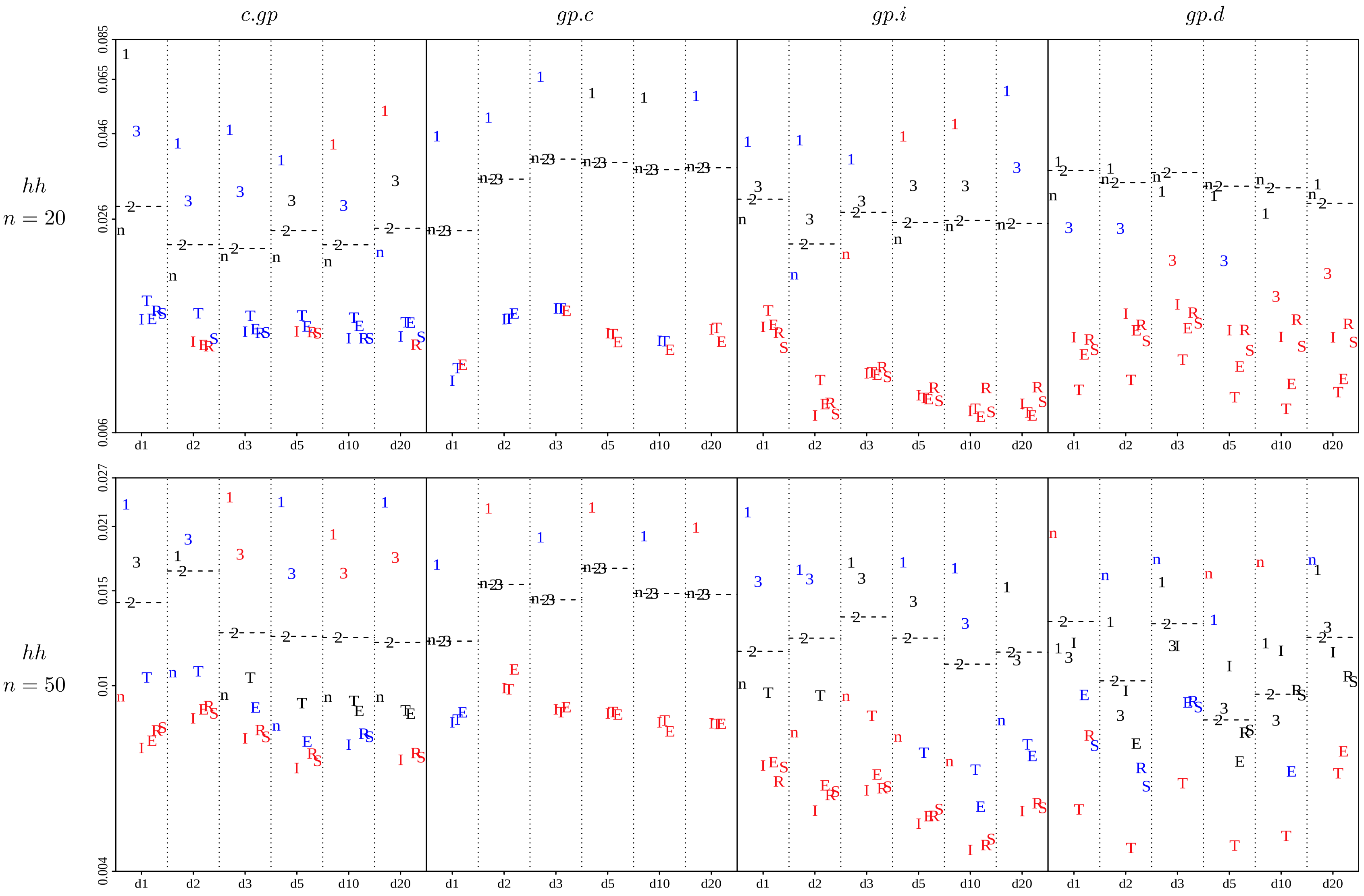


Figure A.22: High correlation and highly wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{HT_2}$ (dashed). ‘n’: $\widehat{\phi}_{naive}$; ‘1’: $\widehat{\phi}_{HT_1}$; ‘2’: $\widehat{\phi}_{HT_2}$; ‘3’: $\widehat{\phi}_{HT_3}$; ‘I’: $\widehat{\phi}_{GP_I}$; ‘T’:, $\widehat{\phi}_{GP_T}$; ‘E’: $\widehat{\phi}_{GP_E}$; ‘R’: $\widehat{\phi}_{GP_R}$; ‘S’: $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

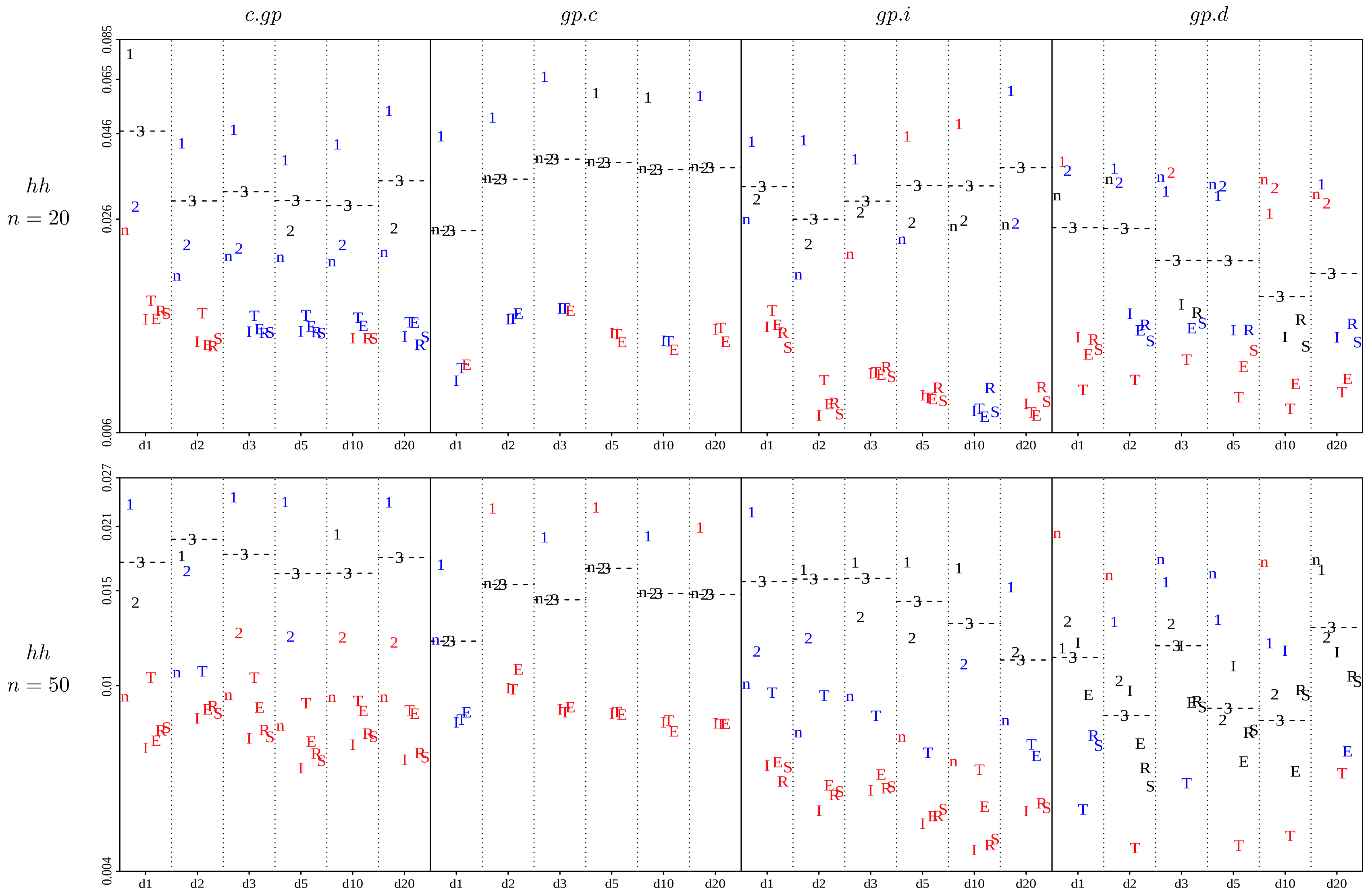


Figure A.23: High correlation and highly wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{HT_3}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

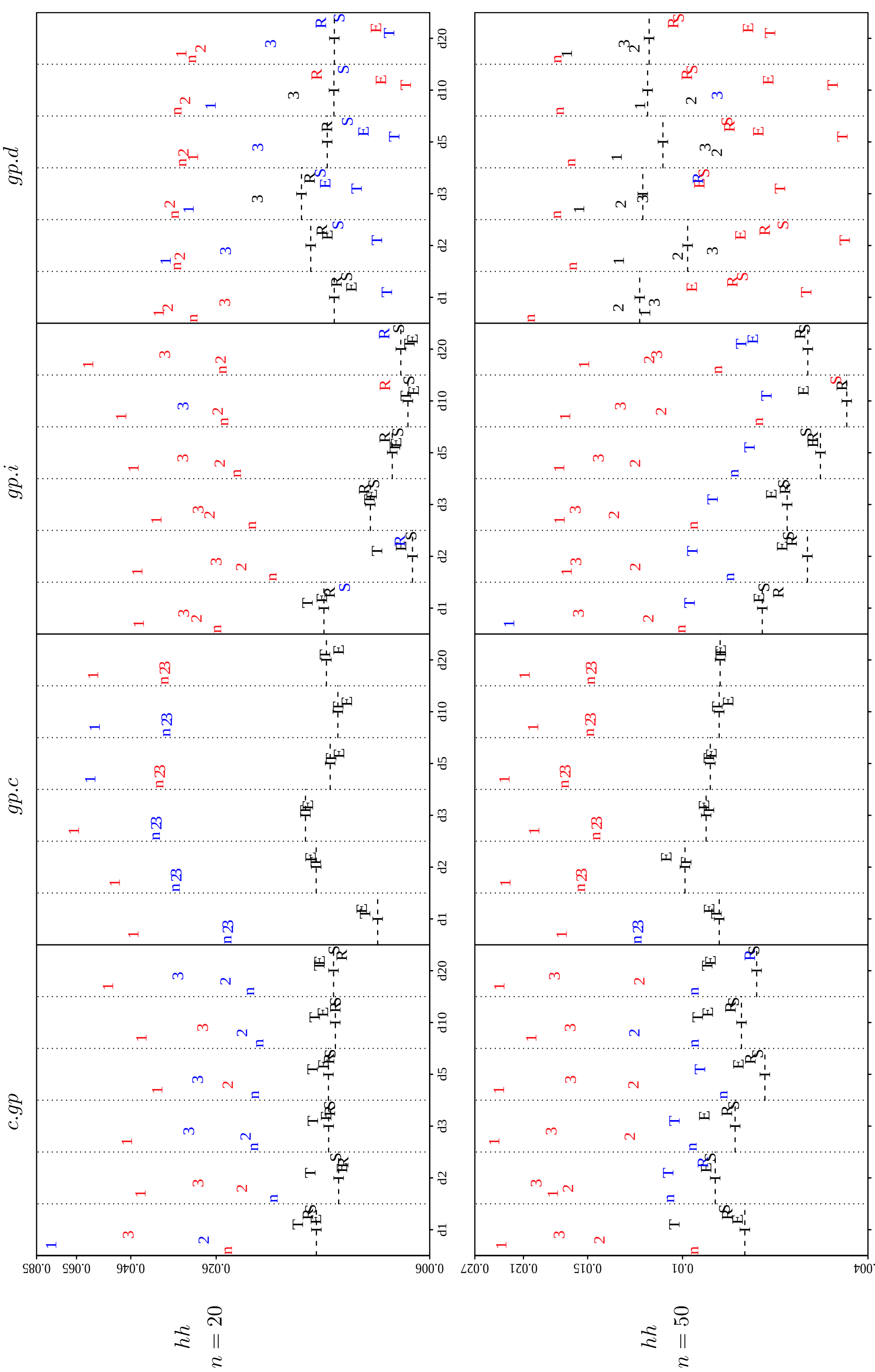


Figure A.24: High correlation and highly wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_I}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T': $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

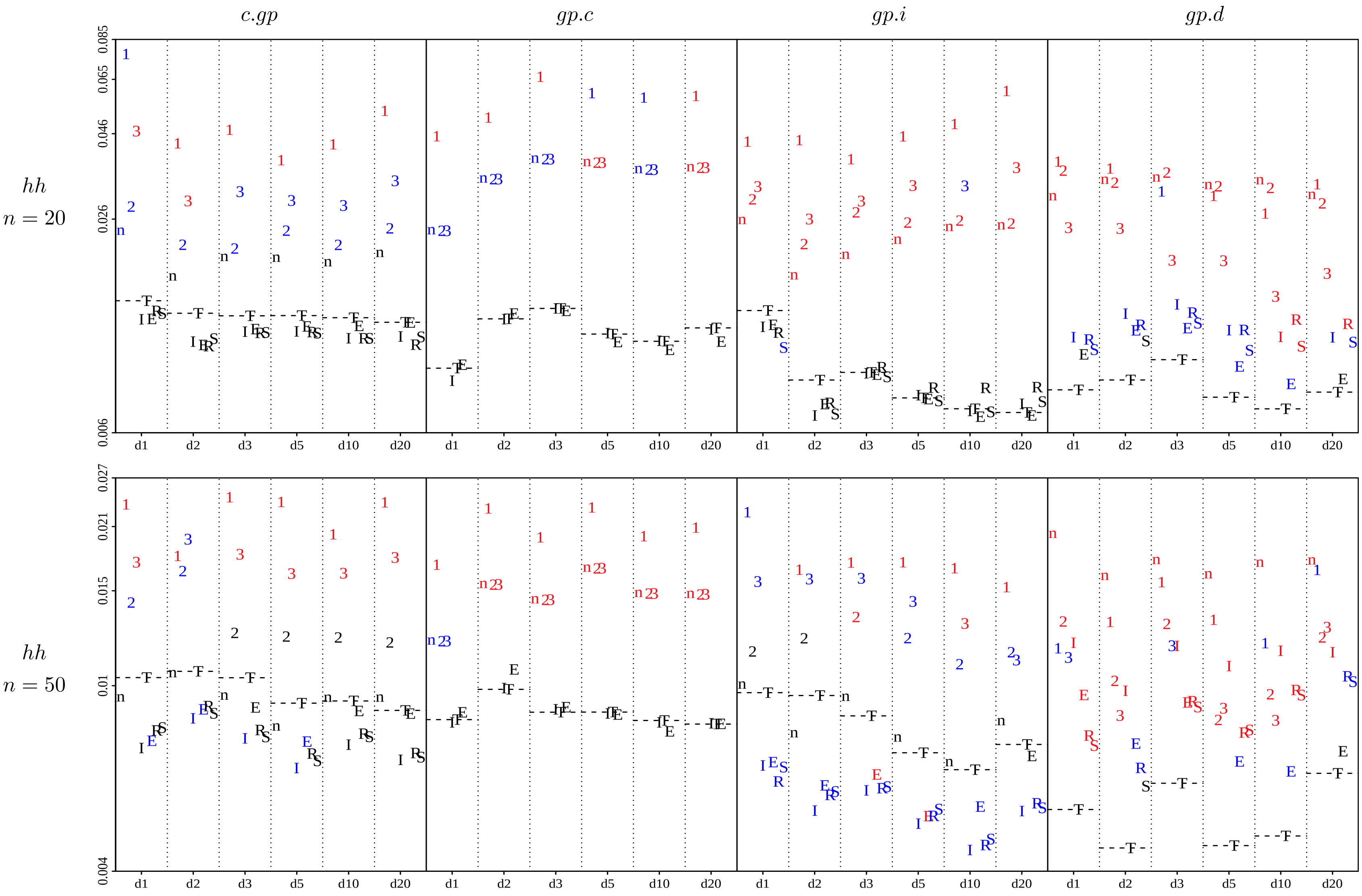


Figure A.25: High correlation and highly wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_T}$ (dashed). ‘n’: $\widehat{\phi}_{naive}$; ‘1’: $\widehat{\phi}_{HT_1}$; ‘2’: $\widehat{\phi}_{HT_2}$; ‘3’: $\widehat{\phi}_{HT_3}$; ‘I’: $\widehat{\phi}_{GP_I}$; ‘T’:, $\widehat{\phi}_{GP_T}$; ‘E’: $\widehat{\phi}_{GP_E}$; ‘R’: $\widehat{\phi}_{GP_R}$; ‘S’: $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

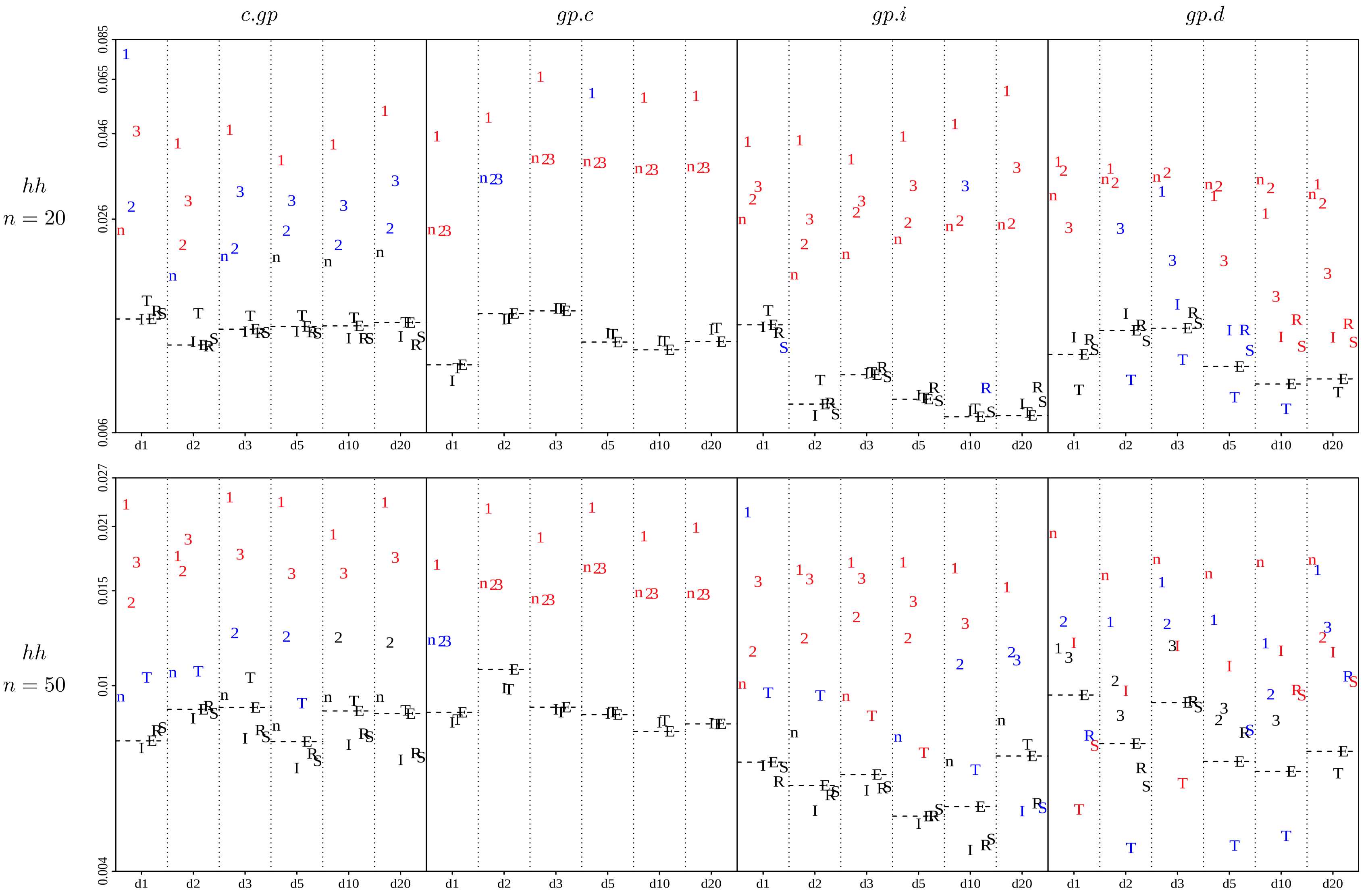


Figure A.26: High correlation and highly wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_E}$ (dashed). ‘n’: $\widehat{\phi}_{naive}$; ‘1’: $\widehat{\phi}_{HT_1}$; ‘2’: $\widehat{\phi}_{HT_2}$; ‘3’: $\widehat{\phi}_{HT_3}$; ‘I’: $\widehat{\phi}_{GP_I}$; ‘T’:, $\widehat{\phi}_{GP_T}$; ‘E’: $\widehat{\phi}_{GP_E}$; ‘R’: $\widehat{\phi}_{GP_R}$; ‘S’: $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

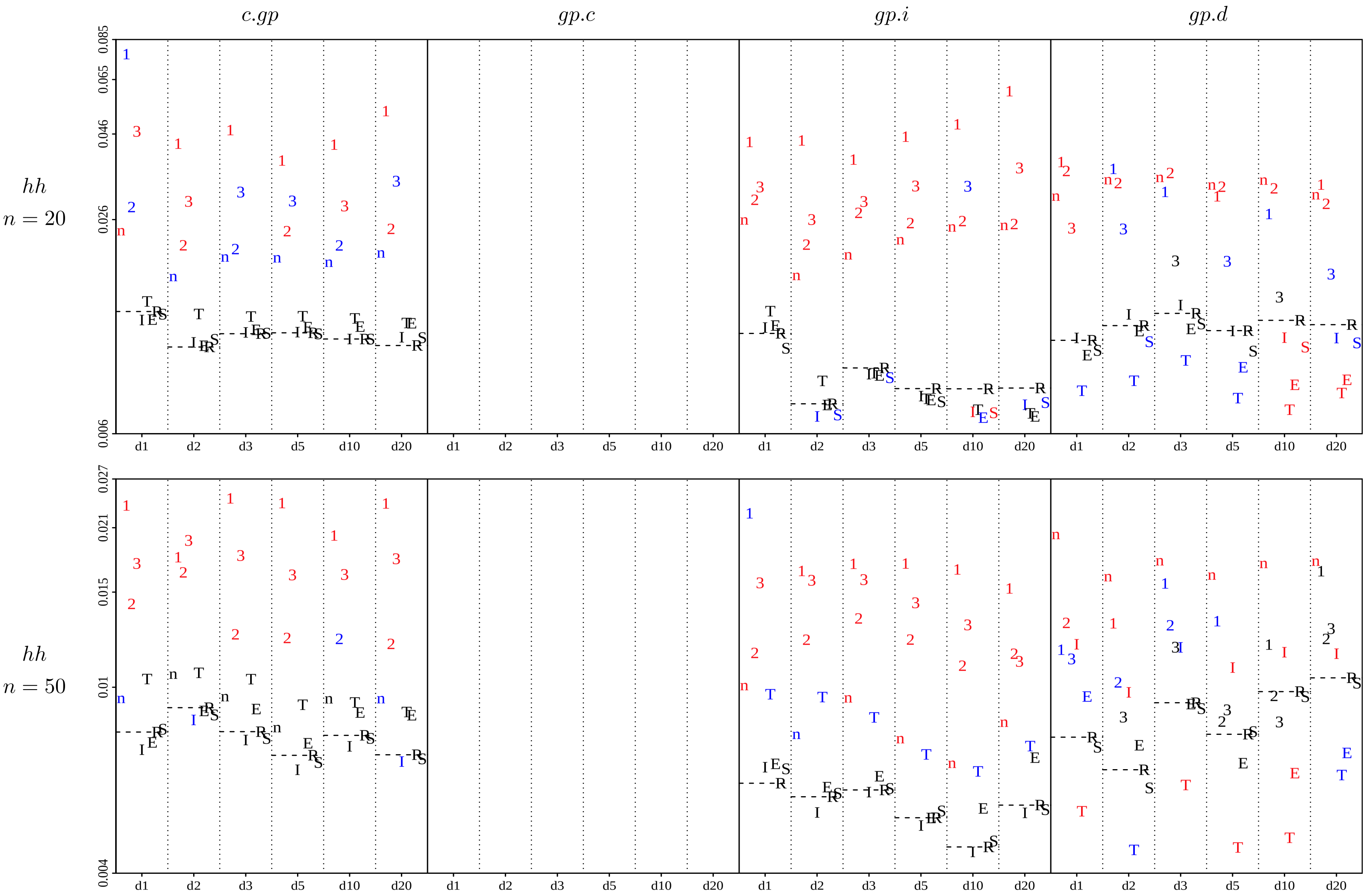


Figure A.27: High correlation and highly wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_R}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

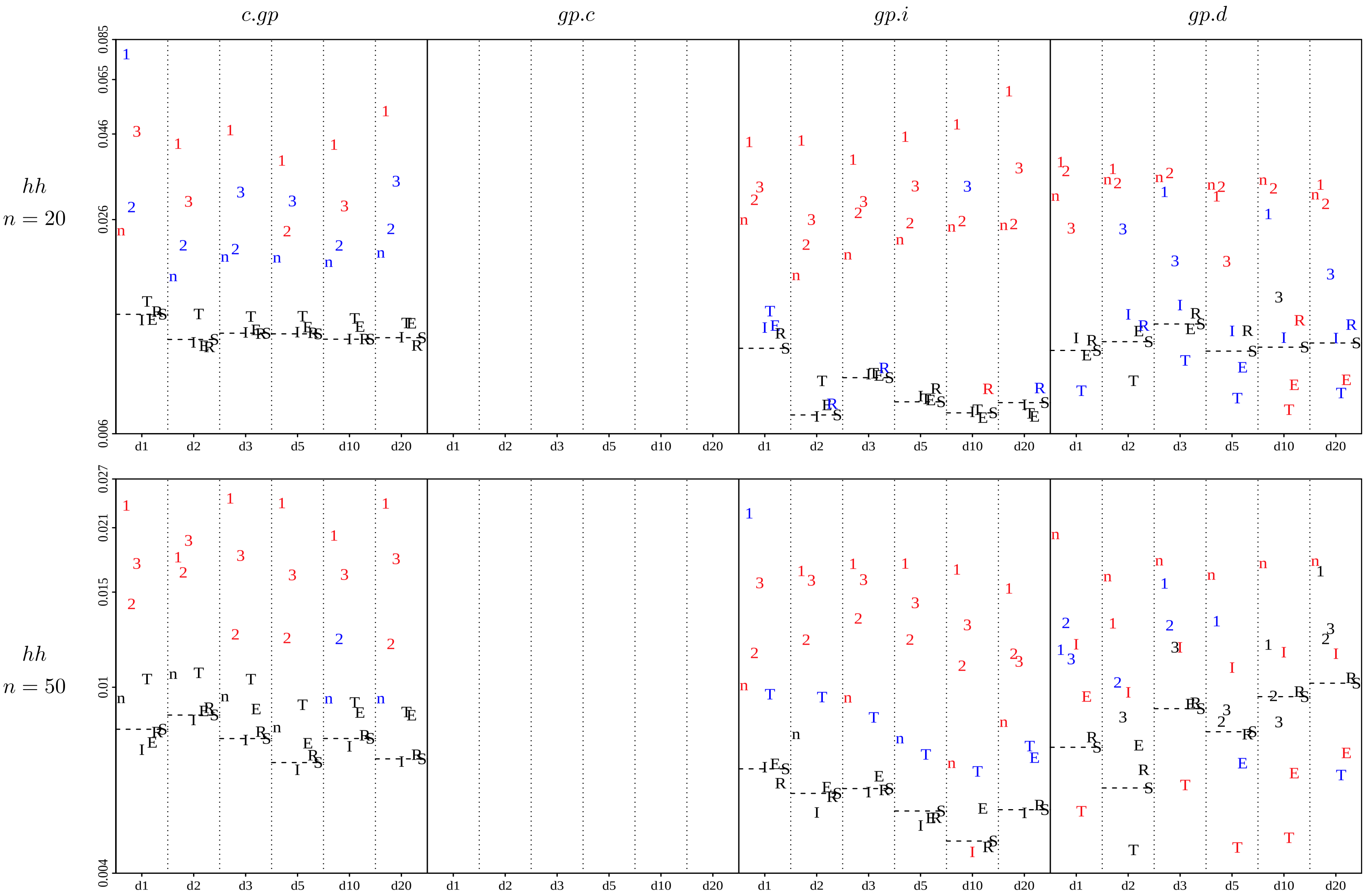


Figure A.28: High correlation and highly wiggly. Mean sqaure errors (MSE) in logarithm with comparison to that of $\widehat{\phi}_{GP_S}$ (dashed). 'n': $\widehat{\phi}_{naive}$; '1': $\widehat{\phi}_{HT_1}$; '2': $\widehat{\phi}_{HT_2}$; '3': $\widehat{\phi}_{HT_3}$; 'I': $\widehat{\phi}_{GP_I}$; 'T':, $\widehat{\phi}_{GP_T}$; 'E': $\widehat{\phi}_{GP_E}$; 'R': $\widehat{\phi}_{GP_R}$; 'S': $\widehat{\phi}_{GP_S}$. Red: p-value < 0.01; blue: $0.01 \leq$ p-value < 0.1. Top: $n = 20$; bottom: $n = 50$.

Bibliography

Austin, P. C. (2008). Goodness-of-fit Diagnostics for the Propensity Score Model When Estimating Treatment Effects Using Covariate Adjustment with the Propensity Score. *Pharmacoepidemiology and Drug Safety*, **17** 1202–1217.

Bickel, P. J. and Kleijin, B. J. (2012). The Semi-parametric Bernstein-von Mises Theorem. *The Annals of Statistics*, **40** 206–237.

Cochran, W. G. (1957). Analysis of Covariance: Its Nature and Uses. *Biometrics*, **13** 261–288.

Cochran, W. G. (1965). The Planning of Observational Studies of Human Populations. *Journal of the Royal Statistical Society, Series A*, **128** 234–266.

Cochran, W. G. (1968). The Effectiveness of Adjustment by Subclassification in Removing Bias in Observational Studies. *Biometrics*, **24** 295–313.

Cochran, W. G. and Rubin, D. B. (1973). Controlling Bias in Obsevational Studies: A Review. *SANKHYA: The Indian Journal of Statistics, Series A*, **35** 417–446.

Dehejia, R. H. (2005). Practical Propensity Score Matching: A Reply to Smith and Todd. *Journal of Econometrics*, **125** 355–364.

Dehejia, R. H. and Wahba, S. (2002). Propensity Score-Matching Methods for Nonexperimental Causal Studies. *The Review of Economics and Statistics*, **84** 151–161.

Gelman, A. (2007). Struggles with Survey Weighting and Regression Modelling. *Statistical Science*, **22** 153–164.

Hastings, W. K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, **57** 97–109.

Horvitz, D. G. and Thompson, D. J. (1952). The Generalization of Sampling without Replacement from a Finite Universe. *Journal of the American Statistical Association*, **47** 663–685.

Kang, J. D. Y. and Schafer, J. L. (2007). Demystifying Double Robustness: A Comparison of Alternative Strategies for Estimating a Population Mean from Incomplete Data. *Statistical Science*, **22** 523–539.

LaLonde, R. J. (1986). Evaluating the Evaluations of Training Programs with Experimental Data. *The American Economic Review*, **76** 604–620.

Murray, I., Adams, R. P. and MacKay, D. J. (2010). Elliptical Slice Sampling. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, **9** 541–548.

Neal, R. M. (1998). Regression and Classification Using Gaussian Process Priors. *J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith (Eds.), Bayesian Statistics*, **6** 475–501.

Neal, R. M. (2001). Annealed Importance Sampling. *Statistics and Computing*, **11** 125–139.

Neal, R. M. (2003). Slice Sampling. *The Annals of Statistics*, **31** 705–767.

Rasmussen, C. E. and William, C. K. I. (2006). *Gaussian Processes for Machine Learning.* MIT press.

Ritov, Y., Bickel, P. J., Gamst, A. C. and Kleijin, B. J. K. (2013). The Bayesian Analysis of Complex High-dimensional Models: Can it be CODA? *arXiv1203.5471v2* 1–35.

Robins, J. M. and Ritov, Y. (1997). Towards a Curse of Dimensionality Appropriate (CODA) Asymptotic Theory for Semi-parametric Models. *Statistics in Medicine*, **16** 285–319.

Robins, J. M. and Rotnitzky, A. (1995). Semiparametric Efficiency in Multivariate Regression Models with Missing Data. *Journal of the American Statistical Association*, **90** 122–129.

Robins, J. M., Sued, M., Lei-Gomez, Q. and Rotnitzky, A. (2007). Comment: Performance of Double-Robust Estimators When "Inverse Probability" Weights Are Highly Variable. *Statistical Science*, **22** 544–559.

Rosenbaum, P. R. and Rubin, D. B. (1983). The Central Role of the Propensity Score in Observational Studies for Causal Effects. *Biometrika*, **70** 41–55.

Rotnitzky, A., Lei, Q., Sued, M. and Robins, J. M. (2012). Improved Double-robust Estimation in Missing Data and Causal Inference Models. *Biometrika*, **99** 439–456.

Rubin, D. B. (2001). Using Propensity Scores to Help Design Observational Studies: Application to Tobacco Design. *Health Services and Outcomes Research Methodology*, **2** 169–188.

Rubin, D. B. (2007). The Design versus the Analysis of Observational Studies for Causal Effects: Parallels with the Design of Randomized Trials. *Statistics in Medicine*, **26** 20–36.

Scharfstein, D. O., Rotnitzky, A. and Robins, J. M. (1999). Adjusting for Nonignorable Drop-out Using Semiparametric Nonresponse Models / Comment / Rejoinder. *Journal of the American Statistical Association*, **94** 1096–1120.

Tan, Z. (2007). Comment: Understanding OR, PS and DR. *Statistical Science*, **22** 560–568.

Thompson, M. B. (2010). A Comparison of Methods for Computing Autocorrelation Time. *arXiv:1011.0175* 1–8.

Wasserman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*. Springer, New York.