

Adaptive Sampling for k -Nearest Neighbor Regression of Black-Box Models

Sören Bettels^{a,b}, Thomas Knispel^{*a,c}, and Stefan Weber^{a,b}

^aHouse of Insurance, Leibniz Universität Hannover

^bInstitute of Actuarial and Financial Mathematics, Leibniz Universität Hannover

^cBerlin School of Economics and Law

April 22, 2026

Abstract

We study the construction of k -nearest neighbor (k -NN) surrogate models for computationally expensive black-box systems when interest is focused on rare or extreme outcomes. The setting assumes a random input with known distribution and a black-box response that can be evaluated only at a limited number of design points. Approximation quality is measured by a tail-restricted $L^2(\nu)$ error, so the central problem is the placement of support points under a constrained simulation budget. We exploit partial structural information about the black box in the form of a growth condition that bounds output differences. Based on this information, we discretize the tail-focused error criterion and derive a computable upper bound for the resulting pathwise error. This bound induces a local score for the estimated error reduction from an additional support point and thereby yields sequential sampling rules. Since the bound is only a proxy for the true error reduction, purely exploitative optimization can perform poorly; the resulting framework therefore balances exploitation of the proxy criterion with exploration of the input space. This leads to semi-adaptive and fully adaptive sampling schemes that concentrate evaluations in regions where the probability-weighted black box varies strongly. We establish consistency results for the resulting k -NN surrogates in the discretized setting and study the computational implementation of the proposed methods. Numerical experiments with controlled functions and a simplified asset-liability management model show that the adaptive schemes are robust across specifications and can substantially improve tail-focused surrogate construction under severe evaluation constraints.

Keywords:

Black-box surrogate modeling; k -nearest neighbor regression; sequential design; adaptive sampling; rare-event simulation

1 Introduction

Many models used in science, engineering, and finance are effectively black boxes: the system can be evaluated for chosen inputs, but its internal structure is unknown or too complex for analytic treatment. In many such settings the input is random and the output distribution is studied by simulation. When each evaluation is computationally expensive, only a limited

*Corresponding author, e-mail: thomas.knispel@hwr-berlin.de

number of simulations can be performed, so the construction of accurate surrogate models becomes a central problem. We consider the case in which the main interest lies in rare or extreme outcomes, so that the surrogate should be accurate primarily in the tail of the output distribution.

Our objective is therefore not global prediction accuracy, but approximation quality under a prescribed input law in a region of rare outcomes. More precisely, we evaluate a surrogate by an $L^2(\nu)$ error restricted to inputs z for which the black-box output exceeds a threshold γ . This tail-focused criterion reflects the intended use of the surrogate: approximation error matters most where extreme outputs occur and where simulation mistakes are typically most consequential.

As surrogate model we consider k -nearest neighbor (k -NN) regression. This method is attractive here because it is simple, fully nonparametric, and local: predictions are formed from nearby observed function values and do not require a global parametric specification. At the same time, this locality makes its performance highly sensitive to the placement of the support points at which the black box is evaluated. Under a limited simulation budget, the statistical quality of the surrogate depends crucially on how these points are chosen.

A central feature of our framework is the separation between the target input law and the data-acquisition law. The random input $Z \sim \nu$ defines the performance criterion of the surrogate, whereas the training locations X_1, X_2, \dots are auxiliary design points generated by a sampling scheme chosen by the statistician. Thus the final approximation is evaluated under ν , but the locations at which the black box is queried need not themselves be distributed according to ν . This turns surrogate construction into a sequential design problem.

Because black-box evaluations are expensive, only a limited number of support points can be used. The problem is therefore one of efficient data acquisition: design points should be concentrated in regions where the probability-weighted black box varies strongly within the tail, and used more sparsely where the response is locally flat or statistically less relevant. In this sense, the task is not merely to sample rare events more often, but to place evaluations where they are most informative for tail-focused surrogate construction.

To guide this design problem we assume that partial structural information about the black box is available in the form of a growth condition. This condition bounds differences of the form $|s(x) - s(y)|$ by a known function of the inputs and thereby provides information about how rapidly the black box can vary across the input space. Such information is weaker than a parametric model, but still supplies useful nonparametric guidance for placing evaluation points in regions where large changes of the response are possible.

Our main methodological idea is to turn this structural information into a practical sampling rule. We first discretize the tail-focused error criterion on a grid and then use the growth condition to derive a computable upper bound for the resulting pathwise error. This bound yields a local score that quantifies the estimated error reduction produced by an additional support point. Sequential sampling is then driven by this score, which serves as a proxy for the true improvement in surrogate accuracy.

A key conceptual point is that this proxy should not be treated as exact information. Directly maximizing the estimated bound reduction corresponds to pure exploitation and is therefore

only a limiting case of the proposed approach. Since the bound is only an approximation to the true error reduction, overly greedy sampling can perform poorly when the growth condition is inaccurate or locally misleading. Effective data acquisition must therefore balance exploitation of promising regions with continued exploration of the input space. This perspective leads to a family of sampling methods with different degrees of adaptivity. As benchmarks we consider crude sampling from the target law and subset sampling restricted to the estimated tail region. Building on the bound-based score, we then develop semi-adaptive and fully adaptive proportional sampling schemes that combine exploitation and exploration in different ways. Simulated annealing appears in this framework as the purely exploitative boundary case, where the next support point is chosen by approximate maximization of the proxy criterion.

On the theoretical side, we derive a computable upper bound for the discretized tail error, show that the associated error reduction admits a local representation suitable for implementation, and analyze the resulting sampling kernels in the discretized setting. In particular, we establish consistency results for the corresponding k -NN surrogates under the proposed sequential design schemes. The empirical study has two parts. First, controlled examples with tractable black-box functions allow us to isolate the role of the growth condition and to examine how the different sampling rules behave under accurate specification and misspecification. Second, we consider a simplified asset–liability management model, which shows that the proposed methods remain meaningful in a more realistic setting where the black box is structurally richer and the tail region is of direct practical interest. The main empirical message is that the semi-adaptive and fully adaptive methods provide the most robust performance across different specifications of the growth condition, whereas methods based on near-deterministic optimization of the proxy criterion can perform very well only when the growth condition is sufficiently accurate.

Our work is related to several strands of literature. First, it connects to the theory of k -NN regression and nonparametric learning, where classical results study consistency and statistical properties under i.i.d. sampling of the training data. In contrast, the present paper studies the sequential design of training points under a fixed evaluation budget. Second, the problem is related to rare-event simulation and importance sampling, where sampling distributions are modified to improve the estimation of tail probabilities. Our objective is different: rather than estimating a probability, we construct a surrogate model whose accuracy is targeted to the tail. Finally, the adaptive sampling schemes are conceptually related to sequential Monte Carlo and stochastic optimization methods, but here the proxy criterion is derived from structural growth information about the black box rather than from likelihood or gradient information.

The main contributions of the paper are as follows:

- (i) We formulate tail-focused surrogate construction for computationally expensive black-box models as a sequential design problem for k -NN regression under a constrained evaluation budget.
- (ii) Using a growth condition, we derive a computable upper bound for the discretized pathwise tail error and the associated local score for estimated error reduction, which yields adaptive sampling rules balancing exploitation and exploration.

- (iii) We provide theoretical support through consistency results in the discretized setting and empirical support through controlled function examples and an asset-liability management application, showing that the adaptive methods are robust across different specifications of the growth condition.

1.1 Literature

For broad introductions to machine learning and black-box modeling, see Shalev-Shwartz & Ben-David (2014), Mohri, Rostamizadeh & Talwalkar (2018), and Hastie, Tibshirani & Friedman (2009). Reviews of black-box applications in finance and autonomous systems are provided in Huang, Chai & Cho (2020) and Corso et al. (2021), respectively.

The origins of k -NN regression go back to Fix & Hodges (1952) and Cover & Hart (1967). Standard references include Hastie, Tibshirani & Friedman (2009), Györfi et al. (2002), and Devroye, Györfi & Lugosi (2013). Potential limitations of the method are discussed in Beyer et al. (1999), while surveys on extensions of k -NN regression can be found in Taunk et al. (2019), Bhatia (2010), and Jiang et al. (2007).

Examples of financial applications of k -NN regression are given in Tajmouati et al. (2021), Alkhatib et al. (2013), Chen & Hao (2017), and Huang, Chai & Cho (2020). Its broader role in data mining and related applications is discussed in Wu et al. (2008), Dhanabal & Chandramathi (2011), and Kwon & Lee (2000).

Techniques for validating black-box models are reviewed in Corso et al. (2021). For rare-event estimation in black-box settings, see Arief et al. (2021) and Huang, Lam & Zhao (2018); broader overviews of rare-event simulation are given in Juneja & Shahabuddin (2006), L'Ecuyer, Mandjes & Tuffin (2009), and Bucklew (2004). Our sampling methods are also related to ideas from sequential Monte Carlo; standard references are Chopin, Papaspiliopoulos, et al. (2020) and Doucet, De Freitas & Gordon (2001).

1.2 Outline

The paper is organized as follows: Section 2 formally states the problem and the objectives. In Section 3, we discretize the pathwise mean square error on a grid, construct a computable upper error bound in terms of a growth condition, encoding partial information about the black box, and propose effective sampling methods which balance the exploitation of large estimated error reductions and exploration of the input space. Specific sampling methods and challenges encountered during practical implementation are in the core of Section 4. Moreover, for the sampling kernels employed, we provide theoretical insights into the asymptotic behavior of k -NN regression with increasing sample size. Our case studies are presented in Sections 5 and 6. In Section 5, we focus on simple transformations of random variables and compare the accuracy of different sampling methods and growth assumptions. Section 6 demonstrates the application of our methods to a more complex asset-liability management model. The proofs and detailed information on the practical implementation of the sampling methods are postponed to the appendix.

2 The Problem of Efficient Data Acquisition

In this section we formalize the problem, introduce notation, and state the objectives of the paper. Let $s : S \rightarrow \mathbb{R}$ be measurable on $S \subseteq \mathbb{R}^d$, and assume that evaluating s is computationally expensive. Let Z be an S -valued random vector with known distribution ν on (S, \mathcal{S}) , representing the random input to $s(\cdot)$. We treat $s(\cdot)$ as a black box: its internal structure is unknown, but $s(z)$ can be computed for selected $z \in S$ at high cost. Thus the law of $s(Z)$ is accessible only through costly simulation, and is not assumed to belong to a parametric family.

Our objective is to construct an approximation $\hat{s}(\cdot)$ of the black-box function $s(\cdot)$ that is accurate in the region of extreme outcomes, that is, for $s(z) \geq \gamma$ with threshold $\gamma \in \mathbb{R}$. The approximation error is evaluated using the mean squared error¹ (MSE) restricted to this extreme region,

$$\int_{\{z \in S : s(z) \geq \gamma\}} (\hat{s}(z) - s(z))^2 d\nu(z), \quad (1)$$

where ν denotes the distribution of the input Z . A typical example arises in risk management and asset-liability management, where $s(z)$ represents the pathwise computation of the future net asset value of a bank or insurance company within an internal model used to determine regulatory capital requirements.

A standard surrogate for $s(\cdot)$ is k -nearest neighbor (k -NN) regression, $k \in \mathbb{N}$, built from an auxiliary design sequence $(X_i, s(X_i))_{i \in \mathbb{N}}$. Here the X_i are algorithmic sampling locations in (S, \mathcal{S}) , not realizations of the target input $Z \sim \nu$; their joint law is chosen by the data-acquisition scheme, while performance is evaluated under ν . The responses $s(X_i)$ are obtained by evaluating the black box at the sampled inputs. Given $(X_1, s(X_1)), \dots, (X_N, s(X_N))$ with sample size $N \in \mathbb{N}$, and a query point $z \in S$, let

$$(X_{(1,N)}(z), s(X_{(1,N)}(z))), \dots, (X_{(N,N)}(z), s(X_{(N,N)}(z)))$$

denote the sample reordered by increasing Euclidean distance $|X_i - z|$.² We then refer to $X_{(k,N)}(z)$ as the k -th nearest neighbor of z . The k -NN regression is a nonparametric regression method that predicts a model based on a given data set by returning the mean of the function values corresponding to inputs near the specified point, see, e.g., Devroye, Györfi & Lugosi (2013) and Györfi et al. (2002) for a comprehensive introduction and results on the performance of k -NN regressions.

Definition 2.1. Let $(X_1, s(X_1)), \dots, (X_N, s(X_N))$ be sample pairs of inputs and outputs. For $k \leq N$, the k -NN regression at $z \in \mathbb{R}^d$ is defined as

$$\hat{s}_{k,N}(z) = \frac{1}{k} \sum_{i=1}^k s(X_{(i,N)}(z)),$$

where $X_{(i,N)}(z)$ is the i -th nearest neighbor of z .

¹Throughout we assume $s(\cdot) \in L^2(\nu)$.

²Ties may occur when $|X_i - z| = |X_j - z|$ for distinct $i, j \in \{1, \dots, N\}$, so that the k -th nearest neighbor is not uniquely defined; see Györfi et al. (2002) for standard tie-breaking rules. We break ties by augmenting the data to $(X_i, s(X_i), U_i)$ with i.i.d. $U_i \sim \text{Unif}(0, 1)$ and ordering lexicographically by $(|X_i - z|, U_i)$. We suppress the auxiliary variables in the notation and, for simplicity, treat ties as a null event.

Crude k -NN regression corresponds to the basic case where the X_i in the the initial sequence $(X_i, s(X_i))_{i \in \mathbb{N}}$ are sampled independently and identically distributed (i.i.d.). However, the approximation error (1) of a k -NN regression function $\hat{s}_{k,N}(\cdot)$ can be influenced by the stochastic nature of the sampling method employed. In contrast to crude k -NN regression, we begin by sampling X_1 from an initial distribution $\mu_1 : \mathcal{S} \rightarrow [0, 1]$. For subsequent samples X_i , where $i \geq 2$, we utilize stochastic kernels $\mu_i : \mathcal{S}^{i-1} \times \mathcal{S} \rightarrow [0, 1]$, which define the conditional distribution of X_i given the previous samples. To describe the distribution of (X_1, \dots, X_N) , we introduce the distribution $\mathbb{P}_N = \mu_1 \otimes \mu_2 \otimes \dots \otimes \mu_N : \mathcal{S}^{\otimes N} \rightarrow [0, 1]$. According to the Ionescu-Tulcea extension theorem (see Theorem 14.32 in Klenke (2020) or Theorem 6.17 in Kallenberg (2002)), there exists a unique distribution \mathbb{P} on $(\mathcal{S}^{\mathbb{N}}, \mathcal{S}^{\otimes \mathbb{N}})$, which satisfies the relation:

$$\mathbb{P} \circ \pi_N^{-1}(A) = \mathbb{P}_N(A), \quad \forall N \in \mathbb{N}, A \in \mathcal{S}^{\otimes N}, \quad (2)$$

where $\pi_N : \mathcal{S}^{\mathbb{N}} \mapsto \mathcal{S}^N$ is the projection on the first N coordinates, defined by $\pi(x_1, x_2, \dots) = (x_1, \dots, x_N)$. Intuitively, the probability measure \mathbb{P} encodes the sampling method.

We aim to identify \mathbb{P} that minimizes the expected error in (1), thereby reducing the number of samples N required to achieve an accurate approximation. To assess the error in the k -NN regression $\hat{s}_{k,N}(\cdot)$, we consider the expected MSE over $\{z \in \mathcal{S} | s(z) \geq \gamma\}$, defined as

$$\mathbb{E}_{\mathbb{P}} \left[\int_{\{z \in \mathcal{S} | s(z) \geq \gamma\}} (\hat{s}_{k,N}(z) - s(z))^2 d\nu(z) \right]. \quad (3)$$

In this setting, the choice $\mathbb{P} = \mu^{\otimes \mathbb{N}}$ for a probability measure μ on $(\mathcal{S}, \mathcal{S})$ yields the baseline method of crude k -NN regression. If ν is absolutely continuous with respect to μ with essentially bounded Radon–Nikodým derivative, then Corollary 2.2 recovers the standard consistency result for crude k -NN regression (Theorem 6.1 in Györfi et al. (2002)) in our setting, extended to allow a potential change of measure and the inclusion of the threshold γ restricting attention to extreme outcomes of the black box $s(\cdot)$.

Corollary 2.2. *Let $(X_i, s(X_i))_{i \in \mathbb{N}}$ be i.i.d. with law $\mathbb{P} = \mu^{\otimes \mathbb{N}}$, and assume that $\nu \ll \mu$ with $\frac{d\nu}{d\mu} \leq C$ μ -a.s. Let $\hat{s}_{k_N, N}(\cdot)$ denote the k -NN regression estimator based on the first N samples. If $k_N \rightarrow \infty$ and $k_N/N \rightarrow 0$ as $N \rightarrow \infty$, then $\hat{s}_{k_N, N}(\cdot)$ is weakly consistent in the sense that*

$$\lim_{N \rightarrow \infty} \mathbb{E}_{\mathbb{P}} \left[\int_{\{z \in \mathcal{S} : s(z) \geq \gamma\}} (\hat{s}_{k_N, N}(z) - s(z))^2 d\nu(z) \right] = 0.$$

Proof. Since $\frac{d\nu}{d\mu} \leq C$ μ -a.s., we have

$$0 \leq \mathbb{E}_{\mathbb{P}} \left[\int_{\{z \in \mathcal{S} : s(z) \geq \gamma\}} (\hat{s}_{k_N, N}(z) - s(z))^2 d\nu(z) \right] \leq C \cdot \mathbb{E}_{\mathbb{P}} \left[\int (\hat{s}_{k_N, N}(z) - s(z))^2 d\mu(z) \right].$$

The right-hand side converges to 0 as $N \rightarrow \infty$ by Theorem 6.1 in Györfi et al. (2002). □

3 Error Analysis for k -NN Regressions of Black Box Models

We assume that partial information about the black box $s(\cdot)$ is available through a growth condition $\varphi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_+$ satisfying

$$|s(x) - s(y)| \leq \varphi(x, y), \quad x, y \in \mathbb{R}^d.$$

We further assume that for all $x, y, y' \in \mathbb{R}^d$ the function $\varphi(\cdot, \cdot)$ satisfies

- (i) $\varphi(x, y) > 0$ if $x \neq y$,
- (ii) $\varphi(x, y) = \varphi(y, x)$,
- (iii) $\varphi(x, y) \leq \varphi(x, y')$ whenever $|x - y| \leq |x - y'|$.

Our objective is to design a sampling method that uses the growth condition $\varphi(\cdot, \cdot)$ to construct accurate k -NN approximations of the black-box function $s(\cdot)$, particularly for inputs where $s(\cdot)$ exceeds a threshold γ . Since the true MSE cannot be optimized directly and the information encoded in $\varphi(\cdot, \cdot)$ is only a proxy, the resulting procedure combines optimization of a computable error bound with stochastic exploration of the sampling space.

The construction proceeds in three steps. First, the MSE is discretized on a grid to obtain a tractable approximation of the error. Second, the growth condition is used to derive a computable upper bound on the discretized error and an associated measure of error reduction. Third, candidate samples are either selected by direct optimization of the estimated bound reduction or drawn from sampling kernels defined by transformations of this quantity. Direct optimization corresponds to pure exploitation and serves as a limiting case, but is generally not reliable because the bound reduction is only a proxy for the true error reduction. The sampling-kernel approach therefore balances exploitation of promising regions with continued exploration of the input space.

Step 1: Discretization

To obtain a tractable approximation of the MSE in (3), we discretize the input space on the equidistant grid

$$G_\delta := \{(\delta i_1, \dots, \delta i_d) \mid i_1, \dots, i_d \in \mathbb{Z}\} \subset S$$

with mesh size $\delta > 0$. On this grid, the distribution ν is approximated by the discrete measure ν_{G_δ} defined by

$$\nu_{G_\delta}(z) = \nu(C_{z,\delta}), \quad z \in G_\delta,$$

where $C_{z,\delta}$ denotes the d -dimensional cube centered at z with side length δ .

For given samples $(X_1, \dots, X_N) \in G_\delta^N$, the objective is to choose the next sample $X_{N+1} \in G_\delta$ so as to reduce the regression error of $\hat{s}_{k,N+1}(\cdot)$. The overall objective remains the expected MSE in (3). However, since the sampling procedure is constructed sequentially, we analyze the MSE inside the expectation pathwise and approximate it on the grid.

Under this discretization, the pathwise error after adding the sample X_{N+1} is approximated by

$$\mathcal{E}_{N+1} := \sum_{z \in G_\delta} \mathbb{1}_{\{s(z) \geq \gamma\}} \left(\frac{1}{k} \sum_{i=1}^k s(X_{(i,N+1)}(z)) - s(z) \right)^2 \nu_{G_\delta}(z). \quad (4)$$

Step 2: Error Analysis for an Upper Bound

The pathwise objective is to choose X_{N+1} to minimize the discretized error \mathcal{E}_{N+1} given in (4), conditional on the first N samples X_1, \dots, X_N . However, since s is a black box and evaluating s on the full grid G_δ remains computationally expensive, directly minimizing \mathcal{E}_{N+1} over X_{N+1} is infeasible.

Instead, we derive a computable upper bound based on the growth condition $\varphi(\cdot, \cdot)$,

$$\bar{\mathcal{E}}_{N+1} := \frac{1}{k^2} \sum_{z \in G_\delta} \mathbb{1}_{\{s(z) \geq \gamma\}} \left(\sum_{i=1}^k \varphi(X_{(i,N+1)}(z), z) \right)^2 \nu_{G_\delta}(z) \geq \mathcal{E}_{N+1}.$$

Sequential sampling is then guided by the reduction of this bound. Since $\bar{\mathcal{E}}_N$ does not depend on the candidate point X_{N+1} , minimizing $\bar{\mathcal{E}}_{N+1}$ is equivalent to maximizing the bound reduction

$$\bar{\mathcal{E}}_N - \bar{\mathcal{E}}_{N+1}.$$

The following lemma shows that this quantity is non-negative and therefore provides a meaningful surrogate measure of improvement.

Lemma 3.1. *For all samples $(X_1, \dots, X_N, X_{N+1}) \in G_\delta^{N+1}$,*

$$\bar{\mathcal{E}}_N - \bar{\mathcal{E}}_{N+1} \geq 0.$$

Proof. See Appendix A.1. □

To obtain a representation suitable for implementation, we identify the grid points where the candidate sample X_{N+1} affects the k -NN regression. Define

$$G_k(X_{N+1}) = \{z \in G_\delta \mid X_{N+1} \in \{X_{(i,N+1)}(z) : i = 1, \dots, k\}\} \cup \{X_{N+1}\}. \quad (5)$$

Thus $G_k(X_{N+1})$ contains precisely the grid points whose k -nearest neighbors change after adding X_{N+1} . For all other points $z \in G_\delta \setminus G_k(X_{N+1})$, the k -NN regression remains unchanged.

The next lemma shows that the bound reduction can be written in a form that depends only on this local set.

Lemma 3.2. *For all samples $(X_1, \dots, X_N, X_{N+1})$, the error reduction $\bar{\mathcal{E}}_N - \bar{\mathcal{E}}_{N+1}$ coincides with $\frac{1}{k^2} \mu^o((X_1, \dots, X_N), X_{N+1})$, defined by*

$$\mu^o((X_1, \dots, X_N), X_{N+1}) := \sum_{z \in G_k(X_{N+1})} \mathbb{1}_{\{s(z) \geq \gamma\}} \left[\varphi(X_{(k,N)}(z), z)^2 - \varphi(X_{N+1}, z)^2 \right]$$

$$+ 2\left(\varphi(X_{(k,N)}(z), z) - \varphi(X_{N+1}, z)\right) \left(\sum_{i=1}^{k-1} \varphi(X_{(i,N)}(z), z)\right) \Big] \nu_{G_\delta}(z). \quad (6)$$

Proof. See Appendix A.1. □

This representation in terms of $\mu^o((X_1, \dots, X_N), X_{N+1}) \geq 0$ is convenient computationally. The sum is restricted to the local set $G_k(X_{N+1})$, rather than the full grid G_δ . Moreover, for fixed samples (X_1, \dots, X_N) the dependence on the candidate point X_{N+1} enters only through the terms $\varphi(X_{N+1}, z)$, which allows efficient evaluation when searching for promising sampling locations.

Step 3: Optimization of the Error Reduction

A purely exploitative strategy would select

$$X_{N+1} \in \arg \max_{y \in G_\delta} \mu^o((X_1, \dots, X_N), y),$$

i.e., a grid point that maximizes the estimated reduction of the bound. This deterministic maximization can be implemented with simulated annealing (SA), as described in Section 4.1.1.

However, $\mu^o(\cdot, \cdot)$ is derived from an upper bound on the true MSE reduction. Treating this proxy as exact information may lead to overly greedy sampling and poor performance when the bound is inaccurate. Effective sampling therefore requires a balance between exploitation of large estimated reductions and exploration of the input space.

To formalize this trade-off, we consider a family of sampling kernels proportional to

$$\mu^o((X_1, \dots, X_N), y)^\vartheta, \quad \vartheta > 0,$$

that is, a power-tempered version of the estimated error reduction. Small values of ϑ encourage exploration, while larger values concentrate probability mass near the maximizers of $\mu^o(\cdot, \cdot)$.

Section 4.1 analyzes this exploration–exploitation trade-off in more detail, and Appendix B.4 discusses practical choices of ϑ . In the main text we set $\vartheta = 1$, which corresponds to sampling proportional to the estimated error reduction (Section 4.1.2).

4 Practical Implementation

In this section, we delve deeper into the numerical implementation. First, we sketch four sampling methods for balancing exploration of the input space and exploitation of large estimated reductions. These four methods and a baseline approach relying on crude sampling from the discretized distribution ν_{G_δ} are compared in the case studies of Sections 5 and 6. Second, we address two specific implementation issues, namely the approximation of the sets $G_k(\cdot)$ and $\mathbb{1}_{\{s(z) \geq \gamma\}}$ in (6), and provide tractable solutions. Third, for the sampling kernels employed, we analyze the asymptotic approximation error of the k -NN regression function $\hat{s}_{k,N}(\cdot)$ when the number of samples N increases.

4.1 Methods for Balancing Exploration and Exploitation

4.1.1 Optimal Bound Sampling

The first method focuses on exploitation and directly optimizes the bound (6), i.e. $X_{N+1} \in \arg \max_{y \in G_\delta} \mu^o((X_1, \dots, X_N), y)$. To approximate X_{N+1} , we apply simulated annealing (SA). For a brief review on SA and details of the implementation we refer to Appendix B.1.

4.1.2 Fully Adaptive Proportional Sampling

Alternatively, emphasizing exploration, we propose sampling proportionally to the error improvement bound $\mu^o((X_1, \dots, X_N), \cdot)$ using the probability kernel

$$\mu_{N+1}^*((X_1, \dots, X_N), y) \propto \mu^o((X_1, \dots, X_N), y). \quad (7)$$

This approach eliminates the need to search on the grid for the optimal sample while still assigning higher probabilities to points that significantly reduce the bound. Moreover, the bound itself may be misleading, making pure exploitation without exploration ineffective. Since the sampling kernel is defined only up to proportionality, we use the Metropolis-Hastings algorithm to generate samples from this distribution, see Appendix B.2.

4.1.3 Semi-Adaptive Proportional Sampling

Fully adaptive proportional sampling requires updating the probability kernel at each step. To reduce computational effort, we draw instead all samples X_{N+1}, \dots, X_{N+M} for some fixed M from the kernel $\mu_{N+1}^*((X_1, \dots, X_N), \cdot)$. By construction, $X_{N+1}, \dots, X_{N+M} \in \{z \in G_\delta | s(z) \geq \gamma\}$ with probability 1. If M is small relative to N , semi-adaptive sampling should closely approximate fully adaptive sampling. The idea is to use M as the length of a moving window, recomputing the kernel after every M steps. By avoiding updates at each step, we do not systematically overlook potentially beneficial samples:

Lemma 4.1. *Let $X_1, \dots, X_{N+M} \in G_\delta$ and $y \in G_\delta$. Then*

$$\mu_{N+M+1}^*((X_1, \dots, X_{N+M}), y) > 0 \quad \text{implies} \quad \mu_{N+1}^*((X_1, \dots, X_N), y) > 0.$$

Proof. See Appendix A.2. □

4.1.4 Subset Sampling

A natural benchmark for selecting support points in k -NN regression with low MSE on $\{z \in S | s(z) \geq \gamma\}$, weighted by ν , is the probability measure ν conditioned on this subset. To enable comparison with the methods in Sections 4.1.1–4.1.3, we discretize this conditional distribution on the grid G_δ , yielding the sampling kernel

$$\mu_S^*(z) \propto \mathbb{1}_{\{s(z) \geq \gamma\}} \nu_{G_\delta}(z).$$

A detailed discussion of this subset sampling method is provided in Appendix B.3. Note that this approach does not make use of the growth condition.

4.2 Specific Implementation Issues

Implementing the SA method and the proportional sampling kernels requires the evaluation of $\mu^\circ((X_1, \dots, X_N), y)$, for $y \in G_\delta$, and this comes with two particular challenges. First, handling the complete set $G_k(y)$ is computationally expensive. In Section 4.2.1, we construct a simpler subset $H_{l_N}(y) \subseteq G_k(y)$, defined as a neighborhood of y , as a surrogate of $G_k(y)$. Second, since the black box $s(\cdot)$ is opaque, it is not possible to evaluate $\mathbb{1}_{\{s(z) \geq \gamma\}}$ during sampling. To address this issue, we introduce a sequence $(\gamma_N)_{N \in \mathbb{N}}$, with $\gamma_N \uparrow \gamma$, and replace $\mathbb{1}_{\{s(z) \geq \gamma\}}$ with $\mathbb{1}_{\{\hat{s}_{k,N}(z) \geq \gamma_N\}}$, cf. Section 4.2.2.

In our simulation case studies in Sections 5 and 6, we finally use instead of $\mu^\circ((X_1, \dots, X_N), y)$ the approximation

$$\begin{aligned} \hat{\mu}_{N+1}^{\circ,H}((X_1, \dots, X_N), y) := & \sum_{z \in H_{l_N}(y)} \mathbb{1}_{\{\hat{s}_{k,N}(z) \geq \gamma_N\}} \left[\varphi(X_{(k,N)}(z), z)^2 - \varphi(y, z)^2 \right. \\ & \left. + 2(\varphi(X_{(k,N)}(z), z) - \varphi(y, z)) \left(\sum_{i=1}^{k-1} \varphi(X_{(i,N)}(z), z) \right) \right] \nu_{G_\delta}(z). \end{aligned} \quad (8)$$

4.2.1 A Tractable Surrogate of the Set $G_k(\cdot)$

Consider the set

$$H_l(y) := \{z \in G_\delta \mid |z - y| \leq \varepsilon_l \delta\}, \quad l \in \mathbb{N}_0,$$

where ε_l is the $(l + 1)$ -th smallest element of

$$I_d = \left\{ n \in \sqrt{\mathbb{N}} \mid \exists i_1, \dots, i_d \in \mathbb{Z} : n = \sqrt{i_1^2 + \dots + i_d^2} \right\}.$$

Intuitively, $H_l(y)$ corresponds to a neighborhood of y on the grid with diameter $\varepsilon_l \delta$. For any $l \in \mathbb{N}_0$ such that $H_l(y) \subseteq G_k(y)$, replacing $G_k(y)$ in (6) by the simpler subset $H_l(y)$ provides a lower bound:

$$\begin{aligned} \mu^\circ((X_1, \dots, X_N), y) \geq & \sum_{z \in H_l(y)} \mathbb{1}_{\{s(z) \geq \gamma\}} \left[\varphi(X_{(k,N)}(z), z)^2 - \varphi(y, z)^2 \right. \\ & \left. + 2(\varphi(X_{(k,N)}(z), z) - \varphi(y, z)) \left(\sum_{i=1}^{k-1} \varphi(X_{(i,N)}(z), z) \right) \right] \nu_{G_\delta}(z). \end{aligned} \quad (9)$$

Note that the latter estimate also relies on the non-negativity of the term in the brackets, as proved in Lemma A.1 for any $z \in G_k(y)$.

Since the sets $H_l(y)$ increase in l , the deviation between $\mu^\circ((X_1, \dots, X_N), y)$ and the lower bound in (9) becomes smaller for larger l . This suggests to use the largest $l \in \mathbb{N}_0$ such that $H_l(y) \subseteq G_k(y)$ to improve the lower bound. In our implementation, however, we avoid to identify the largest l , but choose for each $y \in G_\delta$ a suitable parameter l_N such that $H_{l_N}(y) \subseteq G_k(y)$. We

emphasize that l_N in principle depends both on y and the sample (X_1, \dots, X_N) (cf. equation (5)) for $G_k(\cdot)$, but suppress this dependency to keep the notation simple. To approximate $\mu^\circ((X_1, \dots, X_N), y)$ from below, we use

$$\begin{aligned} \mu_{N+1}^{\circ, H}((X_1, \dots, X_N), y) := & \sum_{z \in H_{l_N}(y)} \mathbf{1}_{\{s(z) \geq \gamma\}} \left[\varphi(X_{(k, N)}(z), z)^2 - \varphi(y, z)^2 \right. \\ & \left. + 2(\varphi(X_{(k, N)}(z), z) - \varphi(y, z)) \left(\sum_{i=1}^{k-1} \varphi(X_{(i, N)}(z), z) \right) \right] \nu_{G_\delta}(z). \end{aligned}$$

The following lemma formulates a sufficient condition to verify that $H_l(y) \subseteq G_k(y)$. In particular, this criterion provides a strategy for determining l_N as large as possible. We employ this strategy in our case studies in Sections 5 and 6.

Lemma 4.2. *Let $y, X_1, \dots, X_N \in G_\delta$. If*

$$k \geq |H_l(y)| + d(H_l(y)) \text{ for } l \in \mathbb{N}_0,$$

where

$$d(G) = \sum_{x \in \{X_1, \dots, X_N\} \cap G} (|\{i | X_i = x\}| - 1)$$

denotes the number of duplicated samples on $G \subseteq G_\delta$, then $H_l(y) \subseteq G_k(y)$.

Proof. See Appendix A.3. □

The computation of $|H_l(\cdot)|$ is facilitated by the following result which is also used in the proof of Lemma 4.2.

Lemma 4.3. *For $y \in G_\delta$ we consider $H_l(y) = \{z \in G_\delta | |z - y| \leq \varepsilon_l \delta\}$ for a sequence $0 = \varepsilon_0 < \varepsilon_1 < \varepsilon_2 < \dots$ such that each ε_{i+1} is the smallest value such that $H_i(y)$ is a strict subset $H_{i+1}(y)$. Then ε_i is the $i+1$ -th smallest element in $I_d = \left\{ n \in \sqrt{\mathbb{N}_0} \mid \exists j_1, \dots, j_d \in \mathbb{Z} : n = \sqrt{j_1^2 + \dots + j_d^2} \right\}$, $i \in \mathbb{N}_0$. Moreover,*

$$I_1 = \mathbb{N}_0, \quad \mathbb{N}_0 \subsetneq I_2 \subsetneq I_3 \subsetneq \sqrt{\mathbb{N}_0}, \quad I_{d'} = \sqrt{\mathbb{N}_0} \text{ for } d' \geq 4.$$

Proof. See Appendix A.3. □

Remark 4.4. (i) *If $X_1, \dots, X_N \in G_\delta$ contain no duplicates, i.e., $X_i \neq X_j$ for all $i, j \in \{1, \dots, N\}$ and $X_i \neq y$ for all $i \in \{1, \dots, N\}$, then $d(H_l(y)) = 0$. Thus, for $H_l(y) \subseteq G_k(y)$ to hold, it suffices that $k \geq |H_l(y)|$, in accordance with Lemma 4.2.*

(ii) *Since $H_l(y) = \bigcup_{i=0}^l \tilde{H}_i(y)$, with $\tilde{H}_i(y) := \{z \in G_\delta | |z - y| = \varepsilon_i \delta\}$, we have*

$$|H_l(y)| = \sum_{i=0}^l |\tilde{H}_i(y)|.$$

We can write $|x - y| = \delta \sqrt{\sum_{i=1}^d j_i^2}$ for some $j_1, \dots, j_d \in \mathbb{Z}$, which may not yield a unique solution. The cardinality $|\tilde{H}_i(y)|$ represents the number of ways to express the natural

number $n := i_1^2 + \dots + i_d^2$ as the sum of d squares.

For $d = 1$, every natural number that can be expressed as a square has two representations, leading to $|\tilde{H}_i(y)| = 2$ if $i > 1$. For $d > 1$, depending on the dimension, there may be closed-form formulas for $|\tilde{H}_i(y)|$. Specifically, for $d = 2$, the formula is provided by the Sum of Two Squares Theorem; for $d = 3$, by Legendre's Three-Square Theorem; and for $d = 4$, by Jacobi's Four-Square Theorem. For further details and precise conditions characterizing these numbers, we refer to Hardy & Wright (2009).

4.2.2 The Support of $\mu^\circ(\cdot, \cdot)$

Computing $\mu_N^{\circ, H}(\cdot, \cdot)$ requires an evaluation of the indicator function $\mathbb{1}_{\{s(y) \geq \gamma\}}$. This, in turn, relies on the knowledge of $s(y)$ for all $y \in G_\delta$, but such exhaustive knowledge of $s(y)$ is not available for the black-box function. Naively substituting $\mathbb{1}_{\{\hat{s}_{k, N}(y) \geq \gamma\}}$ as estimator in place of $\mathbb{1}_{\{s(y) \geq \gamma\}}$, using the k -NN regression $\hat{s}_{k, N}(\cdot)$ based on the previous N samples, could result in the situation where $\{y \in G_\delta | \hat{s}_{k, N}(y) \geq \gamma\} \subsetneq \{y \in G_\delta | s(y) \geq \gamma\}$. In this case, values in $\{y \in G_\delta | s(y) \geq \gamma\} \setminus \{y \in G_\delta | \hat{s}_{k, N}(y) \geq \gamma\}$ would have weight 0 and would thus never be generated in the introduced sampling methods.

To address this issue, we introduce a sequence $(\gamma_N)_{N \in \mathbb{N}}$ with $\gamma_N \uparrow \gamma$. By choosing γ_N sufficiently small, we ensure that

$$\{y \in G_\delta | s(y) \geq \gamma\} \setminus \{y \in G_\delta | \hat{s}_{k, N}(y) \geq \gamma_N\}$$

is a sufficiently small set. Fixing this sequence $(\gamma_N)_{N \in \mathbb{N}}$, we use $\hat{\mu}_{N+1}^{\circ, H}((X_1, \dots, X_N), \cdot)$ defined in (8) to approximate $\mu_{N+1}^\circ((X_1, \dots, X_N), \cdot)$.

Remark 4.5. (i) For small sample sizes N , the k -NN regression $\hat{s}_{k, N}(\cdot)$ may be inaccurate or even constant if $N = k$. To address this, it can be beneficial to draw N_0 pivot samples from ν_{G_δ} , before drawing the remaining samples from $\hat{\mu}_{N+1}^{\circ, H}((X_1, \dots, X_N), \cdot)$, where $N \geq N_0$. In the implementations discussed in Sections 5 and 6, we will employ this strategy by defining a specified number of pivot samples prior to utilizing the proposed kernels.

(ii) When implementing $\hat{\mu}_N^{\circ, H}(\cdot, \cdot)$, it is usually impossible to guarantee that the assumption $\{y \in G_\delta | s(y) \geq \gamma\} \subseteq \{y \in G_\delta | \hat{s}_{k, N}(y) \geq \gamma_N\}$ holds for all $N \in \mathbb{N}$ with probability 1 with respect to ν_{G_δ} , particular without prior knowledge of $s(\cdot)$. Consequently, there might be cases where the support of $\hat{\mu}_N^{\circ, H}(\cdot, \cdot)$ is too small, leading to the omission of potential samples within $\{y \in G_\delta | s(y) \geq \gamma\}$ by the sampling distribution. To address this, the sequence γ_N should be chosen to gradually increase towards γ , thereby reducing the probability of missing samples in $\{y \in G_\delta | s(y) \geq \gamma\}$.

4.3 Asymptotic Error under the Sampling Kernels

To analyze the asymptotic behavior of the k -NN regression for an increasing number of samples N under the sampling kernels introduced in Sections 4.1.2, 4.2.1, and 4.2.2, we state the following proposition:

Proposition 4.6. *We assume that $(X_i)_{i \in \mathbb{N}}$ is generated according to the distribution \mathbf{P} defined in (2), and that one of the following assumptions holds for the stochastic kernels $\mu_{N+1} : S^N \times \mathcal{S} \rightarrow [0, 1]$, which define the conditional distribution of X_{N+1} given the previous samples X_1, \dots, X_N .*

- (i) *Let $\mu_{N+1}(\cdot, \cdot) \propto \mu_{N+1}^\circ(\cdot, \cdot)$, if $N \geq k$ and $\mu_{N+1}(\cdot, \cdot) = \nu_{G_\delta}$, if $N < k$.*
- (ii) *Let $\mu_{N+1}(\cdot, \cdot) \propto \mu_{N+1}^{\circ, H}(\cdot, \cdot)$, if $N \geq k$ and $\mu_{N+1}(\cdot, \cdot) = \nu_{G_\delta}$, if $N < k$. In addition, let $(l_N)_{N \in \mathbb{N}}$ be a sequence such that $H_{l_N}(y) \subseteq G_k(y)$ for all N .*
- (iii) *Let $\mu_{N+1}(\cdot, \cdot) \propto \hat{\mu}_{N+1}^{\circ, H}(\cdot, \cdot)$, if $N \geq k$ and $\mu_{N+1}(\cdot, \cdot) = \nu_{G_\delta}$, if $N < k$. Moreover, let $(l_N)_{N \in \mathbb{N}}$ be a sequence such that $H_{l_N}(y) \subseteq G_k(y)$ for all N and $(\gamma_N)_{N \in \mathbb{N}}$ be a sequence with $\gamma_N \uparrow \gamma$ such that $\{z \in G_\delta | s(z) \geq \gamma\} \subseteq \{z \in G_\delta | \hat{s}_{k, N}(z) \geq \gamma_N\}$ for every $N \in \mathbb{N}$.*

Then, with probability 1 under \mathbf{P} ,

$$\lim_{N \rightarrow \infty} |\hat{s}_{k, N}(y) - s(y)| = 0, \quad \text{for all } y \in \{z \in G_\delta | s(z) \geq \gamma\}.$$

Moreover, if the following additional assumptions hold in the corresponding cases,

- (i)/(ii) $|\{z \in G_\delta | s(z) \geq \gamma\}| < \infty$
- (iii) $|G_\delta| < \infty$

then there exists $N' \in \mathbb{N}$ such that, with probability 1 under \mathbf{P} , $|\hat{s}_{k, N'}(y) - s(y)| = 0$ for all $y \in \{z \in G_\delta | s(z) \geq \gamma\}$. Here, $N' \geq k |\{x \in G_\delta | s(x) \geq \gamma\}|$. If $M' > N'$, then $\mu_{M'}((X_1, \dots, X_M), \cdot)$ becomes degenerate.

Proof. See Appendix A.4. □

5 Functional Regression Case Studies

In this section we apply the sampling kernels of Section 4 to simple transformations of random variables. We compare the methods and study how the accuracy of the growth condition $\varphi(\cdot, \cdot)$ affects the performance of k -NN regression.

5.1 Simulation Design

Linear function of a uniform random variable

We first consider the black-box function $s_1(z) = z/100 - 9.5$ for $z \in (0, 1000)$ with input $Z \sim \text{Unif}(0, 1000)$. With $\delta = 0.01$ we use the grid $G_\delta = \{0, 0.01, 0.02, \dots, 1000\}$ and threshold $\gamma_1 = 0$, so the regression error is evaluated on $\{z \in G_\delta : s_1(z) \geq 0\} = \{z \in G_\delta : z \geq 950\}$. We take the exact growth condition

$$\varphi_1^1(x, y) = |s_1(x) - s_1(y)| = \frac{1}{100} |x - y|,$$

and compare it with progressively less accurate alternatives,

$$\varphi_1^2(x, y) = \frac{1}{10} |x - y|, \quad \varphi_1^3(x, y) = \frac{1}{10} (x^2 + y^2 + 1), \quad \varphi_1^4(x, y) = \frac{1}{10} (|x| + |y|).$$

Nonlinear function of a normal random variable

We next consider $s_2(z) = (1 + z^2)^{-1}$ on \mathbb{R} with input $Z \sim \mathcal{N}(-3, 1)$. With $\delta = 0.001$ we use the grid $G_\delta = \{0, 0.001, 0.002, \dots, 1\}$ on $[0, 1]$. We set $\gamma_2 = 0.3$ and evaluate the MSE on

$$\{z \in G_\delta : s_2(z) \geq 0.3\} = \left\{z \in G_\delta : z \in (-\sqrt{7/3}, \sqrt{7/3})\right\}.$$

As growth conditions we use the exact form

$$\varphi_2^1(x, y) = |s_2(x) - s_2(y)| = \frac{|x^2 - y^2|}{(1 + x^2)(1 + y^2)},$$

and the following less accurate alternatives,

$$\varphi_2^2(x, y) = |x^2 - y^2|, \quad \varphi_2^3(x, y) = 10|x^2 - y^2|, \quad \varphi_2^4(x, y) = x^2 + y^2 + 1, \quad \varphi_2^5(x, y) = |x| + |y|.$$

χ^2 distribution

For $z = (z_1, z_2, z_3) \in \mathbb{R}^3$ we consider the black-box function $s_3(z) = z_1^2 + z_2^2 + z_3^2$. If the random input $Z = (Z_1, Z_2, Z_3)$ has i.i.d. components $Z_i \sim \mathcal{N}(0, 1)$, then $s_3(Z) \sim \chi_3^2$. With $\delta = 0.1$ we use the grid

$$G_\delta = \{(\delta i_1, \delta i_2, \delta i_3) : i_1, i_2, i_3 \in \mathbb{Z}\} \subseteq \mathbb{R}^3,$$

set $\gamma_3 = 10$, and evaluate the MSE on $\{z \in G_\delta : s_3(z) \geq 10\}$. We consider the growth conditions

$$\begin{aligned} \varphi_3^1(x, y) &= |s_3(x) - s_3(y)|, & \varphi_3^2(x, y) &= 10|s_3(x) - s_3(y)|, \\ \varphi_3^3(x, y) &= \sum_{i=1}^3 |x_i^2 - y_i^2|, & \varphi_3^4(x, y) &= \sum_{i=1}^3 (x_i^2 + y_i^2) + 1. \end{aligned}$$

Remark 5.1. *The functions $\varphi_1^3(\cdot)$, $\varphi_1^4(\cdot)$, $\varphi_2^4(\cdot)$, $\varphi_2^5(\cdot)$, and $\varphi_3^4(\cdot)$ do not satisfy condition (iii) of the growth-condition definition in Section 2. We nevertheless include them to illustrate how poorly chosen growth conditions can degrade performance.*

Implementation

To compare efficiency we implement five approaches: crude sampling from ν_{G_δ} , subset sampling (Section 4.1.4), semi-adaptive sampling (Section 4.1.3), fully adaptive sampling (Section 4.1.2), and simulated annealing (SA; Section 4.1.1). Each method draws 500 samples. We set $k = 5$ in the first two case studies and $k = 7$ in the χ^2 case. We use $l_N = 2$ when no duplicate samples occur; otherwise l_N is adjusted within the algorithms according to Lemma 4.2.

For SA, adaptive, and subset sampling we first draw $N_0 = 100$ pivot samples from ν_{G_δ} . For the estimated support $\{z \in G_\delta : \hat{s}_{k,N}(z) \geq \gamma_{i,N}\}$ with $N \geq 100$ (Section 4.2.2), we set

$$\gamma_{1,N} = -\frac{1}{2} + \frac{1}{30} \left\lfloor \frac{N - 100}{25} \right\rfloor, \quad \gamma_{2,N} = \frac{1}{50} \left\lfloor \frac{N - 100}{25} \right\rfloor, \quad \gamma_{3,N} = 5 + \frac{5}{16} \left\lfloor \frac{N - 100}{25} \right\rfloor,$$

so that $\gamma_{i,N}$ increases every $N_1 = 25$ samples. In the SA and subset methods, the support of $\hat{\mu}_N^{\circ,H}(\cdot, \cdot)$ is updated every N_1 samples; in the semi-adaptive method, the sampling kernel

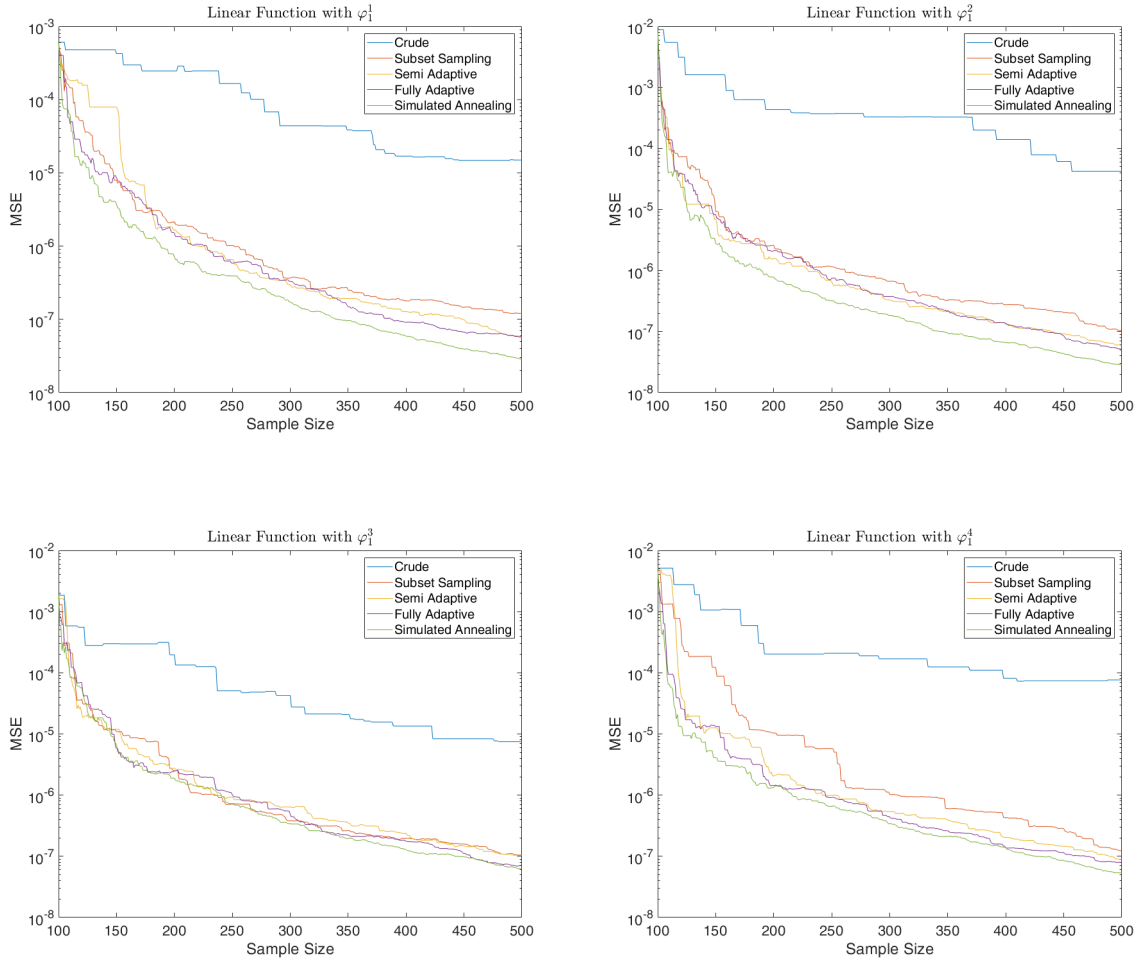


Figure 1: Logarithmic-scale MSE versus sample size for the k -NN regression of the linear function $s_1(\cdot)$ with uniform inputs comparing the five discussed methods.

is updated at the same frequency to reflect the change in $\gamma_{i,N}$. Thus, for $i \in \{1, \dots, 16\}$, the samples

$$X_{N_0+iN_1+1}, \dots, X_{N_0+(i+1)N_1}$$

are drawn from $\hat{\mu}_{N_0+iN_1}^{*,H}(\cdot, \cdot)$. In the fully adaptive method the kernel is updated after each sample, so $X_{N_0+i} \sim \hat{\mu}_{N_0+i-1}^{*,H}(\cdot, \cdot)$ for $i \in \{1, \dots, 400\}$. Implementation details are given in Appendices B.1 and B.2.

5.2 Results

The simulation results are summarized in Figures 1, 2, and 3, which plot the k -NN regression MSE (log scale) against sample size.

For the linear case (Figure 1), the MSE decreases with sample size for all five methods. For every growth condition, crude sampling performs worst. With $\varphi_1^1(\cdot)$ and $\varphi_1^2(\cdot)$, SA attains the lowest MSE at larger N , with the largest gap relative to crude sampling occurring at $N = 500$. With $\varphi_1^3(\cdot)$ and $\varphi_1^4(\cdot)$, SA, semi-adaptive, and fully adaptive sampling perform similarly. Overall, differences among subset sampling, semi-adaptive, fully adaptive, and SA are small, with a mild ordering from subset to semi-adaptive to fully adaptive to SA.

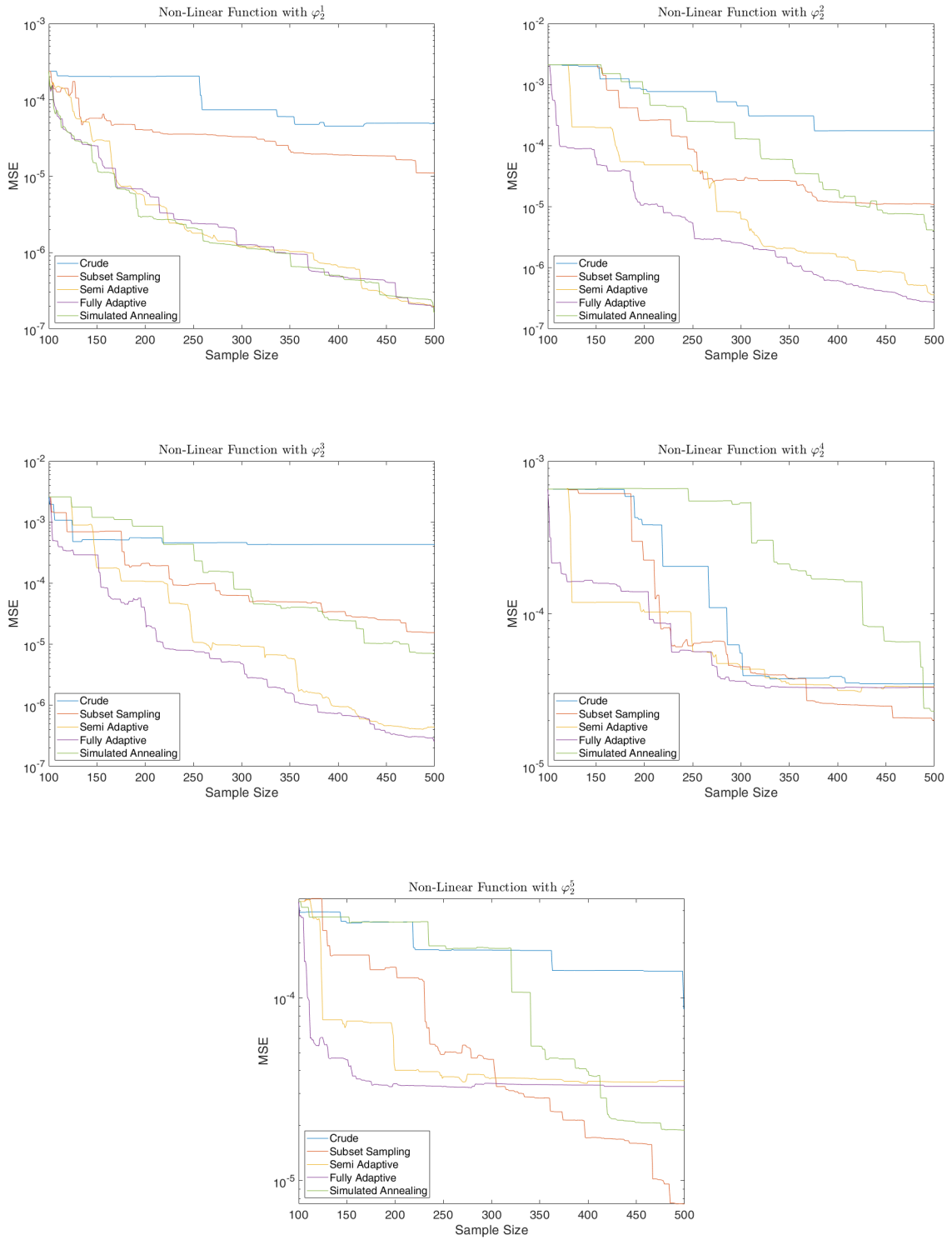


Figure 2: Logarithmic-scale MSE versus sample size for the k -NN regression of the non-linear function $s_2(\cdot)$ with normal inputs comparing the five discussed methods.

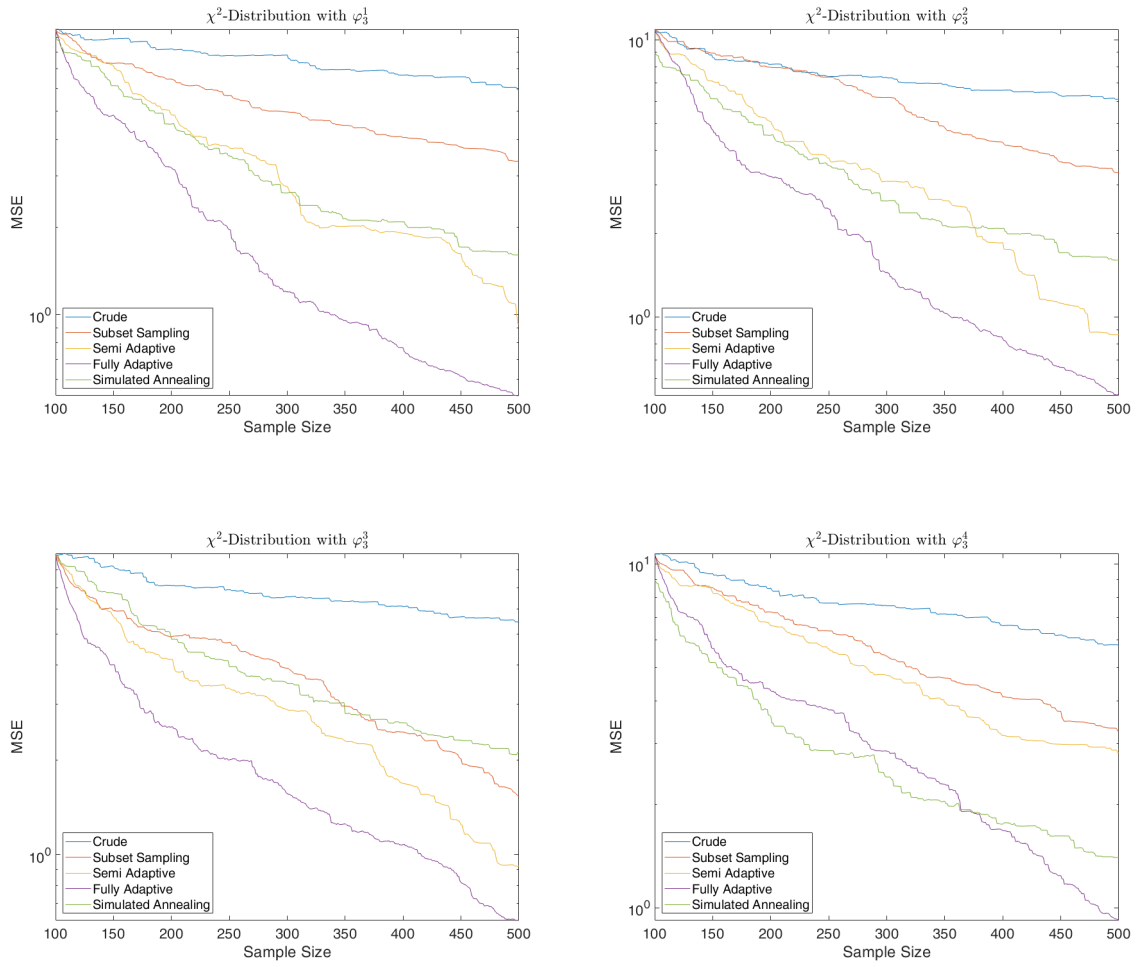


Figure 3: Logarithmic-scale MSE versus sample size for the k -NN regression of the χ^2 -distribution $s_3(\cdot)$ comparing the five discussed methods.

For the nonlinear case (Figure 2), the MSE decreases with sample size for all methods. The performance of SA varies strongly with the growth condition and with N . For $\varphi_2^1(\cdot)$, $\varphi_2^2(\cdot)$, and $\varphi_2^3(\cdot)$, semi-adaptive and fully adaptive sampling substantially improve on crude and subset sampling. For $\varphi_2^4(\cdot)$, SA performs worst except at very large N , while the other four methods are similar. For $\varphi_2^5(\cdot)$, SA and subset sampling yield only small gains over the adaptive methods at larger N . Overall, SA is effective only when the growth condition is accurate, whereas semi-adaptive and fully adaptive sampling perform well and remain stable across growth conditions and sample sizes.

For the χ^2 case (Figure 3), the MSE decreases with sample size for all methods. For every growth condition, crude sampling performs worst, while the fully adaptive method is best overall. For $\varphi_3^1(\cdot)$ and $\varphi_3^2(\cdot)$, fully adaptive sampling dominates, with SA and semi-adaptive sampling close and subset sampling worse. For $\varphi_3^3(\cdot)$, SA and subset sampling are similar but both underperform semi-adaptive sampling. For $\varphi_3^4(\cdot)$, SA is better at small N but is overtaken as N grows and ultimately yields a higher MSE. Overall, fully adaptive sampling is strong and stable across growth conditions in this example.

Overall, the fully adaptive method shows the most robust performance across growth conditions and sample sizes. When the growth condition is sufficiently accurate, the SA method can outperform the others, but its performance deteriorates under misspecification or small sample sizes.

6 ALM Model Regression Study

Asset-liability management (ALM) concerns the joint management of assets, liabilities, and cash flows, with particular emphasis on the risk of changes in net asset value (NAV). This requires stochastic models for the joint evolution of assets and liabilities, which are often complex and analytically intractable. In this section, we apply the sampling methods from Section 4 to a simplified ALM model.

6.1 Model Description

We consider a one-period model for the net asset value (NAV) of a non-life insurer, following Becker et al. (2014); Hamm, Knispel & Weber (2020). Let A_t , L_t , and E_t denote the book values of assets, liabilities, and NAV at times $t \in \{0, 1\}$, with balance-sheet identity

$$A_t = L_t + E_t, \quad t \in \{0, 1\}.$$

We assume that liabilities are constant over the period and equal to a claims reserve v , so $L_t = v$ for $t \in \{0, 1\}$. At time 0, the insurer receives a premium π . Claims are settled at time 1 and are modeled by the collective risk model

$$C = \sum_{k=1}^N \xi_k,$$

where $N \in \mathbb{N}$ is the claim count and $\xi_k \geq 0$ are the individual claim sizes.

On the asset side, the insurer invests in two financial instruments: a risky asset S_t and a

bond B_t , both normalized to $S_0 = B_0 = 1$. We assume zero interest rates, so $B_1 = 1$, and model the risky asset by

$$S_1 = \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma\sqrt{\Delta t}Z\right),$$

where $\mu \in \mathbb{R}$, $\Delta t > 0$, $\sigma > 0$, and $Z \sim \mathcal{N}(0, 1)$. At time 0, the insurer invests a fraction b of total assets A_0 in the risky asset and the remainder in the bond. The corresponding holdings during the one-year period ('buy and hold') are

$$\eta^S = bA_0, \quad \eta^B = (1 - b)A_0,$$

that is, the numbers of units held in S_0 and B_0 , respectively.

At time 1, the asset value is

$$A_1 = \eta^S S_1 + \eta^B B_1 = A_0(bS_1 + 1 - b),$$

and the NAV is

$$E_1 = A_1 - L_1 = A_0(bS_1 + 1 - b) - v = E_0 + (E_0 + v)b(S_1 - 1) - C + \pi.$$

In the simulation study, we consider

$$s_{\text{ALM}}(Z, C) = -(E_1 - E_0),$$

the negative change in NAV over the period.

6.2 Implementation

The model parameters are as follows. The initial NAV is $E_0 = 2000$, the risky asset has drift $\mu = 0.01$ and volatility $\sigma = 0.05$, the risky investment fraction is $b = 0.1$, and the time horizon is $\Delta t = 1$. The claim count N is Poisson with parameter $\lambda = 5$, and the claim sizes ξ_k are i.i.d. exponentially distributed with rate $\theta = 1/10$. All random variables N , $(\xi_k)_{k \in \mathbb{N}}$, and Z are independent. The premium and reserve are determined by a pure premium principle with safety loading, yielding

$$\pi = 1.03 \cdot 5 \cdot 10 = 51.5, \quad v = 4 \cdot 5 \cdot 10 = 200.$$

To define the growth conditions, we introduce the misspecified proxy

$$\tilde{s}(z, c \mid \tilde{\mu}, c_\sigma, a) := -(E_0 + v)b \left(\exp\left(\tilde{\mu} - \frac{(\sigma + c_\sigma)^2}{2} + (\sigma + c_\sigma)z\right) - 1 \right) + ac - \pi.$$

The growth condition used in the implementation is then

$$\psi(z_1, c_1, z_2, c_2 \mid \tilde{\mu}, c_\sigma, a) := \left| \max \left\{ \tilde{s}(z_1, c_1 \mid \tilde{\mu}, c_\sigma, a) - \tilde{s}(z_2, c_2 \mid \tilde{\mu}, c_\sigma, a), \right. \right.$$

$$\left. \tilde{s}(z_1, c_1 \mid \tilde{\mu}, -c_\sigma, 1/a) - \tilde{s}(z_2, c_2 \mid \tilde{\mu}, -c_\sigma, 1/a) \right\} \Big|.$$

In Section 6.3.1 we consider the following growth conditions:

$$\begin{aligned} \varphi_{\text{ALM}}^1(z_1, c_1, z_2, c_2) &= \psi(z_1, c_1, z_2, c_2 \mid \mu, 0, 1), \\ \varphi_{\text{ALM}}^2(z_1, c_1, z_2, c_2) &= \psi(z_1, c_1, z_2, c_2 \mid 4\mu, 0, 1), \\ \varphi_{\text{ALM}}^3(z_1, c_1, z_2, c_2) &= \psi(z_1, c_1, z_2, c_2 \mid \mu, 0.025, 1), \\ \varphi_{\text{ALM}}^4(z_1, c_1, z_2, c_2) &= \psi(z_1, c_1, z_2, c_2 \mid \mu, 0, 2). \end{aligned}$$

In the case studies of Section 6.3.3 we analyze growth conditions under varying misspecification. Specifically, we consider

$$\begin{aligned} \varphi_{\text{ALM}}^5(z_1, c_1, z_2, c_2) &= \psi(z_1, c_1, z_2, c_2 \mid \tilde{\mu}, 0, 1), & \tilde{\mu} &\in [-0.5, 0.5], \\ \varphi_{\text{ALM}}^6(z_1, c_1, z_2, c_2) &= \psi(z_1, c_1, z_2, c_2 \mid \mu, c_\sigma, 1), & c_\sigma &\in [0, 0.8], \\ \varphi_{\text{ALM}}^7(z_1, c_1, z_2, c_2) &= \psi(z_1, c_1, z_2, c_2 \mid \mu, 0, a), & a &\in [0, 4]. \end{aligned}$$

For the simulations, we set $\delta = 0.05$ and use a grid $G_\delta \subset \mathbb{R} \times \mathbb{R}_+$. Analogous to Section 5 all five sampling methods are implemented, each with 2100 samples. In the k -NN regression we set $k = 9$ and $l_N = 3$, reducing l_N when necessary; see Lemma 4.2. For subset sampling, semi-adaptive sampling, fully adaptive sampling, and SA, we first draw $N_0 = 100$ pivot samples from ν_{G_δ} .

We fix the target threshold at $\gamma_{\text{ALM}} = 125$ and evaluate the MSE on

$$\{(z, c) \in G_\delta : s_{\text{ALM}}(z, c) \geq 125\}.$$

The sampling support is updated through

$$\gamma_{\text{ALM},N} := 30 + 19 \left\lfloor \frac{N - 100}{200} \right\rfloor, \quad 100 \leq N \leq 1100.$$

Thus, for $100 \leq N \leq 1100$, samples are drawn from

$$\{(z, c) \in G_\delta : \hat{s}_{k,N}(z, c) \geq \gamma_{\text{ALM},N}\},$$

whereas for $N > 1100$ they are drawn from

$$\{(z, c) \in G_\delta : \hat{s}_{k,N}(z, c) \geq 125\}.$$

In the SA, semi-adaptive, and subset sampling methods, the support is updated every $M = 50$ samples. In the fully adaptive method, the kernel is updated after each additional sample. Further implementation details are given in Appendices B.1 and B.2.

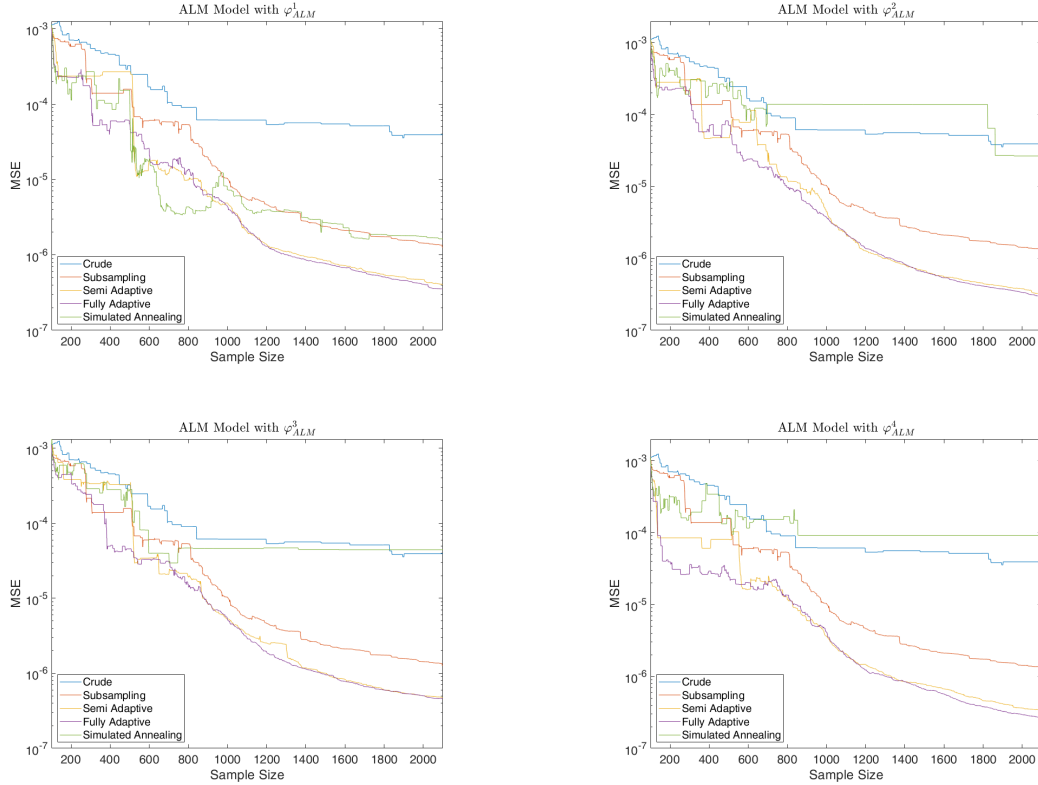


Figure 4: Logarithmic-scale MSE versus sample size for the k -NN regression of the ALM model comparing the five discussed methods.

6.3 Results

6.3.1 MSE versus the Sample Size

Figure 4 plots the MSE on a logarithmic scale against sample size for four growth conditions. Across all four cases, the crude, subset sampling, semi-adaptive, and fully adaptive methods show similar qualitative behavior. Crude sampling yields the slowest error decay, whereas semi-adaptive and fully adaptive sampling reduce the MSE substantially faster. For φ_{ALM}^1 , SA performs comparably to subset sampling; under the remaining growth conditions, however, its performance deteriorates markedly and becomes close to that of crude sampling. Overall, the semi-adaptive and fully adaptive methods are the most robust across growth conditions.

6.3.2 Sample Selection of the Algorithms

The construction of a k -NN surrogate for a complex black box depends critically on the selection of support points. The aim is to generate samples that cover the relevant regions of the input space, weighted both by the probability measure ν and by the behavior of the black box s , in particular in regions where s varies strongly.

Figure 5 compares the sample distributions generated by the fully adaptive method (using the exact growth condition φ_{ALM}^1) and the subset sampling method. The samples are shown together with contour lines of $\hat{s}_{\text{ALM}}(Z, C) \nu_{G_\delta}(Z, C)$, which indicate the target region of the approximation.

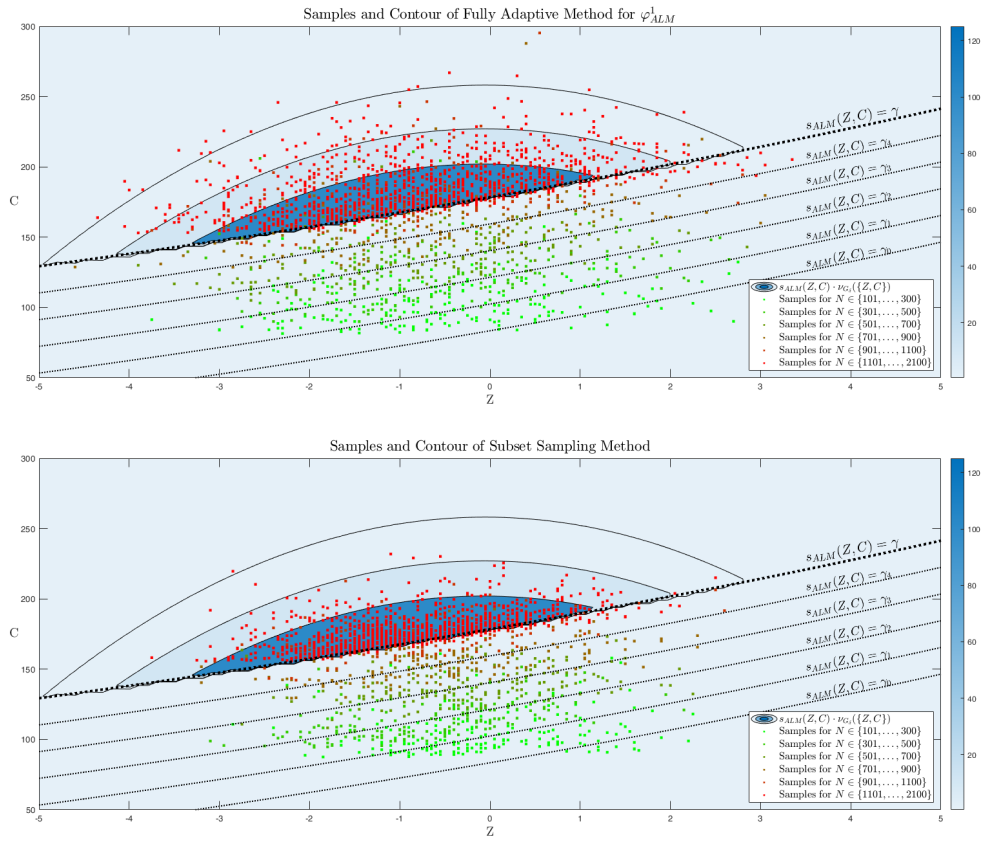


Figure 5: Comparison of samples drawn with the fully adaptive and subset sampling methods, overlaid on the contour lines of $\hat{s}_{ALM}(Z, C) \cdot \nu_{G_\delta}(\{Z, C\})$. The 2000 samples shown are taken from the case studies in Section 6.3.1, excluding pivot samples.

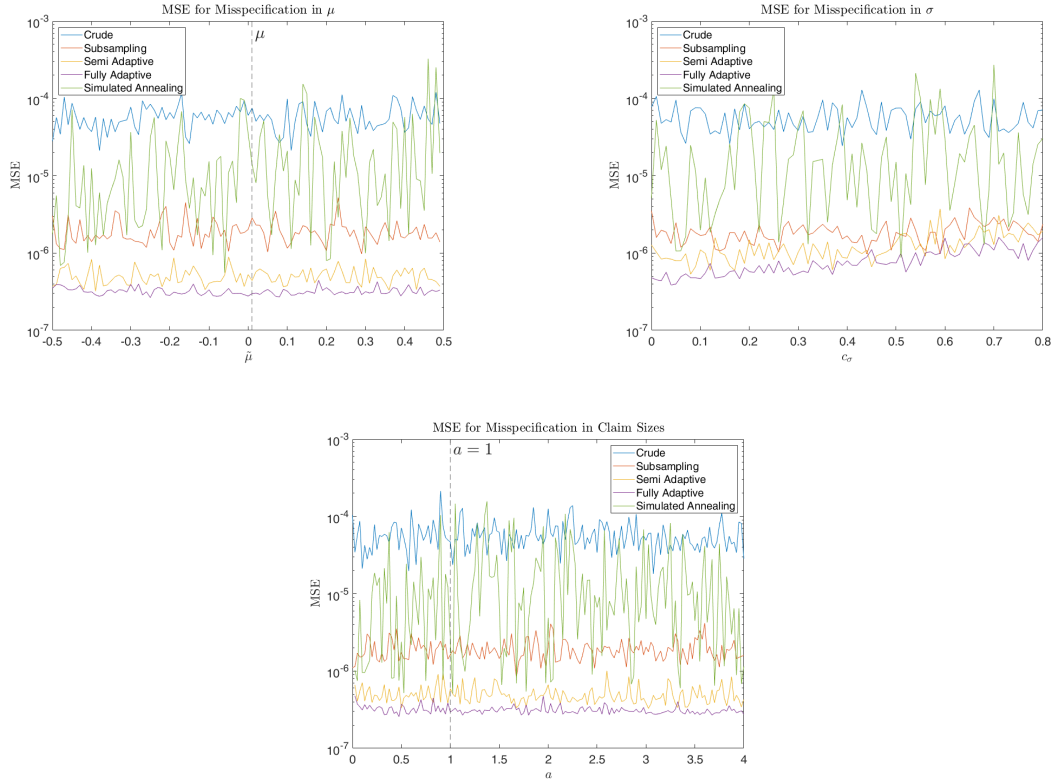


Figure 6: Logarithmic-scale MSE under varying parameter misspecifications in φ_{ALM}^2 , φ_{ALM}^3 , and φ_{ALM}^4 for the k -NN regression in the ALM model.

As Figure 5 shows, the fully adaptive method, which attains the lowest MSE at larger sample sizes, generates a more dispersed sample set while remaining better aligned with the target region. By contrast, subset sampling, which yields a higher MSE, allocates points solely according to the measure ν and does not adapt to the structure or growth behavior of the black box s . In both methods, the sampling concentrates increasingly on the relevant region $\{s(\cdot) \geq \gamma\}$ as $\gamma_N \rightarrow \gamma$.

This analysis shows that the fully adaptive method not only performs well numerically, but also selects representative scenarios for constructing an accurate k -NN surrogate under the error criterion (1). These scenarios may also help to interpret the input-output structure of the black box.

6.3.3 MSE for Parameter Misspecification

Figure 6 reports the MSE after 2100 samples for each method, plotted against the parameters defining the growth conditions; see Section 6.2. Consistent with the results of Section 6.3.1, the fully adaptive method attains the lowest MSE throughout. The semi-adaptive method performs similarly well, whereas the performance of SA varies substantially across specifications.

Some methods are sensitive to misspecification, but no systematic effect is visible for μ in the plot corresponding to φ_{ALM}^2 , even when μ is scaled by a factor of 50. By contrast, misspecification in σ , shown for φ_{ALM}^3 , leads to a clear deterioration in the semi-adaptive and

fully adaptive methods. Once the deviation in σ reaches about 0.5, their performance falls to the level of subset sampling, which is less affected. In the third plot, corresponding to inflated claim sizes under φ_{ALM}^4 , the average performance remains broadly stable across methods, although variability increases in some cases as a grows.

7 Conclusion

In this paper we developed several sampling methods for k -NN regression aimed at rare outcomes, using prior information in the form of a growth condition for the underlying black box. The goal is to construct an accurate k -NN surrogate for a complex system, with potential applications such as counterfactual analysis in machine learning. Since the number of support points in k -NN regression is necessarily limited, their placement is critical: samples should be concentrated in regions where the probability-weighted black box varies strongly in the tail, and used more sparsely where this weighted response is locally flat.

Our results show that direct maximization of the error bound is generally not optimal, reflecting the fundamental trade-off between exploitation of the proxy criterion and exploration of the input space. Among the proposed methods, the semi-adaptive and fully adaptive approaches deliver the most robust overall performance.

Future research should examine this exploration-exploitation trade-off more systematically, develop methods to estimate suitable growth conditions from data, and derive corresponding error guarantees. Improving the accuracy of support estimation for the sampling distributions also appears to be a promising direction.

References

- Alkhatib, K., H. Najadat, I. Hmeidi & M. K. A. Shatnawi (2013). “Stock Price Prediction Using K-Nearest Neighbor (kNN) Algorithm”. *International Journal of Business, Humanities and Technology* 3 (3), pp. 32–44.
- Arief, M., Y. Bai, W. Ding, S. He, Z. Huang, H. Lam & D. Zhao (2021). “Certifiable Deep Importance Sampling for Rare-Event Simulation of Black-Box Systems”. *Proceedings of the 24-th International Conference on Artificial Intelligence and Statistics*. Vol. 130, pp. 595–603.
- Asmussen, S. & P. W. Glynn (2007). *Stochastic Simulation: Algorithms and Analysis*. 1st ed. Springer.
- Becker, T., C. Cottin, M. Fahrenwaldt, A. M. Hamm, S. Nörtemann & S. Weber (2014). “Market Consistent Embedded Value—Eine Praxisorientierte Einführung”. *Der Aktuar* 1, pp. 4–8.
- Beyer, K., J. Goldstein, R. Ramakrishnan & U. Shaft (1999). “When Is “Nearest Neighbor” Meaningful?” *Database Theory—ICDT’99*. Vol. 1540. Springer Berlin, pp. 217–235.
- Bhatia, N. (2010). “Survey of Nearest Neighbor Techniques”. *arXiv: Computer Vision and Pattern Recognition*.
- Bouttier, C. & I. Gavra (2019). “Convergence Rate of a Simulated Annealing Algorithm with Noisy Observations”. *Journal of Machine Learning Research* 20 (4), pp. 1–45.
- Bucklew, J. A. (2004). *Introduction to Rare Event Simulation*. 5th ed. Springer New York.
- Černý, V. (1985). “Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm”. *Journal of Optimization Theory and Applications* 45, pp. 41–51.
- Cérou, F., P. Del Moral, T. Furon & A. Guyader (2012). “Sequential Monte Carlo for Rare Event Estimation”. *Statistics and Computing* 22 (3), pp. 795–808.
- Chen, Y. & Y. Hao (2017). “A Feature Weighted Support Vector Machine and K-Nearest Neighbor Algorithm for Stock Market Indices Prediction”. *Expert Systems with Applications* 80, pp. 340–355.
- Chopin, N., O. Papaspiliopoulos, et al. (2020). *An Introduction to Sequential Monte Carlo*. 1st ed. Springer.
- Corso, A., R. Moss, M. Koren, R. Lee & M. Kochenderfer (2021). “A Survey of Algorithms for Black-Box Safety Validation of Cyber-Physical Systems”. *Journal of Artificial Intelligence Research* 72, pp. 377–428.
- Cover, T. & P. Hart (1967). “Nearest Neighbor Pattern Classification”. *Transactions on Information Theory* 13 (1), pp. 21–27.
- Del Moral, P., A. Doucet & A. Jasra (2006). “Sequential Monte Carlo Samplers”. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 68 (3), pp. 411–436.

- Delahaye, D., S. Chaimatanan & M. Mongeau (2018). “Simulated Annealing: From Basics to Applications”. *Handbook of Metaheuristics*. 2nd ed. Springer, pp. 1–35.
- Devroye, L., L. Györfi & G. Lugosi (2013). *A Probabilistic Theory of Pattern Recognition*. 1st ed. Springer.
- Dhanabal, S. & S. Chandramathi (2011). “A Review of Various K-Nearest Neighbor Query Processing Techniques”. *International Journal of Computer Applications* 31 (7), pp. 14–22.
- Doucet, A., N. De Freitas & N. J. Gordon (2001). *Sequential Monte Carlo Methods in Practice*. Vol. 1. 2. Springer.
- Fix, E. & J. L. Hodges (1952). “Discriminatory Analysis: Nonparametric Discrimination: Small Sample Performance”. *USAF School of Aviation Medicine*.
- Franzin, A. & T. Stützle (2019). “Revisiting Simulated Annealing: A Component-Based Analysis”. *Computers & Operations Research* 104, pp. 191–206.
- Guilmeau, T., E. Chouzenoux & V. Elvira (2021). “Simulated Annealing: A Review and a New Scheme”. *2021 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, pp. 101–105.
- Györfi, L., M. Kohler, A. Krzyżak & H. Walk (2002). *A Distribution-Free Theory of Nonparametric Regression*. 1st ed. Springer.
- Hajek, B. (1988). “Cooling Schedules for Optimal Annealing”. *Mathematics of Operations Research* 13 (2), pp. 311–329.
- Hamm, A. M., T. Knispel & S. Weber (2020). “Optimal Risk Sharing in Insurance Networks: An Application to Asset–Liability Management”. *European Actuarial Journal* 10 (1), pp. 203–234.
- Hardy, G. H. & E. M. Wright (2009). *An Introduction to the Theory of Numbers*. 6th ed. Oxford University Press.
- Hastie, T., R. Tibshirani & J. H. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. Springer.
- Henderson, D., S. H. Jacobson & A. W. Johnson (2003). “The Theory and Practice of Simulated Annealing”. *Handbook of Metaheuristics*, pp. 287–319.
- Huang, J., J. Chai & S. Cho (2020). “Deep Learning in Finance and Banking: A Literature Review and Classification”. *Frontiers of Business Research in China* 14 (1), pp. 1–24.
- Huang, Z., H. Lam & D. Zhao (2018). “Rare-Event Simulation without Structural Information: A Learning-Based Approach”. *2018 Winter Simulation Conference*, pp. 1826–1837.
- Javidrad, F. & M. Nazari (2017). “A New Hybrid Particle Swarm and Simulated Annealing Stochastic Optimization Method”. *Applied Soft Computing* 60, pp. 634–654.
- Jiang, L., Z. Cai, D. Wang & S. Jiang (2007). “Survey of Improving k-Nearest-Neighbor for Classification”. *Fourth International Conference on Fuzzy Systems and Knowledge Discovery*. Vol. 1, pp. 679–683.

- Juneja, S. & P. Shahabuddin (2006). “Rare-Event Simulation Techniques: An Introduction and Recent Advances”. *Handbooks in Operations Research and Management Science* 13, pp. 291–350.
- Kallenberg, O. (2002). *Foundations of modern probability*. 2nd ed. Springer.
- Kirkpatrick, S. (1984). “Optimization by Simulated Annealing: Quantitative Studies”. *Journal of Statistical Physics* 34, pp. 975–986.
- Klenke, A. (2020). *Probability Theory: a Comprehensive Course*. 3rd ed. Springer.
- Kwon, O. W. & J. H. Lee (2000). “Web Page Classification Based on K-Nearest Neighbor Approach”. *Proceedings of the Fifth International Workshop on Information Retrieval with Asian Languages*, pp. 9–15.
- L’Ecuyer, P., M. Mandjes & B. Tuffin (2009). “Importance Sampling in Rare Event Simulation”. *Rare Event Simulation Using Monte Carlo Methods*. Wiley & Sons, pp. 17–38.
- Liang, F., Y. Cheng & G. Lin (2014). “Simulated Stochastic Approximation Annealing for Global Optimization with a Square-Root Cooling Schedule”. *Journal of the American Statistical Association* 109 (506), pp. 847–863.
- Mohri, M., A. Rostamizadeh & A. Talwalkar (2018). *Foundations of Machine Learning*. 1st ed. MIT Press.
- Nourani, Y. & B. Andresen (1998). “A Comparison of Simulated Annealing Cooling Strategies”. *Journal of Physics A: Mathematical and General* 31 (41), 8373–8385.
- Robert, C. P. & G. Casella (2004). *Monte Carlo Statistical Methods*. 2nd ed. Springer.
- Shalev-Shwartz, S. & S. Ben-David (2014). *Understanding Machine Learning: From Theory to Algorithms*. 1st ed. Cambridge University Press.
- Tajmouati, S., B. E. Wahbi, A. Bedoui, A. Abarda & M. Dakkon (2021). “Applying K-Nearest Neighbors to Time Series Forecasting: Two New Approaches”. *Journal of Forecasting* 43 (5).
- Taunk, K., S. De, S. Verma & A. Swetapadma (2019). “A Brief Review of Nearest Neighbor Algorithm for Learning and Classification”. *International Conference on Intelligent Computing and Control Systems*, pp. 1255–1260.
- Van Laarhoven, P. J. & E. H. Aarts (1987). *Simulated Annealing*. 1st ed. Springer.
- Wu, X., V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, et al. (2008). “Top 10 Algorithms in Data Mining”. *Knowledge and Information Systems* 14, pp. 1–37.
- Zuev, K. (2015). “Subset Simulation Method for Rare Event Estimation: An Introduction”. *arXiv preprint arXiv:1505.03506*.

Data Availability

No external data sets were analyzed during the current study. The suggested algorithms were numerically tested, as described in the paper.

Declaration on the Use of AI Tools

This paper reports original research conducted by the authors. AI-based language tools were used for language editing and clarity; they did not contribute to the substantive content, analysis, or conclusions.

A Proofs

A.1 Proofs of Section 3

Proof of Lemma 3.1: For $1 \leq i \leq k$, the i -th nearest neighbor of any query point $z \in G_\delta$ satisfies $|X_{(i,N+1)}(z) - z| \leq |X_{(i,N)}(z) - z|$. Since the growth condition $\varphi(\cdot, \cdot)$ is assumed to be both symmetric and monotone w.r.t. the Euclidean distance, this yields $\varphi(X_{(i,N+1)}(z), z) \leq \varphi(X_{(i,N)}(z), z)$ for any $i = 1, \dots, k$, hence

$$\left(\sum_{i=1}^k \varphi(X_{(i,N)}(z), z) \right)^2 \geq \left(\sum_{i=1}^k \varphi(X_{(i,N+1)}(z), z) \right)^2 \quad (10)$$

and $\bar{\mathcal{E}}_N - \bar{\mathcal{E}}_{N+1} \geq 0$. □

Proof of Lemma 3.2: The set $G_k(X_{N+1})$ defined in (5) contains exactly the grid points where X_{N+1} is among the k -nearest neighbors, i.e., the k nearest neighbors are affected by the additional sample X_{N+1} only for points in $G_k(X_{N+1})$, while they remain unchanged for those in $G_\delta \setminus G_k(X_{N+1})$. With these two sets we obtain the decomposition

$$\begin{aligned} \bar{\mathcal{E}}_{N+1} &= \frac{1}{k^2} \sum_{z \in G_\delta} \mathbb{1}_{\{s(z) \geq \gamma\}} \left(\sum_{i=1}^k \varphi(X_{(i,N+1)}(z), z) \right)^2 \nu_{G_\delta}(z) \\ &= \frac{1}{k^2} \sum_{z \in G_k(X_{N+1})} \mathbb{1}_{\{s(z) \geq \gamma\}} \left(\sum_{i=1}^k \varphi(X_{(i,N+1)}(z), z) \right)^2 \nu_{G_\delta}(z) \\ &\quad + \frac{1}{k^2} \sum_{z \in G_\delta \setminus G_k(X_{N+1})} \mathbb{1}_{\{s(z) \geq \gamma\}} \left(\sum_{i=1}^k \varphi(X_{(i,N)}(z), z) \right)^2 \nu_{G_\delta}(z), \end{aligned}$$

hence

$$\begin{aligned} \bar{\mathcal{E}}_N - \bar{\mathcal{E}}_{N+1} &= \frac{1}{k^2} \sum_{z \in G_\delta} \mathbb{1}_{\{s(z) \geq \gamma\}} \left(\sum_{i=1}^k \varphi(X_{(i,N)}(z), z) \right)^2 \nu_{G_\delta}(z) \\ &\quad - \frac{1}{k^2} \sum_{z \in G_k(X_{N+1})} \mathbb{1}_{\{s(z) \geq \gamma\}} \left(\sum_{i=1}^k \varphi(X_{(i,N+1)}(z), z) \right)^2 \nu_{G_\delta}(z) \\ &\quad - \frac{1}{k^2} \sum_{z \in G_\delta \setminus G_k(X_{N+1})} \mathbb{1}_{\{s(z) \geq \gamma\}} \left(\sum_{i=1}^k \varphi(X_{(i,N)}(z), z) \right)^2 \nu_{G_\delta}(z) \\ &= \frac{1}{k^2} \sum_{z \in G_k(X_{N+1})} \mathbb{1}_{\{s(z) \geq \gamma\}} \left(\sum_{i=1}^k \varphi(X_{(i,N)}(z), z) \right)^2 - \left(\sum_{i=1}^k \varphi(X_{(i,N+1)}(z), z) \right)^2 \nu_{G_\delta}(z) \\ &= \frac{1}{k^2} \sum_{z \in G_k(X_{N+1})} \mathbb{1}_{\{s(z) \geq \gamma\}} \left[\varphi(X_{(k,N)}(z), z)^2 - \varphi(X_{N+1}, z)^2 \right. \\ &\quad \left. + 2(\varphi(X_{(k,N)}(z), z) - \varphi(X_{N+1}, z)) \left(\sum_{i=1}^{k-1} \varphi(X_{(i,N)}(z), z) \right) \right] \nu_{G_\delta}(z) \\ &= \frac{1}{k^2} \mu^o((X_1, \dots, X_N), X_{N+1}). \end{aligned}$$

The penultimate identity is justified by the subsequent Lemma A.1 □

Lemma A.1. *For all $z \in G_k(X_{N+1})$, we obtain*

$$\begin{aligned} 0 &\leq \left(\sum_{i=1}^k \varphi(X_{(i,N)}(z), z) \right)^2 - \left(\sum_{i=1}^k \varphi(X_{(i,N+1)}(z), z) \right)^2 \\ &= \varphi(X_{(k,N)}(z), z)^2 - \varphi(X_{N+1}, z)^2 + 2(\varphi(X_{(k,N)}(z), z) - \varphi(X_{N+1}, z)) \left(\sum_{i=1}^{k-1} \varphi(X_{(i,N)}(z), z) \right). \end{aligned}$$

Proof. Let us fix $z \in G_k(X_{N+1})$. Non-negativity holds due to equation (10) in the proof of Lemma 3.1. For the second identity, note that

$$\begin{aligned} &\left(\sum_{i=1}^k \varphi(X_{(i,N)}(z), z) \right)^2 - \left(\sum_{i=1}^k \varphi(X_{(i,N+1)}(z), z) \right)^2 \\ &= \sum_{i=1}^k \varphi(X_{(i,N)}(z), z)^2 + 2 \sum_{\substack{i,j=1 \\ i < j}}^k \varphi(X_{(i,N)}(z), z) \varphi(X_{(j,N)}(z), z) \\ &\quad - \sum_{i=1}^k \varphi(X_{(i,N+1)}(z), z)^2 - 2 \sum_{\substack{i,j=1 \\ i < j}}^k \varphi(X_{(i,N+1)}(z), z) \varphi(X_{(j,N+1)}(z), z) \\ &= \underbrace{\sum_{i=1}^k \varphi(X_{(i,N)}(z), z)^2 - \varphi(X_{(i,N+1)}(z), z)^2}_{(i)} \\ &\quad + 2 \underbrace{\sum_{\substack{i,j=1 \\ i < j}}^k \varphi(X_{(i,N)}(z), z) \varphi(X_{(j,N)}(z), z) - \varphi(X_{(i,N+1)}(z), z) \varphi(X_{(j,N+1)}(z), z)}_{(ii)}. \end{aligned}$$

For $z \in G_k(X_{N+1})$, the additional sample X_{N+1} belongs to the k nearest neighbors. Thus the k -NN from the initial N samples, denoted as $X_{(1,N)}(z), \dots, X_{(k,N)}(z)$, and the k -NN from the expanded set of $N + 1$ samples, denoted as $X_{(1,N+1)}(z), \dots, X_{(k,N+1)}(z)$, share exactly $k - 1$ nearest neighbors, i.e.,

$$\{X_{(1,N+1)}(z), \dots, X_{(k,N+1)}(z)\} = \{X_{(1,N)}(z), \dots, X_{(k-1,N)}(z), X_{N+1}\}.$$

Based on this observation, we conclude that

$$\begin{aligned} (i) \quad &\sum_{i=1}^k \varphi(X_{(i,N)}(z), z)^2 - \varphi(X_{(i,N+1)}(z), z)^2 \\ &= \sum_{i=1}^{k-1} \varphi(X_{(i,N)}(z), z)^2 - \varphi(X_{(i,N)}(z), z)^2 + \varphi(X_{(k,N)}(z), z)^2 - \varphi(X_{N+1}, z)^2 \\ &= \varphi(X_{(k,N)}(z), z)^2 - \varphi(X_{N+1}, z)^2, \end{aligned}$$

$$\begin{aligned}
(ii) \quad & \sum_{\substack{i,j=1 \\ i < j}}^k \varphi(X_{(i,N)}(z), z) \varphi(X_{(j,N)}(z), z) - \varphi(X_{(i,N+1)}(z), z) \varphi(X_{(j,N+1)}(z), z) \\
&= \sum_{\substack{i,j=1 \\ i < j}}^{k-1} \varphi(X_{(i,N)}(z), z) \varphi(X_{(j,N)}(z), z) - \varphi(X_{(i,N)}(z), z) \varphi(X_{(j,N)}(z), z) \\
&+ \sum_{i=1}^{k-1} \varphi(X_{(i,N)}(z), z) \varphi(X_{(k,N)}(z), z) - \varphi(X_{(i,N)}(z), z) \varphi(X_{N+1}, z) \\
&= (\varphi(X_{(k,N)}(z), z) - \varphi(X_{N+1}, z)) \left(\sum_{i=1}^{k-1} \varphi(X_{(i,N)}(z), z) \right).
\end{aligned}$$

This yields the second identity of the lemma. \square

A.2 Proofs of Section 4.1.3

Proof of Lemma 4.1: The set $G_k(y)$ defined in (5) implicitly depends on the previous samples X_1, \dots, X_N . To distinguish between these sets with respect to X_1, \dots, X_N and with respect to $X_1, \dots, X_N, X_{N+1}, \dots, X_M$, respectively, we introduce the notation $G_{k,N}(y), G_{k,N+M}(y)$. These two sets satisfy $G_{k,N+M}(y) \subseteq G_{k,N}(y)$. Moreover, $\varphi(X_{(k,N+M)}(z), z) \leq \varphi(X_{(k,N)}(z), z)$ for all $z \in G_\delta$, since the growth condition $\varphi(\cdot, \cdot)$ is monotone with respect to the Euclidean distance of its components. For any $y \in G_\delta$, this yields the estimate

$$\begin{aligned}
\mu^o((X_1, \dots, X_N), y) &= \sum_{z \in G_{k,N}(y)} \mathbb{1}_{\{s(z) \geq \gamma\}} \left[\varphi(X_{(k,N)}(z), z)^2 - \varphi(y, z)^2 \right. \\
&\quad \left. + 2(\varphi(X_{(k,N)}(z), z) - \varphi(y, z)) \left(\sum_{i=1}^{k-1} \varphi(X_{(i,N)}(z), z) \right) \right] \nu_{G_\delta}(z) \\
&\geq \sum_{z \in G_{k,N+M}(y)} \mathbb{1}_{\{s(z) \geq \gamma\}} \left[\varphi(X_{(k,N+M)}(z), z)^2 - \varphi(y, z)^2 \right. \\
&\quad \left. + 2(\varphi(X_{(k,N+M)}(z), z) - \varphi(y, z)) \left(\sum_{i=1}^{k-1} \varphi(X_{(i,N+M)}(z), z) \right) \right] \nu_{G_\delta}(z) \\
&= \mu^o((X_1, \dots, X_N, X_{N+1}, \dots, X_{N+M}), y).
\end{aligned}$$

But this ensures that $\mu^o((X_1, \dots, X_N), y) > 0$ if $\mu^o((X_1, \dots, X_{N+M}), y) > 0$, and so the claim follows by the proportionality (7). \square

A.3 Proofs of Section 4.2.1

Proof of Lemma 4.2: Let (X_1, \dots, X_N) be a sample of size N . We first consider the case without duplicate samples, i.e., $d(G_\delta) = 0$. By definition of $G_k(y)$, the set is monotonically decreasing in N . Thus, we can establish a lower bound for $G_k(y)$ by assuming that $y \in \{X_1, \dots, X_N\}$ for all $y \in G_\delta$. Then for every $y \in G_\delta$ the k -NN are just the k nearest points in G_δ . In this case, $G_k(y) = H_l(y)$ holds when $k = |H_l(y)|$, and it follows directly that $H_l(y) \subseteq G_k(y)$ if $k \geq |H_l|$.

Let us consider next the case with duplicate samples, i.e., $d(G) > 0$. Denote with $G'_k(y)$ the

set $G_k(y)$ based on $(X_1, \dots, X_{N'})$, $N' \leq N$, such that the duplicate samples are removed from the sample (X_1, \dots, X_N) . First, we assume that that all duplicate samples are located at y , as this provides a lower bound for $G_k(y)$ again. Then

$$G_{k+d(\{y\})}(y) = G'_k(y).$$

We deduce that $G_k \subseteq H_l(y)$ if $k \geq |H_l(y)| + d(\{y\})$. Second, we assume that the duplicate samples are on $H_l(y)$. As above if k is chosen such that $G'_k(y) \subseteq H_l(y)$, then $G_{k+d(H_l(y))}(y) \subseteq H_l(y)$. So if $k \geq |H_l(y)| + d(H_l(y))$ we have $H_l(y) \subseteq G_k(y)$. \square

Proof of Lemma 4.3: Without loss of generality we can assume that $y = 0$. As G_δ is a discrete grid, we can write $H_l(0) = \bigcup_{i=0}^l \tilde{H}_i(0)$, with $\tilde{H}_i(0) = \{x \in G_\delta \mid |x| = \varepsilon_i \delta\}$. For $y \in \tilde{H}_i(0)$ we have $|y| = \sqrt{i_1^2 + \dots + i_d^2} \delta$ for some $i_1, \dots, i_d \in \mathbb{Z}$. Therefore $\varepsilon_i = \sqrt{i_1^2 + \dots + i_d^2}$ and ε_i is the $i+1$ -th smallest element in $I_d = \left\{ n \in \sqrt{\mathbb{N}_0} \mid \exists i_1, \dots, i_d \in \mathbb{Z} : n = \sqrt{i_1^2 + \dots + i_d^2} \right\}$.

Regarding the cardinality of I_d , we observe that if $d = 1$, then $\sqrt{i_1^2} = |i_1|$ and therefore, $I_1 = \{n \in \sqrt{\mathbb{N}_0} \mid \exists i_1 \in \mathbb{Z} : n = |i_1|\} = \mathbb{N}$. For cases where $d \geq 2$, we refer to the Sum of Two Squares Theorem, Legendre's Three-Square Theorem, and Lagrange's Four-Square Theorem. \square

A.4 Proofs of Section 4.3

Proof of Proposition 4.6: Let \mathbf{P} be defined as the extension of $\mathbf{P}_N = \mu_1 \otimes \mu_2 \otimes \dots \otimes \mu_N$, as outlined Section 2, and suppose that $(X_i)_{i \in \mathbb{N}}$ admits the distribution \mathbf{P} .

1.) We first focus on the assumptions of case (ii), noting that the following arguments remain valid for case (i). In this case, the stochastic kernel $\mu_{N+1}(\cdot, \cdot) \propto \mu_{N+1}^{\circ, H}(\cdot, \cdot)$ takes for $N \geq k$ the explicit form

$$\begin{aligned} \mu_{N+1}((X_1, \dots, X_N), y) &= \frac{1}{c_N} \sum_{z \in H_{l_N}(y)} \mathbf{1}_{\{s(z) \geq \gamma\}} \left[\varphi(X_{(k, N)}(z), z)^2 - \varphi(y, z)^2 \right. \\ &\quad \left. + 2(\varphi(X_{(k, N)}(z), z) - \varphi(y, z)) \left(\sum_{i=1}^{k-1} \varphi(X_{(i, N)}(z), z) \right) \right] \nu_{G_\delta}(z) \quad (11) \end{aligned}$$

with normalizing constant

$$\begin{aligned} c_N := \sum_{y \in G_\delta} \sum_{z \in H_{l_N}(y)} \mathbf{1}_{\{s(z) \geq \gamma\}} &\left[\varphi(X_{(k, N)}(z), z)^2 - \varphi(y, z)^2 + 2(\varphi(X_{(k, N)}(z), z) - \varphi(z, y)) \right. \\ &\left. \left(\sum_{i=1}^{k-1} \varphi(X_{(i, N)}(z), z) \right) \right] \nu_{G_\delta}(y). \end{aligned}$$

Our objective is to verify the claim that, with probability 1 under the measure \mathbf{P} , the sequence $(X_i)_{i \in \mathbb{N}}$ contains every element of the set $\{z \in G_\delta \mid s(z) \geq \gamma\}$ exactly k times. Then $\hat{s}_{k, N}(y) = s(y)$ for all $y \in \{z \in G_\delta \mid s(z) \geq \gamma\}$ with probability 1 under \mathbf{P} .

To prove this claim, we distinguish whether the first samples X_1, \dots, X_N contain $y \in \{z \in G_\delta \mid s(z) \geq \gamma\}$ less than k -times or at least k -times. For these two cases, we verify the following

properties of the stochastic kernel $\mu_{N+1}((X_1, \dots, X_N), y)$:

(a) $\mu_{N+1}((X_1, \dots, X_N), y) = 0$ if X_1, \dots, X_N contains y at least k -times:

Consider $y \in \{z \in G_\delta | s(z) \geq \gamma\}$, and suppose now that y appears at least k times in X_1, \dots, X_N . Then $G_k(y) = \{y\}$, $H_{l_N}(y) = \{y\}$ and $X_{(i,N)}(y) = y$ for all $i \in \{1, \dots, k\}$. Thus

$$\begin{aligned} & \mu_{N+1}((X_1, \dots, X_N), y) \\ &= \frac{1}{c_N} \left[\varphi(y, y)^2 - \varphi(y, y)^2 + 2(\varphi(y, y) - \varphi(y, y)) \left(\sum_{i=1}^{k-1} \varphi(y, y) \right) \right] \nu_{G_\delta}(y) = 0. \end{aligned}$$

(b) $\mu_{N+1}((X_1, \dots, X_N), y)$ is bounded from below if X_1, \dots, X_N contains y less than k -times:

Consider $y \in \{z \in G_\delta | s(z) \geq \gamma\}$ with $\nu_{G_\delta}(y) > 0$, and suppose that X_1, \dots, X_N contains y strictly less than k times. Define

$$\varepsilon := \min_{\substack{y_1, y_2 \in \{z \in G_\delta | s(z) \geq \gamma\} \\ y_1 \neq y_2}} \varphi(y_1, y_2)$$

in terms of the growth function $\varphi(\cdot, \cdot)$, and note that $\varepsilon > 0$, due to our assumption $\varphi(x, y) > 0$ if $x \neq y$. Considering at the right-hand side of (11) only $y \in H_{l_N}(y)$ and using Lemma A.1, we obtain together with $X_{(k,N)}(y) \neq y$ the estimate

$$\begin{aligned} & \mu_{N+1}((X_1, \dots, X_N), y) \\ & \geq \frac{1}{c_N} \left[\varphi(X_{(k,N)}(y), y)^2 + 2(\varphi(X_{(k,N)}(y), y) - \varphi(y, y)) \left(\sum_{i=1}^{k-1} \varphi(X_{(i,N)}(y), y) \right) \right] \nu_{G_\delta}(y) \\ & \geq \frac{1}{c_N} \varphi(X_{(k,N)}(y), y)^2 \nu_{G_\delta}(y) \\ & \geq \frac{1}{c_N} \varepsilon^2 \nu_{G_\delta}(y) > 0. \end{aligned}$$

Note now that each summand in the normalizing constant c_N is non-negative (cf. Lemma A.1) and decreases as $N \rightarrow \infty$. Moreover, the inclusion $H_{l_M}(y) \subseteq H_{l_N}(y)$ holds whenever $M \geq N$. Consequently, c_N is a decreasing sequence with respect to N , and so we have

$$\mu_{M+1}((X_1, \dots, X_M), y) \geq \frac{1}{c_N} \varepsilon^2 \nu_{G_\delta}(y) > 0 \quad \text{for all } M \geq N. \quad (12)$$

This implies that for all X_{M+1}, X_{M+2}, \dots the probability of sampling y is bounded below by a positive constant.

As $\mu_{N+1}(\cdot, \cdot) = \nu_{G_\delta}$ only if $N < k$, property (a) ensures that the sequence $(X_i)_{i \in \mathbb{N}}$ contains every value y at maximum k times with probability 1 under \mathbb{P} . Suppose now that the sample X_1, \dots, X_N contains $y \in \{z \in G_\delta | s(z) \geq \gamma\}$ only k' -times with $k' < k$. Using property (b), we show that \mathbb{P} -a.s. another y occurs in the samples $(X_i)_{i \in \mathbb{N}}$. By induction, the argument can be repeated as long as $k' = k - 1$, and so we finally verify our claim that the sequence $(X_i)_{i \in \mathbb{N}}$

contains y exactly k times, with probability 1 under \mathbb{P} . Indeed, for the events $A_i := \{X_{N+1} \neq y, \dots, X_{N+i} \neq y\}$, $i \in \mathbb{N}$, and the given samples X_1, \dots, X_N , we obtain together with (12)

$$\begin{aligned} \mathbb{P}[A_i | X_1, \dots, X_N] &= (1 - \mu_{N+1}((X_1, \dots, X_N), y)) \prod_{j=2}^i (1 - \mu_{N+j}((\cdot \neq y, \dots \neq y), y)) \\ &\leq \prod_{j=1}^i (1 - \frac{1}{c_N} \varepsilon^2 \nu_{G_\delta}(y)) = (1 - \frac{1}{c_N} \varepsilon^2 \nu_{G_\delta}(y))^i \end{aligned}$$

with constant $\alpha_N := 1 - \frac{1}{c_N} \varepsilon^2 \nu_{G_\delta}(y) \in (0, 1)$. This implies $\mathbb{P}[A_i] \leq (\alpha_N)^i$ for any $i \in \mathbb{N}$ and $\sum_{i=1}^{\infty} \mathbb{P}[A_i] \leq \frac{\alpha_N}{1 - \alpha_N} < \infty$, due to the geometric series. Thus, the Borel-Cantelli lemma yields $\mathbb{P}[A_i \text{ for infinitely many } i] = 0$.

Let us finally analyze the behavior of the k -NN regression under the additional assumption $|\{z \in G_\delta | s(z) \geq \gamma\}| < \infty$ for case ii). To verify that there is a N' , such that $|\hat{s}_{k, N'}(y) - s(y)| = 0$ for all $y \in \{z \in G_\delta | s(z) \geq \gamma\}$ and $\mu_{M'+1}(\cdot, \cdot) \equiv 0$ for $M' > N'$, we show the following additional properties:

- $\mu_{N+1}((X_1, \dots, X_N), y) \rightarrow 0$ if $y \notin \{x \in G_\delta | s(x) \geq \gamma\}$:

Assume $y \notin \{z \in G_\delta | s(z) \geq \gamma\}$. From above we have $H_{l_N}(y') \rightarrow \{y'\}$ for $y' \in \{z \in G_\delta | \{z \in G_\delta | s(z) \geq \gamma\}\}$ with probability 1 under \mathbb{P} . Thus, there exists a sufficiently large M such that $y \notin \cup_{y' \in \{z \in G_\delta | s(z) \geq \gamma\}} H_{l_M}(y')$. For such M we have $\mu_{M+1}((X_1, \dots, X_M), y) = 0$.

- $N' \geq k |\{x \in G_\delta | s(x) \geq \gamma\}|$:

From the properties above, we conclude there exists a minimum N' such that $X_1, \dots, X_{N'}$ contains every $y \in \{z \in G_\delta | s(z) \geq \gamma\}$ exactly k -times with probability 1 under \mathbb{P} such that $|\hat{s}_{k, N'}(y) - s(y)| = 0$. So N' must satisfy $N' \geq k |\{z \in G_\delta | s(z) \geq \gamma\}|$. By definition, $\mu_{M'+1}((X_1, \dots, X_{M'}), y)$ is proportional to 0 for every $M' > N'$ and $y \in G_\delta$.

2.) Under assumption (iii) most of the arguments above remain valid, and so we obtain analogously $\lim_{N \rightarrow \infty} |\hat{s}_{k, N}(y) - s(y)| = 0$ for $y \in \{x \in G_\delta | s(x) \geq \gamma\}$. For the sake of completeness, we will only address two specific aspects.

First, when sampling under $\mu_{N+1}(\cdot, \cdot) \propto \hat{\mu}_{N+1}^{\circ, H}(\cdot, \cdot)$ we consider a sampling kernel depending on the indicator $\mathbf{1}_{\{\hat{s}_{k, N}(x) \geq \gamma_N\}}$. If now $\mathbf{1}_{\{s(z) \geq \gamma\}} = 1 \Rightarrow \mathbf{1}_{\{\hat{s}_{k, N}(z) \geq \gamma_N\}} = 1$ for all $z \in G_\delta$ the arguments of case (ii) can be transferred to case (iii). Therefore we have to assume that $\{z \in G_\delta | s(z) \geq \gamma\} \subseteq \{z \in G_\delta | \hat{s}_{k, N}(z) \geq \gamma_N\}$ for the result to hold.

Second, if $N < k$, $\hat{s}_{k, k}(z)$ is a constant function in z . Thus, depending on the choice of γ_k ,

$$\{z \in G_\delta | \hat{s}_{k, k}(z) \geq \gamma_k\} \in \{\emptyset, G_\delta\}.$$

If $\{z \in G_\delta | \hat{s}_{k, k}(z) \geq \gamma_k\} = \emptyset$, then $\mu_{N+1}(\cdot, \cdot) \equiv 0$. Otherwise, $G_\delta = \{z \in G_\delta | \hat{s}_{k, k}(z) \geq \gamma_k\}$, thereby making $\mu_{k+1}(\cdot, \cdot) \propto \hat{\mu}_{k+1}^{\circ, H}(\cdot, \cdot)$ well-defined.

Under the additional assumption $|G_\delta| < \infty$ for case iii), the sets $\{z \in G_\delta | s(z) \geq \gamma\}$ and $\{z \in G_\delta | \hat{s}_{k, N}(z) \geq \gamma_N\}$ are also finite. By assumption, we have $\{z \in G_\delta | s(z) \geq \gamma\} \subseteq \{z \in G_\delta | \hat{s}_{k, N}(z) \geq \gamma_N\}$ for any N . Analogous to the results established above, we conclude that

there exists a minimum N' such that $X_1, \dots, X_{N'}$ contains every $y \in \{z \in G_\delta | s(z) \geq \gamma\}$ exactly k -times with probability 1 under \mathbb{P} . Then $\hat{s}_{k, M'}(y) = s(y)$ for all $M' \geq N'$. As γ_N is chosen increasingly, the kernel $\mu_M(\cdot, \cdot)$ will be proportional to 0 for some $M \geq N'$. Since $k|\{z \in G_\delta | s(z) \geq \gamma_N\}| \leq k|G_\delta| < \infty$, it follows that $N' < \infty$. \square

B Online Appendix

B.1 Simulated Annealing

Simulated Annealing (SA) is a meta-heuristic method used for approximating solutions to optimization problems. The method was introduced in Kirkpatrick (1984) and Černý (1985). An extensive overview can be found in Van Laarhoven & Aarts (1987). SA continues to be widely used across various fields. Comprehensive surveys on SA, discussing various approaches, are available in Franzin & Stützle (2019) and Delahaye, Chaimatana & Mongeau (2018). The field of meta-heuristics for optimization problems is vast, resulting in numerous extensions of SA and its integration with other meta-heuristic techniques. Examples of such extensions to the standard SA method can be found in Javidrad & Nazari (2017), Bouttier & Gavra (2019), Guilmeau, Chouzenoux & Elvira (2021), and Liang, Cheng & Lin (2014). The following summary and selection of components for SA are primarily based on the work of Franzin & Stützle (2019).

Let us consider a function $\mu : G \rightarrow \mathbb{R}$, defined on some countable set G . Our objective is to identify x^* , the global maximum of $\mu(\cdot)$, such that $x^* \in \arg \max_{x \in G} \mu(x)$. The neighborhood function $\mathbf{N}(x)$ is defined to include all values that can be accessed by the algorithm in one step. Additionally, let $(T_i)_{i \in \mathbb{N}}$ be the temperature sequence of the algorithm, where each $T_i > 0$. In Franzin & Stützle (2019), the specific SA algorithm is viewed as a combination of various components. These components are outlined in Algorithm 1 and will be discussed in detail below.

Algorithm 1 Simulated Annealing Algorithm

- 1: **Input:** a neighborhood function $\mathbf{N}(\cdot)$, an initial guess x_0 ;
 - 2: **Output:** approximation for x^* ;
 - 3: Set $i = 0$;
 - 4: Set $T_0 = \textit{initial temperature}$;
 - 5: **while** *stopping criterion* is not met **do**
 - 6: Generate $\tilde{x} \in \mathbf{N}(x_i)$;
 - 7: **if** $\mu(\tilde{x}) \geq \mu(x_i)$ **then**
 - 8: Set $x_{i+1} = \tilde{x}$;
 - 9: **else**
 - 10: With probability p_i accept set $x_{i+1} = \tilde{x}$, otherwise $x_{i+1} = x_i$;
 - 11: **if** *temperature length* is met **then**
 - 12: Update T_{i+1} according to cooling schedule;
 - 13: **else**
 - 14: $T_{i+1} = T_i$;
 - 15: Set $i = i + 1$;
 - 16: **Return** x_{i+1} ;
-

Neighborhood Function

The choice of the generation mechanism in the neighborhood $N(x)$ is highly problem-specific, as discussed in Henderson, Jacobson & Johnson (2003) and Hajek (1988). Following the approach in Liang, Cheng & Lin (2014), we employ a Gaussian random walk to generate the proposal \tilde{x} in Algorithm 1. Specifically, $\tilde{x} = x_i + \Delta x$, where $\Delta x \sim \mathcal{N}(0, \Sigma)$. In our case studies, we will use the same covariance matrix Σ as in the implementation of the Metropolis-Hastings algorithm (see Appendix B.2). The generated \tilde{x} is subsequently rounded to the nearest point in G .

Acceptance Probability

In Franzin & Stützle (2019), various acceptance probability strategies are discussed. For our purposes, we will use the traditional Metropolis-based acceptance probability, initially proposed in Kirkpatrick (1984) and Černý (1985). Therefore, we define the acceptance probability as

$$p_{i+1} := \exp\left(-\frac{1}{T_i}(\mu(x_i) - \mu(\tilde{x}))\right).$$

Initial Temperature

We implement the adaptive initial temperature strategy from Franzin & Stützle (2019) to control the acceptance of worsening samples. We begin by generating N_{avg} samples $x'_1, \dots, x'_{N_{avg}}$ from the Metropolis-Hastings algorithm described in Appendix B.2. The initial temperature is then set as $T_0 = \left\lceil \frac{\Delta_{avg}}{\log(p_0)} \right\rceil$, where

$$\Delta_{avg} = \frac{1}{N_{avg}} \sum_{i=1}^{N_{avg}} \mu(x'_i) - \mu(x'_{i+1}).$$

In our case studies, we will set $p_0 = 0.3$. For the estimation of Δ_{avg} , we draw 300 samples, with a burn-in of 100, considering every other sample. This yields $N_{avg} = 99$.

The Cooling Schedule

The transition of the algorithm from exploration to exploitation is typically managed through temperature updates, where $T_i \rightarrow 0$ as the algorithm progresses. In the seminal work by Hajek (1988), it was shown that simulated annealing converges when the temperature schedule is given by $T_i = 1/\log(t+d)$, where d represents the height of the largest local maximum, with its precise definition available in Hajek (1988). However, as argued in Nourani & Andresen (1998), this asymptotically slow cooling schedule can render the algorithm inefficient for practical use, effectively reducing it to a random search in the state space over time. To address this issue, alternatives tailored to specific problems have been proposed. Among these, both Franzin & Stützle (2019) and Nourani & Andresen (1998) advocate for the use of a geometric cooling schedule. In a geometric cooling schedule, the temperature is updated according to the rule $T_{n+1} = \alpha T_n$, where the parameter $\alpha \in (0, 1)$, dictates the rate of temperature decline. In practice, α is often selected to be close to 1, resulting in a gradual reduction of temperature that balances exploration and exploitation more effectively in many scenarios.

Temperature Length

The temperature length determines the frequency of updates to the cooling schedule and is crucial for effectively transitioning from exploration to exploitation within the algorithm. We will adopt the adaptive temperature length approach proposed in Franzin & Stützle (2019). In this method, the temperature is updated based on the acceptance of a predefined number of samples. Specifically, the temperature remains constant until a sufficient number of samples, drawn under the current temperature, have been accepted. This adaptive strategy allows the algorithm to dynamically adjust the exploration- exploitation balance, enhancing its efficiency across different stages of the optimization process. In our case studies, we found that updating the temperature after accepting 5 different samples provides a good balance between exploration and exploitation.

Stopping Criterion

For our specific purposes, implementing an adaptive termination criterion for the algorithm is advantageous. Among the different strategies discussed in Franzin & Stützle (2019), we have chosen to terminate the algorithm when the acceptance rate of the last samples falls below a predefined threshold. This approach enables the algorithm to dynamically conclude its execution based on the acceptance rate, providing flexibility in how thoroughly we search for the global maximum versus focusing on sampling in its vicinity. In our case study, we stop the algorithm when no new samples have been accepted in the last 10 iterations or when overall more than 200 proposals were made.

B.2 The Metropolis Hastings Algorithm

In Section 4, we examine stochastic kernels only known up to a normalizing constant. To circumvent the challenge of calculating this constant, we employ the Metropolis-Hastings algorithm. In this section we give a summary on the Metropolis-Hastings algorithm and sketch details on the implementations of our case studies.

Summary

This summary is based on Asmussen & Glynn (2007) and Robert & Casella (2004). The Metropolis-Hastings algorithm aims to construct an ergodic Markov chain $(\xi_i)_{i \in \mathbb{N}}$, which stationary distribution has a density resp. mass function proportional to a function $\mu(\cdot)$. By simulating the trajectories of this Markov chain, we obtain samples whose distribution converges to the targeted stationary distribution.

Let $q(x, y)$ be the probability density or mass function for transitioning from state x to state y in the Markov chain, serving as the proposal distribution. The Metropolis-Hastings algorithm simulates ξ_{k+1} based on the current state ξ_k as follows:

- (1.) Generate a proposed transition $Y \sim q(\xi_k, \cdot)$,

(2.) Update the state according to:

$$\xi_{k+1} = \begin{cases} Y & \text{with probability } \alpha(\xi_k, Y) \\ \xi_k & \text{with probability } 1 - \alpha(\xi_k, Y) \end{cases},$$

where

$$\alpha(x, y) := \min \left\{ 1, \frac{\mu(y)q(y, x)}{\mu(x)q(x, y)} \right\}.$$

Results concerning the requirements for the target $\mu(\cdot)$ and the proposal $q(\cdot, \cdot)$ that ensure convergence of the Markov chain to the desired stationary distribution can be found for the discrete case in Asmussen & Glynn (2007) and for the general case in Robert & Casella (2004).

A frequently considered special case is the random walk Metropolis-Hastings algorithm. Here, the proposal density depends only on the difference between the current state and the potential next state, such that $q(x, y) = q'(x - y)$. Consequently, the acceptance probability is given by

$$\alpha(x, y) \min \left\{ 1, \frac{\mu(y)}{\mu(x)} \right\}.$$

Implementation Details

In the simulations in Sections 5 and 6, we utilize the random walk Metropolis-Hastings algorithm to generate samples from the distribution $\hat{\mu}_{N+1}^{*,H}(\cdot, \cdot)$. These samples are obtained by evaluating the Markov chain after a predetermined burn-in period. We select a Gaussian random walk for the proposal distribution, defined as

$$q(x, y) = q'(x - y) = f_{x, \Sigma}(y),$$

where $f_{x, \Sigma}$ is the density function of the $\mathcal{N}(x, \Sigma)$ distribution, and Σ is the covariance matrix. In all of the case studies of the paper at hand, Σ is chosen as a diagonal matrix. The resulting sample is then rounded to the nearest grid point. For the subset sampling methods, we drew a total of 300 samples using $\mu(x) = \mathbb{1}_{\{\hat{s}_{k, N}(x) \geq \gamma_i\}}$. The burn-in period consisted of 100 samples, and from the remaining 200 samples, we selected every $\frac{200}{N_1}$ -th sample. Similarly, for the semi-adaptive method, we drew 300 samples from $\hat{\mu}_N^{*,H}(\cdot, \cdot)$, with a burn-in period of 100 samples, and used every $\frac{200}{N_1}$ -th sample. In the case of the fully adaptive method, we employed a burn-in period of 300 samples. To determine the initial temperature of the SA method, we again drew 300 samples from $\hat{\mu}_N^{*,H}(\cdot, \cdot)$, with a burn-in period of 100 samples, and considered every other sample thereafter.

In Robert & Casella (2004), a minimum requirement for convergence to the desired stationary distribution is specified as

$$\text{supp}(\hat{\mu}_{N+1}^{*,H}((X_1, \dots, X_N), y)) \subseteq \cup_{x \in \text{supp}(\hat{\mu}_{N+1}^{*,H}((X_1, \dots, X_N), y))} \text{supp}(q(x, \cdot)),$$

for all $y \in \{x \in G_\delta | s_{k, N}(x) \geq \gamma_N\}$. This is satisfied if the set $\{x \in G_\delta | s_{k, N}(x) \geq \gamma_N\}$ is connected. The minimum requirement will also be met even if the set $\{x \in G_\delta | s_{k, N}(x) \geq \gamma_N\}$ is not connected, provided that the smallest distances between the connected subsets are smaller

than the maximum transition distances of the random walk in every dimension.

B.3 Subset Sampling

In Section 4.1.2 (Fully Adaptive Proportional Sampling) and Section 4.1.3 (Semi-Adaptive Proportional Sampling) the kernels $\mu_{N+1}^*(\cdot, \cdot)$ utilize the growth condition $\varphi(\cdot, \cdot)$ and restrict samples to the set $\{z \in G_\delta | s(z) \geq \gamma\}$. A natural alternative is to sample from $\nu_{G_\delta}(\cdot)$ restricted to $\{z \in G_\delta | s(z) \geq \gamma\}$. This sampling method does not rely on the growth condition and focuses more on the exploration on $\{x \in G_\delta | s(x) \geq \gamma\}$. It is inspired by subset sampling techniques from reliability engineering, as discussed in Zuev (2015). This approach, also known as sequential Monte Carlo, is further detailed in works such as Cérou et al. (2012) and is based on the foundational research outlined in Del Moral, Doucet & Jasra (2006). A comprehensive overview of sequential Monte Carlo and its applications can be found in Chopin, Papaspiliopoulos, et al. (2020) and Doucet, De Freitas & Gordon (2001).

To motivate our subset sampling approach in more detail, we directly consider the discretized pathwise MSE \mathcal{E}_{N+1} after adding the sample X_{N+1} , as defined in (4), and analyze analogous to Section 3 the corresponding error reduction term $\mathcal{E}_N - \mathcal{E}_{N+1}$. Again, minimizing the pathwise MSE \mathcal{E}_{N+1} among the additional sample X_{N+1} is equivalent to maximizing the error reduction term. Repeating the arguments in the proofs of Lemma 3.2 and Lemma A.1, we obtain

$$\begin{aligned} \mathcal{E}_N - \mathcal{E}_{N+1} &= \frac{1}{k^2} \sum_{z \in G_k(X_{N+1})} \mathbb{1}_{\{s(z) \geq \gamma\}} \left[\left(\sum_{i=1}^k s(X_{(i,N)}(z)) - s(z) \right)^2 \right. \\ &\quad \left. - \left(\sum_{i=1}^k s(X_{(i,N+1)}(z)) - s(z) \right)^2 \right] \nu_{G_\delta}(z) \\ &= \frac{1}{k^2} \left[\sum_{z \in G_k(X_{N+1})} \mathbb{1}_{\{s(z) \geq \gamma\}} \left[(s(X_{(k,N)}(z)) - s(z))^2 - (s(X_{N+1}) - s(z))^2 \right. \right. \\ &\quad \left. \left. + 2(s(X_{(k,N)}(z)) - s(X_{N+1})) \left(\sum_{i=1}^{k-1} s(X_{(i,N+1)}(z)) - s(z) \right) \right] \nu_{G_\delta}(z) \right]. \quad (13) \end{aligned}$$

The optimal sample X_{N+1} is thus a value that maximizes the right-hand side in (13). Hence, a reduction in the MSE can only occur if X_{N+1} belongs to the k -NN of some $y \in \{z \in G_\delta | s(z) \geq \gamma\}$. For other possible samples, the inner part at the right-hand side of (13) remains zero. Thus, the potentially beneficial samples are contained in

$$\bigcup_{\substack{x \in G_\delta \\ \exists y \in G_k(x): s(y) \geq \gamma}} G_k(x).$$

For $\mathbb{P} = \nu_{G_\delta}^{\otimes N}$ we have $G_k(x) \rightarrow \{x\}$ as $N \rightarrow \infty$ with probability 1, hence

$$\bigcup_{\substack{x \in G_\delta \\ \exists y \in G_k(x): s(y) \geq \gamma}} G_k(x) \xrightarrow{N \rightarrow \infty} \{x \in G_\delta | s(x) \geq \gamma\},$$

with probability 1. To shift more weight in the area where the possible beneficial samples are,

we condition the underlying probability ν_{G_δ} to $\{z \in G_\delta | s(z) \geq \gamma\}$ and sample from

$$\mu_S^*(z) \propto \mathbb{1}_{\{s(z) \geq \gamma\}} \nu_{G_\delta}(z).$$

B.4 Balancing Exploitation and Exploration

As outlined in Section 3, we balance exploration and exploitation by sampling from kernels proportional to

$$\mu^\circ((X_1, \dots, X_N), y)^\vartheta, \quad \vartheta > 0,$$

that is, a power-tempered version of the estimated error reduction. In this section, we introduce a target function to choose the optimal parameter ϑ and explain its calculation based on stochastic gradient ascent. Moreover, we discuss practicability issues and quantify the impact of ϑ on the MSE of k -NN regression.

B.4.1 Sampling Kernels and Stochastic Gradient Ascent

For fixed previous samples (X_1, \dots, X_N) , we write

$$\mu_{N+1}^\circ := \mu^\circ((X_1, \dots, X_N), y), \quad y \in G_\delta,$$

and define for $\vartheta > 0$ the kernel

$$\mu_{\vartheta, N+1}(y) := \mu_{\vartheta}((X_1, \dots, X_N), y) = \frac{\mu^\circ((X_1, \dots, X_N), y)^\vartheta}{C(\vartheta)}, \quad y \in G_\delta,$$

with normalizing constant $C(\vartheta) = \sum_{y \in G_\delta} \mu^\circ((X_1, \dots, X_N), y)^\vartheta$.

To choose ϑ , we target the expected reduction in the true error when adding the $(N+1)$ -th sample. Let $R(X) = \mathcal{E}_N - \mathcal{E}_{N+1}$ denote the actual reduction obtained by adding X . Our objective is

$$\vartheta^* \in \arg \max_{\vartheta > 0} \mathbf{E}_{\mu_{\vartheta, N+1}} [R(X)].$$

To identify the optimal ϑ , we derive a first order condition.

Lemma B.1. *Under the assumption that derivation and expectation can be interchanged we have*

$$\frac{\partial}{\partial \vartheta} \mathbf{E}_{\mu_{\vartheta, N+1}} [R(X)] = \text{Cov}_{\mu_{\vartheta, N+1}}(R(X), \log(\mu_{N+1}^\circ(X))).$$

Proof. Direct calculations yield

$$\begin{aligned} \frac{\partial}{\partial \vartheta} \mathbf{E}_{\mu_{\vartheta, N+1}} [R(X)] &= \frac{\partial}{\partial \vartheta} \sum_{x \in G_\delta} R(x) \frac{\mu_{N+1}^\circ(x)^\vartheta}{C(\vartheta)} = \sum_{x \in G_\delta} R(x) \left[\frac{\partial}{\partial \vartheta} \frac{\mu_{N+1}^\circ(x)^\vartheta}{\sum_{y \in G_\delta} \mu_{N+1}^\circ(y)^\vartheta} \right] \\ &= \sum_{x \in G_\delta} R(x) \left[\frac{\mu_{N+1}^\circ(x)^\vartheta \log(\mu_{N+1}^\circ(x)) C(\vartheta) - \mu_{N+1}^\circ(x)^\vartheta \left(\sum_{y \in G_\delta} \mu_{N+1}^\circ(y)^\vartheta \log(\mu_{N+1}^\circ(y)) \right)}{C(\vartheta)^2} \right] \\ &= \sum_{x \in G_\delta} R(x) \frac{\mu_{N+1}^\circ(x)^\vartheta}{C(\vartheta)} \left[\log(\mu_{N+1}^\circ(x)) - \mathbf{E}_{\mu_{\vartheta, N+1}} [\log(\mu_{N+1}^\circ(X))] \right] \end{aligned}$$

$$\begin{aligned}
&= \mathbf{E}_{\mu_{\vartheta, N+1}} [R(X) \log(\mu_{N+1}^o(X))] - \mathbf{E}_{\mu_{\vartheta, N+1}} [R(X)] \mathbf{E}_{\mu_{\vartheta, N+1}} [\log(\mu_{N+1}^o(X))] \\
&= \text{Cov}_{\mu_{\vartheta, N+1}} (R(X), \log(\mu_{N+1}^o(X))).
\end{aligned}$$

□

Using this first order condition, the optimal ϑ is estimated via stochastic gradient ascent. The procedure is summarized in Algorithm 2.

Algorithm 2 Stochastic gradient ascent for estimating ϑ

- 1: **Input:** Initial guess $\vartheta_0 > 0$, learning rate α , Monte Carlo batch size M
- 2: **Output:** Approximate maximizer of ϑ
- 3: Set $t = 0$;
- 4: **while** *stopping criterion* is not met **do**
- 5: Generate samples Y_1, \dots, Y_M from $\mu_{\vartheta_t, N+1}$ via Metropolis-Hastings algorithm;
- 6: Compute estimates $\hat{R}(Y_1), \dots, \hat{R}(Y_M)$;
- 7: Compute $\mu^o(Y_1), \dots, \mu^o(Y_M)$;
- 8: Compute the sample covariance

$$\widehat{\text{Cov}}_{\mu_{\vartheta_t, N+1}} (R(Y), \log(\mu_{N+1}^o(Y)));$$

- 9: Update

$$\vartheta_{t+1} = \vartheta_t + \alpha \widehat{\text{Cov}}_{\mu_{\vartheta_t, N+1}} (R(Y), \log(\mu_{N+1}^o(Y)));$$

- 10: Set $t = t + 1$;
 - 11: **Return** ϑ_t ;
-

Remark B.2. *The error reduction $R(Y) = \mathcal{E}_N - \mathcal{E}_{N+1}$ for a candidate sample Y takes the explicit form*

$$\begin{aligned}
R(Y) = \frac{1}{k^2} &\left[\sum_{z \in G_k(Y)} \mathbf{1}_{\{s(z) \geq \gamma\}} \left((s(X_{(k,N)}(z)) - s(z))^2 + (s(Y) - s(z))^2 \right. \right. \\
&\quad \left. \left. - 2(s(X_{(k,N)}(z)) - s(Y)) \left(\sum_{i=1}^{k-1} (s(X_{(i,N+1)}(z)) - s(z))^2 \right) \right) \nu_{G_\delta}(z) \right],
\end{aligned}$$

as shown in (13). To derive the estimator $\hat{R}(Y)$, we approximate $s(z)$ for $z \in G_k(Y)$ by $s(X_{(1,N)}(z))$ and $s(Y)$ by $\hat{s}_{k,N}(Y)$. If this plug-in estimator is too noisy for stable updates, additional evaluations of $s(\cdot)$ may be allocated (e.g., refining $s(Y)$ or local surrogates for $s(z)$ to improve the accuracy of $\hat{R}(Y)$).

B.4.2 Practicability

The stochastic gradient ascent in Algorithm 2 requires substantial computational effort. In many practical settings, estimating the optimal ϑ takes more computational resources than generating the samples for the k -NN regression.

The main driver of the computational effort is the evaluation of $\hat{R}(Y)$. Each evaluation requires determining $G_k(Y)$, which can be very large (even infinite in some cases). Consequently, a single gradient step entails: drawing M samples from $\mu_{\vartheta_t, N+1}$ via the MH-Algorithm, and

(ii) computing M computational costly values of $\widehat{R}(Y)$. Repeating this over many updates dominates the total runtime.

A more direct alternative is an offline selection from a small candidate set $\vartheta'_1, \dots, \vartheta'_T$. For each ϑ'_t , we run the full methods of Section 3 to obtain N samples for the k -NN regression for each of the ϑ'_t . Since the evaluation of the k -NN regression in the algorithms involve sorting/neighbor search, the run cost scales at least as $\mathcal{O}(N \log(N))$. This is often more efficient than embedding an inner ϑ -optimization within a simulation. To compare the candidate ϑ_t we draw a common evaluation set Z_1, \dots, Z_M from ν_{G_δ} and estimate the discrete error in (4) as

$$\frac{1}{M} \sum_{j=1}^M \mathbb{1}_{\{s(Z_j) \geq \gamma\}} \left(\frac{1}{k} \sum_{i=1}^k s(X_{(i, N+1)}(Z_j) - s(Z_j)) \right)^2.$$

Effect of ϑ on k -NN MSE

To quantify the impact of ϑ on the MSE of the k -NN regression, we ran the semi-adaptive sampling method in the setting of Section 5.1 over different values of ϑ for the linear and the non-linear functions. We used the same growth conditions and algorithmic settings as in that section. For each configuration, we performed 300 independent runs and report the mean MSE. The results are summarized in Figure 7.

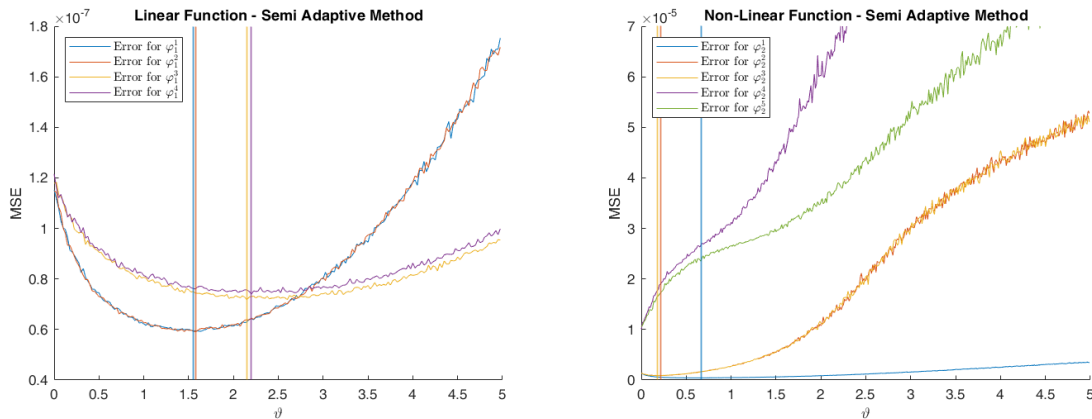


Figure 7: Mean MSE versus ϑ for the semi-adaptive method, for the linear and non-linear functions, under the respective growth conditions. For each growth condition, the ϑ attaining the lowest mean MSE is indicated by a vertical line.

- *Linear Function.* The empirical optimum occurs near $\vartheta \approx 1.5$ for $\varphi_1^1(\cdot)$ and $\varphi_1^2(\cdot)$, and near $\vartheta \approx 2.2$ for $\varphi_1^3(\cdot)$ and $\varphi_1^4(\cdot)$. Moving away from these values in either direction increases the MSE. Moreover, $\varphi_1^1(\cdot)$ and $\varphi_1^2(\cdot)$ achieve lower minimal MSE, but exhibit a higher growth when ϑ deviates from the minimum.
- *Non-linear Function.* The optimal ϑ is small: approximately 0.6 for $\varphi_2^1(\cdot)$, 0.2 for $\varphi_2^2(\cdot), \varphi_2^3(\cdot)$, and effectively 0 for $\varphi_2^4(\cdot)$ and $\varphi_2^5(\cdot)$ (i.e., close to uniform sampling over the effective support). In all non-linear cases, increasing ϑ leads to a worse MSE.

Overall, in these case studies relatively small values of ϑ are preferable.

B.5 Computational Effort of the Case Studies

	Linear Function	Non-Linear Function	χ^2 -Distribution	ALM Model
Crude	0.5ms	3.5ms	8ms	0.1s
Subset	0.1s	0.6s	1s	3.3s
Semi-Adaptive	0.8s	1.5s	2.4s	16.7s
Fully-Adaptive	28s	40s	86s	632s
SA	76s	62s	96s	1047s

Table 1: Computation Times for the sampling methods in the different case studies.

All case studies presented in this paper were implemented in MATLAB and executed on a single core with a clock speed of 1.80 GHz. Table 1 provides the computation times required to obtain the samples for the various methods and case studies. The algorithms are times according to the setup in Section 5.1 and 6.2. It is important to note that the time taken to calculate the resulting MSE, which represents the primary computational workload in these case studies, is not included in these timing estimates.