Journal of Scientific Computing (2023) 95:12
https://doi.org/10.1007/s10915-023-02130-8

Computation of Conditional Expectations with Guarantees

Patrick Cheridito[1] · **Balint Gersey**[1]

Received: 27 February 2022 / Revised: 23 December 2022 / Accepted: 5 January 2023 /
Published online: 17 February 2023

Abstract
Theoretically, the conditional expectation of a square-integrable random variable Y given a d-dimensional random vector X can be obtained by minimizing the mean squared distance between Y and $f(X)$ over all Borel measurable functions $f\colon \mathbb{R}^d \to \mathbb{R}$. However, in many applications this minimization problem cannot be solved exactly, and instead, a numerical method which computes an approximate minimum over a suitable subfamily of Borel functions has to be used. The quality of the result depends on the adequacy of the subfamily and the performance of the numerical method. In this paper, we derive an expected value representation of the minimal mean squared distance which in many applications can efficiently be approximated with a standard Monte Carlo average. This enables us to provide guarantees for the accuracy of any numerical approximation of a given conditional expectation. We illustrate the method by assessing the quality of approximate conditional expectations obtained by linear, polynomial and neural network regression in different concrete examples.

Keywords Conditional expectation · Least squares regression · Monte Carlo methods · Numerical guarantees · Trustworthy AI

Mathematics Subject Classification 62J02 · 65G99 · 65C05 · 65C20 · 68T05

1 Introduction

The goal of this paper is to compute the conditional expectation $\mathbb{E}[Y \mid X]$ of a square-integrable random variable Y given a d-dimensional random vector X, both defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The accurate estimation of conditional expectations is an important problem arising in different branches of science and engineering as well as finance, economics and various business applications. In particular, it plays a central role in regression analysis, which tries to model the relationship between a response variable Y and a number of explanatory variables $X_1, \ldots, X_d$ [see, e.g., 17, 20, 29, 36]. But it also

We thank Daniel Bartl, Sebastian Becker and Peter Bühlmann for fruitful discussions and helpful comments.

✉ Patrick Cheridito
patrick.cheridito@math.ethz.ch

[1] Department of Mathematics, ETH Zurich, Zürich, Switzerland

appears in different computational problems, such as the numerical approximation of partial differential equations and backward stochastic differential equations [see, e.g., 4, 7, 10, 19, 21, 25, 26], stochastic partial differential equations [see, e.g., 6], stochastic control problems [see, e.g., 2, 3], stochastic filtering [see, e.g., 31], complex financial valuation problems [see, e.g. 8, 11, 12, 16, 35, 37] as well as financial risk management [see, e.g. 5, 13, 18, 28, 34]. In addition, conditional expectations are closely related to squared loss minimization problems arising in various machine learning applications [see, e.g., 27, 29].

If it is possible to simulate from the conditional distribution of Y given X, the conditional expectation $\mathbb{E}[Y \mid X]$ can be approximated with nested Monte Carlo simulation; see, e.g., [5, 13, 14]. While the approach can be shown to converge for increasing sample sizes, it often is too time-consuming to be useful in practical applications. On the other hand, it is well known that $\mathbb{E}[Y \mid X]$ is of the form $\bar{f}(X)$ for a regression function $\bar{f}\colon \mathbb{R}^d \to \mathbb{R}$ which can be characterized as a minimizer[1] of the mean squared distance

$$\mathbb{E}\big[(Y - f(X))^2\big] \tag{1.1}$$

over all Borel functions $f\colon \mathbb{R}^d \to \mathbb{R}$ [see, e.g., 15]. However, in many applications, the minimization problem (1.1) cannot be solved exactly. For instance, the joint distribution of X and Y might not be known precisely, or the problem might be too complicated to admit a closed-form solution. In such cases, it can be approximated with a least squares regression, consisting in minimizing an empirical mean squared distance

$$\frac{1}{M}\sum_{m=1}^{M}\big(\mathcal{Y}^m - f(\mathcal{X}^m)\big)^2 \tag{1.2}$$

based on realizations $(\mathcal{X}^m, \mathcal{Y}^m)$ of (X, Y) over a suitable family $\mathcal{S}$ of Borel functions $f\colon \mathbb{R}^d \to \mathbb{R}$. This typically entails the following three types of approximation errors:

(i) a function approximation error if the true regression function $\bar{f}$ does not belong to the function family $\mathcal{S}$;
(ii) a statistical error stemming from estimating the expected value (1.1) with (1.2);
(iii) a numerical error if the minimization of (1.2) over $\mathcal{S}$ has to be solved numerically.

Instead of analyzing the errors (i)–(iii), we here derive an alternative representation of the minimal mean squared distance $\mathbb{E}[(Y - \bar{f}(X))^2]$, which does not involve a minimization problem or require knowledge of the true regression function $\bar{f}$. This enables us to provide quantitative estimates on the accuracy of any numerical approximation $\hat{f}$ of $\bar{f}$. In particular, if $\hat{f}$ is determined with a machine learning method that is difficult to interpret, our approach contributes to trustworthy AI.

While the empirical mean squared distance (1.2) can directly be minimized using realizations $(\mathcal{X}^m, \mathcal{Y}^m)$ of (X, Y), our approach to derive error bounds for the approximation of $\bar{f}$ requires Y to be of the form $Y = h(X, V)$ for a known function $h\colon \mathbb{R}^{d+k} \to \mathbb{R}$ and a k-dimensional random vector V independent of X. In typical statistical applications, only realizations of (X, Y) can be observed and a structure of the form $Y = h(X, V)$ would have to be inferred from the data. But in many of the computational problems mentioned above, Y is directly given in the form $Y = h(X, V)$.

The rest of the paper is organized as follows: In Sect. 2, we first introduce the notation and some preliminary results before we formulate the precise mean squared distance minimization problem we are considering along with its empirical counterpart. Then we discuss upper

[1] The conditional expectation $\mathbb{E}[Y \mid X]$ is unique up to $\mathbb{P}$-almost sure equality. Accordingly, the regression function $\bar{f}$ is unique up to almost sure equality with respect to the distribution of X.

bounds of the minimal mean squared distance and their approximation with Monte Carlo averages. In Sect. 3 we derive an expected value representation of the minimal mean squared distance which makes it possible to derive bounds on the L^2-error of any numerical approximation $\hat{f}$ of the true regression function $\bar{f}$. In Sect. 4 we compute conditional expectations in different examples using linear regression, polynomial regression and feedforward neural networks with varying activation functions. We benchmark the numerical results against values obtained from our expected value representation of the minimal mean squared distance and derive L^2-error estimates. Section 5 concludes, and in the Appendix we report auxiliary numerical results used to compute the figures shown in Sect. 4.

2 Numerical Approximation of Conditional Expectations

2.1 Notation and Preliminaries

Let us first note that the mean squared distance (1.1) does not necessarily have to be minimized with respect to the original probability measure $\mathbb{P}$. Indeed, the regression function $\bar{f}\colon \mathbb{R}^d \to \mathbb{R}$ only depends on the conditional distribution of Y given X and not on the distribution ν_X of X. More precisely, the measure $\mathbb{P}$ can be disintegrated as

$$\mathbb{P}[A] = \int_{\mathbb{R}^d} \mathbb{P}[A \mid X = x] d\nu_X(x), \quad A \in \mathcal{F},$$

where $\mathbb{P}[. \mid X = x]$ is a regular conditional version of $\mathbb{P}$ given X. For any Borel probability measure ν on $\mathbb{R}^d$ that is absolutely continuous with respect to ν_X,

$$\mathbb{P}^\nu[A] := \int_{\mathbb{R}^d} \mathbb{P}[A \mid X = x] d\nu(x), \quad A \in \mathcal{F},$$

defines a probability measure on Ω under which X has the modified distribution ν while the conditional distribution of Y given X is the same as under $\mathbb{P}$. Let us denote by $\mathbb{E}^\nu$ the expectation with respect to $\mathbb{P}^\nu$ and by $\mathcal{B}(\mathbb{R}^d; \mathbb{R})$ the set of all Borel functions $f\colon \mathbb{R}^d \to \mathbb{R}$. With this notation, one has the following.

Lemma 2.1 *Assume $\mathbb{E}^\nu Y^2 < \infty$. Then a minimizer $\tilde{f}\colon \mathbb{R}^d \to \mathbb{R}$ of the distorted minimal mean squared distance*

$$D^\nu := \min_{f \in \mathcal{B}(\mathbb{R}^d; \mathbb{R})} \mathbb{E}^\nu\big[(Y - f(X))^2\big] \tag{2.1}$$

agrees with $\bar{f}\colon \mathbb{R}^d \to \mathbb{R}$ ν-almost surely. In particular, if ν has the same null sets as ν_X, then $\tilde{f} = \bar{f}$ ν_X-almost surely.

Proof A Borel function $\tilde{f}\colon \mathbb{R}^d \to \mathbb{R}$ minimizes (2.1) if and only if

$$\tilde{f}(x) = \arg\min_{z \in \mathbb{R}} \int_{\mathbb{R}} (y - z)^2 \, \mathbb{P}[Y \in dy \mid X = x] \quad \text{for } \nu\text{-almost all } x \in \mathbb{R}^d.$$

Since $\bar{f}$ has an analogous representation holding for ν_X-almost all $x \in \mathbb{R}^d$, it follows that $\tilde{f}$ agrees with $\bar{f}$ ν-almost surely. In particular, if ν has the same null sets as ν_X, then $\tilde{f} = \bar{f}$ ν_X-almost surely. □

Lemma 2.1 gives us the flexibility to choose a distribution ν on $\mathbb{R}^d$ which assigns more weight than ν_X to regions of $\mathbb{R}^d$ that are important in a given application. For instance, $\nu \ll \nu_X$

can be chosen so as to concentrate more weight around a given point x_0 in the support of ν_X; see Lemma 3.6 and Sect. 4.2.2 below. On the other hand, in financial risk management one is usually concerned with the tails of loss distributions. Then the distribution ν_X can be tilted in the direction of large losses of a financial exposure; see Sect. 4.4.2 below.

2.2 Upper Bound of the Minimal Mean Squared Distance

In many situations, the minimization problem (2.1) cannot be solved exactly. But if one has access to $\mathbb{P}^\nu$-realizations $(\mathcal{X}^m, \mathcal{Y}^m)$ of (X, Y), the true regression function $\bar{f}$ can be approximated by minimizing the empirical mean squared distance

$$\frac{1}{M}\sum_{m=1}^{M}\left(\mathcal{Y}^m - f(\mathcal{X}^m)\right)^2 \tag{2.2}$$

over f in a subset $\mathcal{S}$ of $\mathcal{B}(\mathbb{R}^d;\mathbb{R})$. In the examples of Sect. 4 below, we compare results obtained by using linear combinations of $1, X_1, \ldots, X_d$, second order polynomials in $X_1, \ldots, X_d$ as well as feedforward neural networks with different activation functions.

But irrespective of the method used to obtain an approximation of $\bar{f}$, any Borel measurable candidate regression function $\hat{f} : \mathbb{R}^d \to \mathbb{R}$ yields an upper bound

$$U^\nu := \mathbb{E}^\nu\left[\left(Y - \hat{f}(X)\right)^2\right] \tag{2.3}$$

of the minimal mean squared distance D^ν. However, since in typical applications, U^ν cannot be calculated exactly, we approximate it with a Monte Carlo estimate

$$U_N^\nu := \frac{1}{N}\sum_{n=1}^{N}\left(Y^n - \hat{f}(X^n)\right)^2 \tag{2.4}$$

based on N independent $\mathbb{P}^\nu$-realizations $(X^n, Y^n)_{n=1}^N$ of (X, Y) drawn independently of any data $(\mathcal{X}^m, \mathcal{Y}^m)_{m=1}^M$ used to determine $\hat{f}$.

Provided that $\mathbb{E}^\nu[(Y - \hat{f}(X))^2] < \infty$, one obtains from the strong law of large numbers that

$$\lim_{N\to\infty} U_N^\nu = U^\nu \quad \mathbb{P}^\nu\text{-almost surely.}$$

To derive confidence intervals, we compute the sample variance

$$v_N^{U,\nu} := \frac{1}{N-1}\sum_{n=1}^{N}\left(\left(Y^n - \hat{f}(X^n)\right)^2 - U_N^\nu\right)^2$$

and denote, for $\alpha \in (0,1)$, by q_α the α-quantile of the standard normal distribution. Then the following holds.

Lemma 2.2 *Assume $\mathbb{E}^\nu Y^4 < \infty$ and $\mathbb{E}^\nu |\hat{f}(X)|^4 < \infty$. Then, for every $\alpha \in (1/2, 1)$,*

$$\liminf_{N\to\infty} \mathbb{P}^\nu\left[\left|U^\nu - U_N^\nu\right| \le q_{1-\alpha}\sqrt{\frac{v_N^{U,\nu}}{N}}\right] \ge 1 - 2\alpha. \tag{2.5}$$

Proof In the special case where $Y = \hat{f}(X)$ $\mathbb{P}^\nu$-almost surely, one has $U^\nu = U_N^\nu = v_N^{U,\nu} = 0$ $\mathbb{P}^\nu$-almost surely for all $N \geq 1$. So (2.5) holds trivially. On the other hand, if $\mathbb{P}^\nu[Y \neq \hat{f}(X)] > 0$, it follows from the assumptions and the strong law of large numbers that $v_N^{U,\nu}$ converges $\mathbb{P}^\nu$-almost surely to $\mathrm{Var}^{\mathbb{P}^\nu}\left((Y - \hat{f}(X))^2\right) > 0$ for $N \to \infty$. Therefore, one obtains from the central limit theorem and Slutky's theorem that

$$\lim_{N\to\infty} \mathbb{P}^\nu\left[\sqrt{\frac{N}{v_N^{U,\nu}}}\left|U^\nu - U_N^\nu\right| \leq q_{1-\alpha}\right] = 1 - 2\alpha,$$

which shows (2.5). □

3 Error Estimates

Now, our goal is to derive bounds on the approximation error $\hat{f} - \bar{f}$ for a given candidate regression function $\hat{f}\colon \mathbb{R}^d \to \mathbb{R}$. To do that we assume in this section that Y has a representation of the form:

(R) *$Y = h(X, V)$ for a Borel function $h\colon \mathbb{R}^{d+k} \to \mathbb{R}$ and a $k-$dimensional random vector V that is independent of X under $\mathbb{P}^\nu$.*

Remark 3.1 Provided that the probability space $(\Omega, \mathcal{F}, \mathbb{P}^\nu)$ is rich enough, Y can always be assumed to be of the form (R). Indeed, if $(\Omega, \mathcal{F}, \mathbb{P}^\nu)$ supports a random variable V which, under $\mathbb{P}^\nu$, is uniformly distributed on the unit interval $(0, 1)$ and independent of X, the function $h\colon \mathbb{R}^d \times (0, 1) \to \mathbb{R}$ can be chosen as a conditional $\mathbb{P}^\nu$-quantile function of Y given X and extended to the rest of $\mathbb{R}^{d+1}$ arbitrarily. Then $(X, h(X, V))$ has the same $\mathbb{P}^\nu$-distribution as (X, Y), and, in particular,

$$\mathbb{E}^\nu[h(X, V) \mid X] = \bar{f}(X) \quad \nu\text{-almost surely.}$$

However, for our method to be applicable, the function h needs to be known explicitly.

A representation of the form (R) with a known function h is available in computational problems involving numerical regressions, such as regression methods to solve PDEs and BSDEs [see, e.g., 4, 7, 10, 19, 21, 25, 26], SPDEs [see, e.g., 6], financial valuation problems [see, e.g. 8, 11, 12, 16, 35, 37] or financial risk management problems [see, e.g. 5, 13, 18, 28, 34].

3.1 Alternative Representation of the Minimal Mean Squared Distance

The key ingredient of our approach is an alternative representation of the minimal mean squared distance

$$D^\nu = \min_{f \in \mathcal{B}(\mathbb{R}^d;\, \mathbb{R})} \mathbb{E}^\nu\left[(Y - f(X))^2\right] = \mathbb{E}^\nu\left[\left(Y - \bar{f}(X)\right)^2\right] \tag{3.1}$$

which does not involve a minimization problem or require knowledge of the true regression function $\bar{f}$ and, at the same time, can be approximated efficiently. An analogous representation exists for the squared L^2-norm of the conditional expectation

$$C^\nu := \mathbb{E}^\nu\, \bar{f}^2(X) = \left\|[\mathbb{E}^\nu[Y \mid X]\right\|^2_{L^2(\mathbb{P}^\nu)}, \tag{3.2}$$

which will be helpful in the computation of relative approximation errors in Sect. 4 below. If necessary, by enlarging[2] the probability space $(\Omega, \mathcal{F}, \mathbb{P}^\nu)$, we can assume it supports a k-dimensional random vector $\tilde{V}$ that has the same $\mathbb{P}^\nu$-distribution as V and is independent of (X, V) under $\mathbb{P}^\nu$. Let us define

$$Z := h(X, \tilde{V}).$$

Then, we have the following.

Proposition 3.2 *If* $\mathbb{E}^\nu Y^2 < \infty$, *then*

$$C^\nu = \mathbb{E}^\nu[YZ] \quad \textit{and} \quad D^\nu = \mathbb{E}^\nu[Y(Y-Z)]\,.$$

Proof It follows from independence of X, V and $\tilde{V}$ that

$$\mathbb{E}^\nu[YZ] = \mathbb{E}^\nu\Big[\mathbb{E}^\nu\Big[h(X,V)h(X,\tilde{V}) \mid X\Big]\Big] = \mathbb{E}^\nu\big[\bar{f}^2(X)\big] = C^\nu.$$

Similarly, one has

$$\mathbb{E}^\nu\big[Y\bar{f}(X)\big] = \mathbb{E}^\nu\big[\mathbb{E}^\nu[Y \mid X]\bar{f}(X)\big] = \mathbb{E}^\nu\big[\bar{f}^2(X)\big],$$

from which one obtains

$$\begin{aligned}\mathbb{E}^\nu[Y(Y-Z)] &= \mathbb{E}^\nu\big[Y^2 - \bar{f}^2(X)\big] = \mathbb{E}^\nu\big[Y^2 - 2Y\bar{f}(X) + \bar{f}^2(X)\big] \\ &= \mathbb{E}^\nu\Big[\big(Y - \bar{f}(X)\big)^2\Big] = D^\nu.\end{aligned}$$

□

3.2 Approximation of C^ν and D^ν

To approximate C^ν and D^ν, we use $\mathbb{P}^\nu$-realizations $Z^n := h(X^n, \tilde{V}^n)$, $n = 1, \ldots, N$, of Z based on independent copies $\tilde{V}^n$ of V drawn independently of $(\mathcal{X}^m, \mathcal{Y}^m)$, $m = 1, \ldots, M$, and (X^n, Y^n, V^n), $n = 1, \ldots, N$. The corresponding Monte Carlo approximations of C^ν and D^ν are

$$C_N^\nu := \frac{1}{N}\sum_{n=1}^N Y^n Z^n \quad \text{and} \quad D_N^\nu := \frac{1}{N}\sum_{n=1}^N Y^n(Y^n - Z^n), \tag{3.3}$$

respectively. If $\mathbb{E}^\nu Y^2 < \infty$, then $\mathbb{E}^\nu Z^2 < \infty$ too, and one obtains from the strong law of large numbers that

$$\lim_{N\to\infty} C_N^\nu = C^\nu \quad \text{and} \quad \lim_{N\to\infty} D_N^\nu = D^\nu \quad \mathbb{P}^\nu\text{-almost surely.}$$

Moreover, for the sample variances

$$v_N^{C,\nu} := \frac{1}{N-1}\sum_{n=1}^N \big(Y^n Z^n - C_N^\nu\big)^2 \quad \text{and} \quad v_N^{D,\nu} := \frac{1}{N-1}\sum_{n=1}^N \big(Y^n(Y^n - Z^n) - D_N^\nu\big)^2,$$

the following analog of Lemma 2.2 holds.

[2] If assumption (R) holds, e.g. the product space $(\Omega \times \Omega, \mathcal{F}\otimes\mathcal{F}, \mathbb{P}^\nu \otimes \mathbb{P}^\nu)$ supports, next to X and V, an independent copy $\tilde{V}$ of V.

Lemma 3.3 *If* $\mathbb{E}^\nu Y^4 < \infty$*, then, for every* $\alpha \in (1/2, 1)$*,*

$$\liminf_{N\to\infty} \mathbb{P}^\nu \left[\left| C^\nu - C_N^\nu \right| \le q_{1-\alpha} \sqrt{\frac{v_N^{C,\nu}}{N}} \right] \ge 1 - 2\alpha \tag{3.4}$$

and

$$\liminf_{N\to\infty} \mathbb{P}^\nu \left[\left| D^\nu - D_N^\nu \right| \le q_{1-\alpha} \sqrt{\frac{v_N^{D,\nu}}{N}} \right] \ge 1 - 2\alpha. \tag{3.5}$$

Proof If $C^\nu = YZ$ $\mathbb{P}^\nu$-almost surely, then $C^\nu - C_N^\nu = v_N^{C,\nu} = 0$ $\mathbb{P}^\nu$-almost surely for all $N \ge 1$, and (3.4) is immediate. On the other hand, if $\mathbb{P}^\nu[C^\nu \ne YZ] > 0$, one obtains from the strong law of large numbers that $v_N^{C,\nu} \to \mathrm{Var}^{\mathbb{P}^\nu}(YZ) > 0$ $\mathbb{P}^\nu$-almost surely for $N \to \infty$, and it follows from the central limit theorem together with Slutky's theorem that

$$\lim_{N\to\infty} \mathbb{P}^\nu \left[\sqrt{\frac{N}{v_N^{C,\nu}}} \left| C^\nu - C_N^\nu \right| \le q_{1-\alpha} \right] = 1 - 2\alpha.$$

This shows (3.4). Equation (3.5) follows analogously. □

3.3 L^2-Bounds on the Approximation Error

We now derive L^2-bounds on the error resulting from approximating the true regression function $\bar{f}$ with a candidate regression function $\hat{f}$. Let us denote by $L^2(\nu)$ the space of all Borel functions $f : \mathbb{R}^d \to \mathbb{R}$ satisfying

$$\|f\|_{L^2(\nu)}^2 := \mathbb{E}^\nu f^2(X) = \int_{\mathbb{R}^d} f^2(x) d\nu(x) < \infty$$

and consider the squared $L^2(\nu)$-norm of the approximation error

$$F^\nu := \|\hat{f} - \bar{f}\|_{L^2(\nu)}^2 = \mathbb{E}^\nu \left[\left(\hat{f}(X) - \bar{f}(X) \right)^2 \right]. \tag{3.6}$$

F^ν has the following alternative representation.

Theorem 3.4 *If* $\mathbb{E}^\nu Y^2 < \infty$ *and* $\mathbb{E}^\nu \hat{f}^2(X) < \infty$*, then*

$$F^\nu = \mathbb{E}^\nu \left[YZ + \hat{f}(X) \left(\hat{f}(X) - Y - Z \right) \right]. \tag{3.7}$$

Proof Since $\bar{f}(X) = \mathbb{E}^\nu[Y \mid X]$, it follows from $\mathbb{E}^\nu Y^2 < \infty$ and the conditional Jensen inequality that $\mathbb{E}^\nu \bar{f}^2(X) < \infty$ as well. Furthermore, $Y - \bar{f}(X)$ is orthogonal to $\hat{f}(X) - \bar{f}(X)$ in $L^2(\mathbb{P}^\nu)$. Therefore, one obtains from Pythagoras' theorem that

$$F^\nu = \left\| \hat{f}(X) - \bar{f}(X) \right\|_{L^2(\mathbb{P}^\nu)}^2 = \left\| Y - \hat{f}(X) \right\|_{L^2(\mathbb{P}^\nu)}^2 - \left\| Y - \bar{f}(X) \right\|_{L^2(\mathbb{P}^\nu)}^2. \tag{3.8}$$

In addition, we know from Proposition 3.2 that

$$\left\| Y - \bar{f}(X) \right\|_{L^2(\mathbb{P}^\nu)}^2 = \mathbb{E}^\nu \left[\left(Y - \bar{f}(X) \right)^2 \right] = \mathbb{E}^\nu [Y(Y - Z)]. \tag{3.9}$$

So, since

$$\mathbb{E}^{\nu}\Big[Y\hat{f}(X)\Big] = \mathbb{E}^{\nu}\Big[\mathbb{E}^{\nu}[h(X,V)\mid X]\,\hat{f}(X)\Big] = \mathbb{E}^{\nu}\Big[\mathbb{E}^{\nu}[h(X,\tilde{V})\mid X]\,\hat{f}(X)\Big] = \mathbb{E}^{\nu}\Big[Z\hat{f}(X)\Big],$$

we obtain from (3.8) and (3.9) that

$$F^{\nu} = \mathbb{E}^{\nu}\Big[Y^2 - (Y+Z)\hat{f}(X) + \hat{f}^2(X) - Y(Y-Z)\Big] = \mathbb{E}^{\nu}\Big[YZ + \hat{f}(X)\Big(\hat{f}(X) - Y - Z\Big)\Big],$$

which shows (3.7). □

In view of (3.7), we approximate F^{ν} with the Monte Carlo average

$$F_N^{\nu} := \frac{1}{N}\sum_{n=1}^{N}\Big\{Y^nZ^n + \hat{f}(X^n)\Big(\hat{f}(X^n) - Y^n - Z^n\Big)\Big\} \tag{3.10}$$

and denote the corresponding sample variance by

$$v_N^{F,\nu} := \frac{1}{N-1}\sum_{n=1}^{N}\Big\{Y^nZ^n + \hat{f}(X^n)\Big(\hat{f}(X^n) - Y^n - Z^n\Big) - F_N^{\nu}\Big\}^2.$$

The following lemma provides approximate confidence upper bounds for the true squared L^2-approximation error (3.6).

Lemma 3.5 *If $\mathbb{E}^{\nu}\,Y^4 < \infty$ and $\mathbb{E}^{\nu}\,\hat{f}^4(X) < \infty$, one has for all $\alpha \in (0,1)$,*

$$\liminf_{N\to\infty}\mathbb{P}^{\nu}\left[F^{\nu} \le F_N^{\nu} + q_{\alpha}\sqrt{\frac{v_N^{F,\nu}}{N}}\right] \ge \alpha. \tag{3.11}$$

Proof In the special case, where

$$YZ + \hat{f}(X)(\hat{f}(X) - Y - Z) = F^{\nu} \quad \mathbb{P}^{\nu}\text{-almost surely},$$

one has

$$F^{\nu} - F_N^{\nu} = v_N^{F,\nu} = 0 \quad \mathbb{P}^{\nu}\text{-almost surely for all } N \ge 1,$$

and (3.11) is clear. On the other hand, if

$$\mathbb{P}^{\nu}\Big[YZ + \hat{f}(X)(\hat{f}(X) - Y - Z) \ne F^{\nu}\Big] > 0,$$

it follows from the strong law of large numbers that

$$\lim_{N\to\infty} v_N^{F,\nu} = \mathrm{Var}^{\mathbb{P}^{\nu}}\Big(YZ + \hat{f}(X)(\hat{f}(X) - Y - Z)\Big) > 0 \quad \mathbb{P}^{\nu}\text{-almost surely}.$$

So, one obtains from the central limit theorem and Slutky's theorem that

$$\liminf_{N\to\infty}\mathbb{P}^{\nu}\left[\sqrt{\frac{N}{v_N^{F,\nu}}}\left(F^{\nu} - F_N^{\nu}\right) \le q_{\alpha}\right] = \alpha,$$

which implies (3.11). □

In applications where $\bar{f}$ needs to be approximated well at a given point x_0 in the support of the distribution ν_X of X, ν_X can be distorted so as to obtain a probability measure $\nu \ll \nu_X$ on $\mathbb{R}^d$ that concentrates more weight around x_0. Then, provided that $\hat{f} - \bar{f}$ is continuous at x_0, $\|\hat{f} - \bar{f}\|_{L^2(\nu)}$ approximates the point-wise difference $|\hat{f}(x_0) - \bar{f}(x_0)|$. More precisely, if $\|.\|_2$ denotes the standard Euclidean norm on $\mathbb{R}^d$, the following holds.

Lemma 3.6 *Assume $\mathbb{E}\, Y^2 < \infty$, $\mathbb{E}\, \hat{f}^2(X) < \infty$ and $\hat{f} - \bar{f}$ is continuous at a point $x_0 \in \mathbb{R}^d$. Let $(\nu_n)_{n\geq 1}$ be a sequence of Borel probability measures on $\mathbb{R}^d$ given by $d\nu_n/d\nu_X = p_n$ for a sequence of Borel functions $p_n \colon \mathbb{R}^d \to [0, \infty)$ satisfying*

$$\int_{\mathbb{R}^d} p_n(x) d\nu_X(x) = 1 \text{ for all } n \geq 1 \quad \text{and} \quad \sup_{x\in\mathbb{R}^d \colon \|x-x_0\|_2 > 1/n} p_n(x) \to 0 \text{ for } n \to \infty.$$

Then

$$\lim_{n\to\infty} \|\hat{f} - \bar{f}\|_{L^2(\nu_n)} = |\hat{f}(x_0) - \bar{f}(x_0)|.$$

Proof It follows from $\mathbb{E}\, Y^2 < \infty$ that $\mathbb{E}\, \bar{f}^2(X) < \infty$, which together with the condition $\mathbb{E}\, \hat{f}^2(X) < \infty$, implies that $f := \hat{f} - \bar{f} \in L^2(\nu_X)$. Moreover, one obtains from the assumptions that for every $\varepsilon > 0$, there exists an $n \geq 1$ such that

$$|f(x) - f(x_0)| \leq \varepsilon \quad \text{for all } x \in \mathbb{R}^d \text{ satisfying } \|x - x_0\|_2 \leq 1/n,$$

and

$$\begin{aligned} &\int_{\{x\in\mathbb{R}^d \colon \|x-x_0\|_2 > 1/n\}} (f(x) - f(x_0))^2 \, d\nu_n(x) \\ &= \int_{\{x\in\mathbb{R}^d \colon \|x-x_0\|_2 > 1/n\}} (f(x) - f(x_0))^2 \, p_n(x) d\nu_X(x) \leq \varepsilon^2. \end{aligned}$$

Hence,

$$\begin{aligned} \|f - f(x_0)\|^2_{L^2(\nu_n)} &= \int_{\{x\in\mathbb{R}^d \colon \|x-x_0\|_2 \leq 1/n\}} (f(x) - f(x_0))^2 \, d\nu_n(x) \\ &\quad + \int_{\{x\in\mathbb{R}^d \colon \|x-x_0\|_2 > 1/n\}} (f(x) - f(x_0))^2 \, d\nu_n(x) \leq 2\varepsilon^2, \end{aligned}$$

and therefore,

$$\big| \|f\|_{L^2(\nu_n)} - |f(x_0)| \big| \leq \|f - f(x_0)\|_{L^2(\nu_n)} \leq \sqrt{2}\, \varepsilon.$$

Since $\varepsilon > 0$ was arbitrary, this proves the lemma. □

4 Examples

In all our examples we compute a candidate regression function $\hat{f} \colon \mathbb{R}^d \to \mathbb{R}$ by minimizing an empirical mean squared distance of the form (2.2). For comparison reasons we minimize (2.2) over different families $\mathcal{S}$ of Borel measurable functions $f : \mathbb{R}^d \to \mathbb{R}$, in each case using a numerical method suited to the specific form of $\mathcal{S}$.

1. First, we use linear regression on $1, X_1, \ldots, X_d$. The corresponding function family $\mathcal{S}$ consists of all linear combinations of $1, x_1, \ldots, x_d$, and the minimization of the empirical mean squared distance (2.2) becomes the ordinary least squares problem

$$\min_{\beta\in\mathbb{R}^{d+1}} \sum_{m=1}^{M} \left(\mathcal{Y}^m - \beta_0 - \sum_{i=1}^{d} \beta_i \mathcal{X}_i^m \right)^2 .$$

This yields a candidate regression function of the form $\hat{f}(x) = \hat{\beta}_0 + \sum_{i=1}^{d} \hat{\beta}_i x_i$, where $\hat{\beta} \in \mathbb{R}^{d+1}$ is a solution of the normal equation

$$A^T A\, \hat{\beta} = A^T y \tag{4.1}$$

for $A \in \mathbb{R}^{M\times(d+1)}$ and $y \in \mathbb{R}^M$ given by

$$A = \begin{pmatrix} 1 & \mathcal{X}_1^1 & \cdots & \mathcal{X}_d^1 \\ 1 & \mathcal{X}_1^2 & \cdots & \mathcal{X}_d^2 \\ \cdots & \cdots & \cdots & \cdots \\ 1 & \mathcal{X}_1^M & \cdots & \mathcal{X}_d^M \end{pmatrix} \quad \text{and} \quad y = \begin{pmatrix} \mathcal{Y}^1 \\ \cdots \\ \cdots \\ \mathcal{Y}^M \end{pmatrix}.$$

In Sects. 4.1 and 4.2 we use $M = 2\times 10^6$ independent Monte Carlo simulations $(\mathcal{X}^m, \mathcal{Y}^m)$ for the linear regression, while in Sects. 4.3 and 4.4, where the examples are higher-dimensional, we use $M = 5 \times 10^5$ of them. If the matrix $A^T A$ is invertible, Eq. (4.1) has a unique solution given by $\hat{\beta} = (A^T A)^{-1} A^T y$. If $A^T A$ is invertible and, in addition, well-conditioned, $\hat{\beta}$ can efficiently be computed using the Cholesky decomposition $R^T R$ of $A^T A$ to solve $R^T z = A^T y$ and $R\hat{\beta} = z$ in two steps. On the other hand, if $A^T A$ is not invertible or ill-conditioned, the Cholesky method is numerically unstable. In this case, we compute a singular value decomposition $U\Sigma V^T$ of A for orthogonal matrices $U \in \mathbb{R}^{M\times M}$, $V \in \mathbb{R}^{(d+1)\times(d+1)}$ and a diagonal matrix $\Lambda \in \mathbb{R}^{M\times(d+1)}$ with diagonal entries $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{d+1} \geq 0$. The solution of (4.1) with the smallest Euclidean norm is then given by

$$\hat{\beta} = V \begin{pmatrix} \lambda_1^{-1} 1_{\{\lambda_1>0\}} & 0 & \ldots\ldots 0 \\ 0 & \lambda_2^{-1} 1_{\{\lambda_2>0\}} & \ldots\ldots 0 \\ 0 & \ldots & \ldots\ldots 0 \\ 0 & \ldots & \ldots\ldots \lambda_{d+1}^{-1} 1_{\{\lambda_{d+1}>0\}} \\ 0 & \ldots & \ldots\ldots 0 \\ \ldots & \ldots & \ldots\ldots\ldots \\ 0 & \ldots & \ldots\ldots 0 \end{pmatrix} U^T y,$$

which, for numerical stability reasons, we approximate with a truncated SVD solution

$$\hat{\beta}_c = V \begin{pmatrix} \lambda_1^{-1} 1_{\{\lambda_1>c\}} & 0 & \ldots\ldots 0 \\ 0 & \lambda_2^{-1} 1_{\{\lambda_2>c\}} & \ldots\ldots 0 \\ 0 & \ldots & \ldots\ldots 0 \\ 0 & \ldots & \ldots\ldots \lambda_{d+1}^{-1} 1_{\{\lambda_{d+1}>c\}} \\ 0 & \ldots & \ldots\ldots 0 \\ \ldots & \ldots & \ldots\ldots\ldots \\ 0 & \ldots & \ldots\ldots 0 \end{pmatrix} U^T y \tag{4.2}$$

for a small cutoff value $c > 0$; see, e.g., e.g., Björck 9.

2. As second method we use second order polynomial regression; that is, we regress on $1, X_1, \ldots, X_d$ and all second order terms $X_i X_j$, $1 \le i \le j \le d$. $\mathcal{S}$ is then the linear span of $1, x_1, \ldots, x_d$ and $x_i x_j$, $1 \le i \le j \le d$, and a canditate regression function can be computed as in 1. above, except that now the feature matrix A has $1 + 3d/2 + d^2/2$ columns. As before, we use $M = 2 \times 10^6$ independent Monte Carlo simulations $(\mathcal{X}^m, \mathcal{Y}^m)$ in Sects. 4.1–4.2 and $M = 5 \times 10^5$ of them in Sects. 4.3–4.4, and again, we use the Cholesky decomposition of $A^T A$ to solve (4.1) if $A^T A$ is well-conditioned and a truncated SVD solution otherwise.[3]
3. In our third method, $\mathcal{S}$ consists of all neural networks of a given architecture. In this paper we focus on feedforward neural networks of the form

$$f^\theta = a_D^\theta \circ \varphi \circ a_{D-1}^\theta \circ \cdots \circ \varphi \circ a_1^\theta, \tag{4.3}$$

where

- $D \ge 1$ is the depth of the network;
- $a_i^\theta : \mathbb{R}^{q_{i-1}} \to \mathbb{R}^{q_i}$, $i = 1, \ldots, D$, are affine transformations of the form $a_i^\theta(x) = A_i x + b_i$ for matrices $A_i \in \mathbb{R}^{q_i \times q_{i-1}}$ and vectors $b_i \in \mathbb{R}^{q_i}$, where q_0 is the input dimension, $q_D = 1$ the output dimension and q_i, $i = 1, \ldots, D-1$, the number of neurons in the i-th hidden layer;
- $\varphi : \mathbb{R} \to \mathbb{R}$ is a non-linear activation function applied component-wise in each hidden layer.

In the examples below, we use networks of depth $D = 4$ and 128 neurons in each of the three hidden layers. We compare the commonly used activation functions $\varphi = \tanh$ and $\mathrm{ReLU}(x) := \max\{0, x\}$ to the following smooth version of $\mathrm{LeakyReLU}(x) := \max\{\alpha x, x\}$:

$$\mathrm{LSE}(x) := \log\left(e^{\alpha x} + e^x\right) \quad \text{for } \alpha = 0.01,$$

which is efficient to evaluate numerically and, by the LogSumExp inequality, satisfies

$$\mathrm{LeakyReLU}(x) \le \mathrm{LSE}(x) \le \mathrm{LeakyReLU}(x) + \log 2.$$

In addition, LSE is everywhere differentiable with non-vanishing derivative, which alleviates the problem of vanishing gradients that can arise in the training of tanh and ReLU networks. We initialize the parameter vector θ according to Xavier initialization 24 and then optimize it by iteratively decreasing the empirical mean squared distance (2.2) with Adam stochastic gradient descent 33 using mini-batches of size 2^{13} and batch-normalization[4] 30 before each activation φ. We perform 250,000 gradient steps with standard Adam parameters, except that we start with a learning rate of 0.1, which we manually reduce to 0.05, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} and 10^{-6} after 1000, 5000, 25,000,

[3] We used Cholesky decomposition for the linear and polynomial regressions without additional feature in Sects. 4.1–4.2 and the pseudoinversion (4.2) based on truncated SVD for all other linear and polynomial regressions in Sect. 4. We computed (4.2) with a standard pseudoinverse command. In most examples the default cutoff value c gave good results. In the high-dimensional examples of Sects. 4.3–4.4 a slightly higher cutoff value c improved the results of the polynomial regressions. Alternatively, one could use ridge regression with a suitable penalty parameter or (stochastic) gradient descent to solve the least squares problem in cases where $A^T A$ is ill-conditioned.

[4] Note that while the trained network is of the form (4.3), training with batch-normalization decomposes each affine transformation into a concatenation $a_i^\theta = a_{i,2}^\theta \circ a_{i,1}^\theta$ for a general affine transformation $a_{i,1}^\theta : \mathbb{R}^{q_{i-1}} \to \mathbb{R}^{q_i}$ and a batch-normalization transformation $a_{i,2}^\theta : \mathbb{R}^{q_i} \to \mathbb{R}^{q_i}$, both of which are learned from the data. This usually stabilizes the training process but increases the number of parameters that need to be learned.

50,000, 100,000 and 150,000 iterations, respectively. To avoid slow cross device communications between the CPU and GPU, we generate all simulations on the fly during the training procedure. Since we simulate from a model, we can produce a large training set and therefore, do not need to worry about overfitting to the training data.

Remark 4.1 In many applications, the performance of the numerical regression can be improved with little additional effort by adding a redundant feature of the form $a(X)$ for a Borel measurable function $a\colon \mathbb{R}^d \to \mathbb{R}$ capturing important aspects of the relation between X and Y. For instance, if Y is given by $Y = h(X, V)$ for a Borel function $h\colon \mathbb{R}^{d+k} \to \mathbb{R}$ and a k-dimensional random vector V, adding the additional feature $a(X) = h(X, 0)$, or something similar, often yields good results. Instead of minimizing the mean squared distance (2.1), one then tries to find a Borel function $\hat{g}\colon \mathbb{R}^{d+1} \to \mathbb{R}$ that minimizes $\mathbb{E}^\nu\Big[\big(Y - \hat{g}(X, a(X))\big)^2\Big]$ and approximates the regression function $\bar{f}\colon \mathbb{R}^d \to \mathbb{R}$ with $\hat{f}(x) = \hat{g}(x, a(x))$, $x \in \mathbb{R}^d$.

In all examples, we report for all different methods used to determine a candidate regression function $\hat{f}$,

- an approximate 95% confidence interval for $U^\nu = \mathbb{E}^\nu\Big[\Big(Y - \hat{f}(X)\Big)^2\Big]$ using (2.5).
- an approximate 95% confidence interval for $D^\nu = \mathbb{E}^\nu\Big[\big(Y - \bar{f}(X)\big)^2\Big]$ using (3.5).
- an estimate of the relative error $\|\hat{f} - \bar{f}\|_{L^2(\nu)}/\|\bar{f}\|_{L^2(\nu)}$ of the form $\sqrt{F_N^\nu/C_N^\nu}$ for C_N^ν and F_N^ν given in (3.3) and (3.10), respectively. Note that while the theoretical values $C^\nu = \|\bar{f}\|_{L^2(\nu)}$ and $F^\nu = \|\hat{f} - \bar{f}\|^2_{L^2(\nu)}$ are both non-negative, in some of our examples, F^ν is close to zero. So due to Monte Carlo noise, the estimate F_N^ν can become negative. In these cases, we report $-\sqrt{-F_N^\nu/C_N^\nu}$ instead of $\sqrt{F_N^\nu/C_N^\nu}$.
- an approximate 95% confidence upper bound for the error $\|\hat{f} - \bar{f}\|_{L^2(\nu)}$ based on (3.11) expressed as a fraction of $\|\bar{f}\|_{L^2(\nu)}$ as estimated by C_N^ν given in (3.3).
- The time in seconds it took to compute the approximate regression function $\hat{f}$.

In Sects. 4.1 and 4.2 below we used $N = 6 \times 10^8$ independent Monte Carlo simulations (X^n, Y^n, Z^n) to compute the estimates U_N^ν, D_N^ν, F_N^ν, C_N^ν together with the corresponding confidence intervals, whereas in Sects. 4.3 and 4.4, due to the higher dimensionality of the examples, we only worked with $N = 6 \times 10^7$ independent Monte Carlo simulations. To fit such large test data sets into the computer memory, we split them into 6,000 independent batches of 100,000 or 10,000 data points, respectively. In most examples, we chose ν to be equal to the original distribution ν_X of X, in which case $\mathbb{E}^\nu$ equals $\mathbb{E}$.

All computations were performed on a Nvidia GeForce RTX 2080 Ti GPU together with Intel Core Xeon CPUs using `Python 3.9.6`, `TensorFlow 2.5.0` with eager mode disabled and `TensorFlow Probability 0.13.0` on `Fedora 32`.

4.1 A Four-Dimensional Polynomial Example

In our first example, we consider a simple non-linear model for (X, Y) in which the conditional expectation $\mathbb{E}[Y \mid X]$ can be computed explicitly. This enables us to benchmark our numerical results against the theoretical values. Let $X = (X_1, X_2, X_3, X_4)$ be a four-dimensional random vector and V, Y random variables such that $X_1, \ldots, X_4, V$ are i.i.d. standard normal and Y is of the form

$$Y = X_1 + X_2^2 + X_3 X_4 + V. \tag{4.4}$$

Then the conditional expectation is

$$\mathbb{E}[Y \mid X] = X_1 + X_2^2 + X_3 X_4, \tag{4.5}$$

from which the minimal mean squared distance under $\mathbb{P}$ can be seen to be

$$D^{\nu_X} = \mathbb{E}\big[(Y - \mathbb{E}[Y \mid X])^2\big] = \mathbb{E}\big[V^2\big] = 1.$$

Replacing V by 0 in the expression (4.4) would suggest to use the additional feature $a(X) = X_1 + X_2^2 + X_3 X_4$. However, since this would directly solve the problem, we are not using it in this example.

Our numerical results are listed in Table 1. More details are provided in Table 7 in the Appendix. As could be expected, since the true regression function (4.5) is a second order polynomial, the accuracy of the linear regression is poor, while the the second order polynomial regression works very well. All three neural networks provide results comparable to the one of the second order polynomial regression, albeit with more computational effort.

4.2 A Five-Dimensional Non-polynomial Example

In our second example, we consider a non-polynomial relationship between Y and X. More precisely, we let $X_1, V_1, \ldots, X_5, V_5$ be i.i.d. standard normal and assume that Y is of the form

$$Y = 5 \log\left(5 + (X_1 + V_1)^2 X_2^2 + V_2^2\right) \tanh\left((X_3 + V_3)(X_4 + V_4)(X_5 + V_5)^2\right). \tag{4.6}$$

Then the conditional expectation $\mathbb{E}[Y \mid X]$ is not known in closed form. Setting $V_1 = \cdots = V_5 = 0$ in (4.6) suggests to use the additional feature

$$a(X) = 5 \log\left(5 + X_1^2 X_2^2\right) \tanh\left(X_3 X_4 X_5^2\right). \tag{4.7}$$

4.2.1 Minimizing the Mean Squared Distance under $\mathbb{P}$

We first search for the function $f\colon \mathbb{R}^d \to \mathbb{R}$ minimizing the mean squared distance $\mathbb{E}[(Y - f(X))^2]$ under the original measure $\mathbb{P}$. The numerical results are reported in Table 2, and more details can be found in Table 8 in the Appendix. It can be seen that the second order polynomial regression yields better results than the linear regression, but now, both are clearly outperformed by the three neural network approaches. Moreover, the inclusion of the additional feature (4.7) improves the accuracy of the linear and second order polynomial regressions, while it does not increase the performance of the neural networks significantly.

4.2.2 Minimizing the Mean Squared Distance under a Distorted Measure $\mathbb{P}^{\nu}$

As a variant, we numerically minimize the mean squared distance $\mathbb{E}^{\nu}[(Y - f(X))^2]$ in the model (4.6) with respect to a distorted measure $\mathbb{P}^{\nu}$ under which $X_1, V_1 \ldots, X_5, V_5$ are independent with $X_1, \ldots X_5 \sim N(1, 1/10)$ and $V_1, \ldots, V_5 \sim N(0, 1)$. The measure ν concentrates more mass around the point $(1, \ldots, 1) \in \mathbb{R}^5$ than the original distribution ν_X of X. But since $(V_1, \ldots, V_5)$ has the same distribution as under $\mathbb{P}$, the minimizing function f coincides with the same theoretical regression function $\bar{f}$ as before. However, L^2-norms are now measured with respect to $\mathbb{P}^{\nu}$ instead of $\mathbb{P}$, which leads to different numerical results in Tables 3 and 9 compared to Tables 2 and 8. It can be seen that, as before in the $\mathbb{P}$-minimization, the three neural networks provide better results than the second order polynomial regression,

Table 1 Numerical results for the polynomial example (4.4)

	95% CI U^{ν_X}	95% CI D^{ν_X}	$\frac{\|\hat{f}-\bar{f}\|_{L^2(\nu_X)}}{\|\bar{f}\|_{L^2(\nu_X)}}$ (%)	95%CB$\frac{\|\hat{f}-\bar{f}\|_{L^2(\nu_X)}}{\|\bar{f}\|_{L^2(\nu_X)}}$ (%)	Comp. time for $\hat{f}$ (s)
Lin. regr.	[3.99957, 4.00105]	[0.99982, 1.00040]	77.46	77.47	0.1
Poly. regr.	[0.99979, 1.00002]	[0.99982, 1.00040]	0.25	0.68	0.1
NN tanh	[0.99986, 1.00009]	[0.99982, 1.00040]	0.45	0.77	1332.0
NN ReLU	[1.00007, 1.00030]	[0.99982, 1.00040]	0.79	1.01	1328.3
NN LSE	[0.99988, 1.00010]	[0.99982, 1.00040]	0.47	0.79	1483.2

Table 2 Numerical results for the non-polynomial example (4.6) regressed under $\mathbb{P}$

	95% CI U^{ν_X}	95% CI D^{ν_X}	$\frac{\|\hat{f}-\bar{f}\|_{L^2(\nu_X)}}{\|\bar{f}\|_{L^2(\nu_X)}}$ (%)	95%CB$\frac{\|\hat{f}-\bar{f}\|_{L^2(\nu_X)}}{\|\bar{f}\|_{L^2(\nu_X)}}$ (%)	Comp. time for $\hat{f}$ (s)
Lin. regr.	[41.57390, 41.58212]	[36.16566, 36.17592]	100.00	100.02	0.1
Lin. regr., add. feature	[37.89192, 37.90030]	[36.16566, 36.17592]	56.48	56.52	0.1
Poly. regr.	[36.66939, 36.67730]	[36.16566, 36.17592]	30.47	30.55	0.2
Poly. regr., add. feature	[36.39638, 36.40441]	[36.16566, 36.17592]	20.58	20.70	0.2
NN tanh	[36.16812, 36.17617]	[36.16566, 36.17592]	0.90	2.44	1410.3
NN tanh, add. feature	[36.16801, 36.17606]	[36.16566, 36.17592]	0.75	2.38	1440.5
NN ReLU	[36.16800, 36.17605]	[36.16566, 36.17592]	0.74	2.38	1399.0
NN ReLU, add. feature	[36.16775, 36.17579]	[36.16566, 36.17592]	0.15	2.27	1470.0
NN LSE	[36.16757, 36.17562]	[36.16566, 36.17592]	−0.49	2.21	1579.7
NN LSE, add. feature	[36.16764, 36.17569]	[36.16566, 36.17592]	−0.36	2.24	1532.1

Table 3 Numerical results for the non-polynomial example (4.6) regressed under $\mathbb{P}^\nu$

	95% CI U^ν	95% CI D^ν	$\frac{\|\hat{f}-\bar{f}\|_{L^2(\nu)}}{\|\bar{f}\|_{L^2(\nu)}}$ (%)	95%CB $\frac{\|\hat{f}-\bar{f}\|_{L^2(\nu)}}{\|\bar{f}\|_{L^2(\nu)}}$ (%)	Comp. time for $\hat{f}$ (s)
Lin. regr.	[39.93194, 39.94025]	[39.84392, 39.85452]	8.75	8.89	0.1
Lin. regr., add. feature	[39.92223, 39.93055]	[39.84392, 39.85452]	8.24	8.39	0.1
Poly. regr.	[39.85144, 39.85974]	[39.84392, 39.85452]	2.01	2.56	0.2
Poly. regr., add. feature	[39.85101, 39.85930]	[39.84392, 39.85452]	1.92	2.48	0.2
NN tanh	[39.84711, 39.85541]	[39.84392, 39.85452]	0.33	1.62	1373
NN tanh, add. feature	[39.84716, 39.85546]	[39.84392, 39.85452]	0.40	1.63	1449
NN ReLU	[39.84717, 39.85547]	[39.84392, 39.85452]	0.41	1.63	1390
NN ReLU, add. feature	[39.84724, 39.85554]	[39.84392, 39.85452]	0.48	1.65	1411
NN LSE	[39.84710, 39.85541]	[39.84392, 39.85452]	0.32	1.62	1515
NN LSE, add. feature	[39.84717, 39.85547]	[39.84392, 39.85452]	0.40	1.63	1561

which works better than the linear regression. But now, including the additional feature (4.7) only improves the accuracy of the linear regression slightly, while it does not help the other methods.

4.3 Max-Call Options

Different pricing and risk management problems require a conditional valuation of a financial product conditional on the state of the world at a later time [see, e.g. 5, 8, 11–13, 16, 18, 28, 34, 35, 37].

Financial market model We assume there exists a financial market consisting of a money market account offering zero interest rate and d risky securities with risk-neutral dynamics[5]

$$S_t^i = S_0^i \exp\left(\sigma_i B_t^i - \frac{1}{2}\sigma_i^2 t\right), \quad t \geq 0, \tag{4.8}$$

for initial prices $S_0^i = 10$, volatilities $\sigma_i = (10 + i/2)\%$ and Brownian motions B^i, $i = 1, \ldots, d$, with instantaneous correlation $\rho = 30\%$ between them. We denote the current time by 0 and consider a financial derivative on $S^1, \ldots, S^d$ with payoff $\phi(S_T)$ at maturity $T = 1/3$ (four months) for a payoff function $\phi\colon \mathbb{R}^d \to \mathbb{R}$. Suppose we are interested in the value of the derivative at time $t = 1/52$ (one week from now) conditional on the prices $S_t^1, \ldots, S_t^d$. According to standard no-arbitrage arguments [see, e.g., 32], it is given by $\mathbb{E}[\phi(S_T) \mid S_t]$, which can be written as $\mathbb{E}[Y \mid X]$ for $Y = \phi(S_T)$ and $X_i = S_t^i$, $i = 1, \ldots, d$. Note that Y has an explicit representation of the form (R) (see the beginning of Sect. 3) since Y can be written as $Y = h(X, V)$ for

$$h(x, v) = \phi\left(x_1 \exp\left\{v_1 - \frac{\sigma_1^2}{2}(T-t)\right\}, \ldots, x_d \exp\left\{v_d - \frac{\sigma_d^2}{2}(T-t)\right\}\right)$$

and the random variables

$$V_i = \sigma_i (B_T^i - B_t^i), \quad i = 1, \ldots, d,$$

which are independent of $X_1, \ldots, X_d$.

Let us first consider a $d = 100$-dimensional max-call option with a time-T payoff of the form

$$\phi(S_T) = \left(\max_{1 \leq i \leq d} S_T^i - K\right)^+ \tag{4.9}$$

with strike price[6] $K = 16.3$. Since the time-t price

$$\mathbb{E}\left[\left(\max_{1 \leq i \leq d} S_T^i - K\right)^+ \,\middle|\, S_t\right]$$

does not admit a closed form solution, it has to be computed numerically. In this example,

$$h(X, 0) = \left(\max_{1 \leq i \leq d} S_t^i e^{-\sigma_i^2 (T-t)/2} - K\right)^+$$

[5] We are considering a standard multi-dimensional Black–Scholes model with zero interest rate for ease of presentation. One could also use a more complicated financial market model as long as it is possible to efficiently simulate from it.

[6] The strike price 16.3 has been chosen so that approximately half of the simulated paths end up in the money at time T.

is zero with high probability. Therefore, it is not useful as an additional feature. Instead, we use the additional feature

$$a(X) = \max_{1\le i\le d} X_i = \max_{1\le i\le d} S_t^i. \tag{4.10}$$

The numerical results are reported in Table 4. Additional results are given in Table 10 in the Appendix. It can be seen that the three neural networks outperform the second order polynomial regression, which works better than the linear regression. The additional feature (4.10) does not improve the results of any of the methods significantly.

4.4 Binary Options

In our next example we consider a $d = 100$-dimensional binary option in the Financial Market Model of Sect. 4.3 with time-T payoff

$$\phi(S_T) = 10 \times 1_{\{\max_{1\le i\le d} S_T^i \ge K\}}, \tag{4.11}$$

where, as above, we choose $K = 16.3$. Again, the time-t price

$$10\,\mathbb{E}\Big[1_{\{\max_{1\le i\le d} S_T^i \ge K\}} \,\Big|\, S_t\Big] = 10\,\mathbb{P}\Big[\max_{1\le i\le d} S_T^i \ge K \,\Big|\, S_t\Big]$$

cannot be computed exactly and therefore, has to be evaluated numerically. As in Sect. 4.3, we use

$$a(X) = \max_{1\le i\le d} X_i = \max_{1\le i\le d} S_t^i \tag{4.12}$$

as additional feature.

4.4.1 Minimizing the Mean Squared Distance Under $\mathbb{P}$

We first compute a function f minimizing the mean squared distance $\mathbb{E}[(Y - f(X))^2]$ under the original measure $\mathbb{P}$. Our main numerical results are listed in Table 5. Additional results are given in Table 11 in the Appendix. Again, the three neural networks work better than the second order polynomial regression, which is more accurate than the linear regression. Adding the additional feature (4.12) does not have a significant influence on any of the methods.

4.4.2 Minimizing the Mean Squared Distance Under a Distorted Measure $\mathbb{P}^\nu$

In financial risk management, one usually is interested in the tail of a loss distribution. If a financial institution sold a contract promising a contingent payoff of $\phi(S_T)$ at time $T > 0$, the resulting exposure at time $t < T$ is $\mathbb{E}[Y \mid X] = \mathbb{E}[\phi(S_T) \mid S_t]$. To obtain a better approximation of $\bar{f} = \mathbb{E}[Y \mid X]$ with $\hat{f}(X)$ in the right tail, the least squares regression (2.2) can be performed under a measure $\mathbb{P}^\nu$ assigning more weight to the right tail of $\bar{f}(X)$ than $\mathbb{P}$. This can be done as in Cheridito et. al 18. By (4.8), $X = S_t$ can be written as $X = u(QW)$ for a d-dimensional standard normal random vector W, a $d \times d$-matrix Q satisfying

Table 4 Numerical results for the max-call option (4.9)

	95% CI U^{ν_X}	95% CI D^{ν_X}	$\frac{\|\hat{f}-\bar{f}\|_{L^2(\nu_X)}}{\|\bar{f}\|_{L^2(\nu_X)}}$ (%)	95%CB$\frac{\|\hat{f}-\bar{f}\|_{L^2(\nu_X)}}{\|\bar{f}\|_{L^2(\nu_X)}}$ (%)	Comp. time for $\hat{f}$ (s)
Lin. regr.	[6.39829, 6.40817]	[6.39167, 6.40396]	4.52	5.06	1.6
Lin. regr., add. feature	[6.39771, 6.40759]	[6.39167, 6.40396]	4.27	4.84	1.7
Poly. regr.	[6.39556, 6.40542]	[6.39167, 6.40396]	3.22	3.94	83.3
Poly. regr., add. feature	[6.39543, 6.40534]	[6.39167, 6.40396]	3.14	3.88	89.86
NN tanh	[6.39249, 6.40237]	[6.39167, 6.40396]	−1.01	2.04	1636.3
NN tanh, add. feature	[6.39255, 6.40242]	[6.39167, 6.40396]	−0.91	2.09	1647.5
NN ReLU	[6.39288, 6.40276]	[6.39167, 6.40396]	0.60	2.36	1620.1
NN ReLU, add. feature	[6.39249, 6.40237]	[6.39167, 6.40396]	−1.03	2.04	1650.5
NN LSE	[6.39276, 6.40263]	[6.39167, 6.40396]	−0.30	2.26	1807.3
NN LSE, add. feature	[6.39260, 6.40247]	[6.39167, 6.40396]	−0.83	2.13	1830.9

Table 5 Numerical results for the binary option (4.11) regressed under $\mathbb{P}$

	95% CI U^{ν_X}	95% CI D^{ν_X}	$\frac{\|\hat{f}-\bar{f}\|_{L^2(\nu_X)}}{\|\bar{f}\|_{L^2(\nu_X)}}$ (%)	95%CB$\frac{\|\hat{f}-\bar{f}\|_{L^2(\nu_X)}}{\|\bar{f}\|_{L^2(\nu_X)}}$ (%)	Comp. time for $\hat{f}$ (s)
Lin. regr.	[24.36374, 24.36761]	[24.33863, 24.36034]	2.52	2.90	4.8
Lin. regr., add. feature	[24.36276, 24.36666]	[24.33863, 24.36034]	2.45	2.84	4.8
poly. regr.	[24.35346, 24.35739]	[24.33863, 24.36034]	1.56	2.12	88.7
Poly. regr., add. feature	[24.35187, 24.35583]	[24.33863, 24.36034]	1.37	1.98	93.3
NN tanh	[24.34708, 24.35103]	[24.33863, 24.36034]	−0.12	1.43	1622.1
NN tanh, add. feature	[24.34615, 24.35011]	[24.33863, 24.36034]	−0.58	1.31	1642.5
NN ReLU	[24.34672, 24.35067]	[24.33863, 24.36034]	−0.38	1.38	1620.2
NN ReLU, add. feature	[24.34605, 24.35001]	[24.33863, 24.36034]	−0.61	1.29	1634.3
NN LSE	[24.34774, 24.35169]	[24.33863, 24.36034]	0.48	1.51	1837.4
NN LSE, add. feature	[24.34626, 24.35022]	[24.33863, 24.36034]	−0.55	1.32	1832.5

Table 6 Numerical results for the binary option (4.11) regressed under $\mathbb{P}^{\nu}$

	95% CI U^{ν}	95% CI D^{ν}	$\frac{\|\hat{f}-\bar{f}\|_{L^2(\nu)}}{\|\bar{f}\|_{L^2(\nu)}}$ (%)	95%CB$\frac{\|\hat{f}-\bar{f}\|_{L^2(\nu)}}{\|\bar{f}\|_{L^2(\nu)}}$ (%)	Comp. time for $\hat{f}$ (s)
Lin. regr.	[21.19898, 21.20766]	[21.1726, 21.19328]	2.14	2.36	4.9
Lin. regr., add. feature	[21.19584, 21.20468]	[21.1726, 21.19328]	1.98	2.21	5.2
Poly. regr.	[21.19190, 21.20076]	[21.1726, 21.19328]	1.76	2.02	89.1
Poly. regr., add. feature	[21.18934, 21.19820]	[21.1726, 21.19328]	1.60	1.88	91.5
NN tanh	[21.18205, 21.19086]	[21.1726, 21.19328]	0.99	1.40	1622.1
NN tanh, add. feature	[21.18185, 21.19064]	[21.1726, 21.19328]	0.97	1.38	1642.5
NN ReLU	[21.18194, 21.19075]	[21.1726, 21.19328]	0.98	1.39	1620.2
NN ReLU, add. feature	[21.18186, 21.19065]	[21.1726, 21.19328]	0.97	1.39	1634.3
NN LSE	[21.18296, 21.19177]	[21.1726, 21.19328]	1.09	1.47	1825.2
NN LSE, add. feature	[21.18185, 21.19065]	[21.1726, 21.19328]	0.97	1.38	1837.4

$$QQ^T = \begin{pmatrix} 1 & 0.3 & \dots & \dots & 0.3 \\ 0.3 & 1 & \dots & \dots & 0.3 \\ \dots & \dots & \dots & \dots & \dots \\ 0.3 & \dots & \dots & 1 & 0.3 \\ 0.3 & \dots & \dots & 0.3 & 1 \end{pmatrix}$$

and the function $u\colon \mathbb{R}^d \to \mathbb{R}^d$ given by

$$u(x_1, \dots, x_d) = \left(S_0^1 \exp\left(\sigma_1 \sqrt{t}\, x_1 - \frac{1}{2}\sigma_1^2 t \right), \dots, S_0^d \exp\left(\sigma_d \sqrt{t}\, x_d - \frac{1}{2}\sigma_d^2 t \right) \right).$$

Even though the regression function $\bar{f}$ is not known in closed form, it follows by monotonicity from the form of the payoff (4.11) that the mapping $\bar{f} \circ u\colon \mathbb{R}^d \to \mathbb{R}$ is increasing in the direction $v = (1, \dots, 1)^T$. So, $\bar{f}(X)$ tends to be large if $v^T QW$ is large. Therefore, we tilt $\mathbb{P}$ so that the distribution of W shifts in the direction of $Q^T v$. Let us denote by q_α the standard normal quantile at level $\alpha \in (0, 1)$. Since $v^T QW / \|v^T Q\|_2$ is one-dimensional standard normal, W lies in the region

$$G = \left\{ w \in \mathbb{R}^d : \frac{v^T Qw}{\|v^T Q\|_2} \geq q_\alpha \right\}$$

with probability $1 - \alpha$, whereas, for

$$b = \frac{Q^T v}{\|Q^T v\|_2} q_\alpha \in \mathbb{R}^d,$$

$W + b$ lies in G with probability 1/2. So if ν is the distribution of $u(Q(W + b))$, the $\mathbb{P}^\nu$-probability that $\bar{f}(X)$ is in its right $\mathbb{P}$-$(1-\alpha)$-tail is approximately 1/2. Table 6 shows results for the approximation of $\bar{f}$ under ν corresponding to $\alpha = 0.99$. More details are given in Table 12 in the Appendix. Again, the three neural networks outperform the polynomial regression, which is more accurate than the linear regression, and the inclusion of the additional feature (4.12) does not improve the performance of any of the methods significantly.

5 Conclusion

In this paper, we have studied the numerical approximation of the conditional expectation of a square-integrable random variable Y given a number of explanatory random variables $X_1, \dots, X_d$ by minimizing the mean squared distance between Y and $f(X_1, \dots, X_d)$ over a family $\mathcal{S}$ of Borel functions $f\colon \mathbb{R}^d \to \mathbb{R}$. The accuracy of the approximation depends on the suitability of the function family $\mathcal{S}$ and the performance of the numerical method used to solve the minimization problem. Using an expected value representation of the minimal mean squared distance which does not involve a minimization problem or require knowledge of the true regression function, we have derived L^2-bounds for the approximation error of a numerical solution to a given least squares regression problem. We have illustrated the method by computing approximations of conditional expectations in a range of examples using linear regression, polynomial regression as well as different neural network regressions and estimating their L^2-approximation errors. Our results contribute to trustworthy AI by providing numerical guarantees for a computational problem lying at the heart of different applications in various fields.

Funding Open access funding provided by Swiss Federal Institute of Technology Zurich The authors have not disclosed any funding.

Data Availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors have not disclosed any competing interests.

A Additional Numerical Results

In this appendix we report in Tables 7, 8, 9, 10, 11, 12 for all our numerical experiments of Sect. 4 our estimates

- U_N^{ν} of the upper bound $U^{\nu} = \mathbb{E}^{\nu}[(Y - \hat{f}(X))^2]$, see (2.4) and (2.3),
- D_N^{ν} of the minimal mean squared distance $D^{\nu} = \mathbb{E}^{\nu}[(Y - \bar{f}(X))^2]$, see (3.3) and (3.1),
- F_N^{ν} of the the squared L^2-approximation error $F^{\nu} = \|\hat{f} - \bar{f}\|^2_{L^2(\nu)}$, see (3.10) and (3.6),
- C_N^{ν} of the squared L^2-norm $C^{\nu} = \|\bar{f}\|^2_{L^2(\nu)}$, see (3.3) and (3.2),

together with the corresponding sample standard errors $\sqrt{v_N^{U,\nu}/N}$, $\sqrt{v_N^{D,\nu}/N}$, $\sqrt{v_N^{F,\nu}/N}$ and $\sqrt{v_N^{C,\nu}/N}$, which were used to compute[7] the quantities in Tables 1, 2, 3, 4, 5, 6 in Sect. 4.

Table 7 Additional numerical results for the polynomial example (4.4)

	$U_N^{\nu_X}$	$\sqrt{\frac{v_N^{U,\nu_X}}{N}}$	$D_N^{\nu_X}$	$\sqrt{\frac{v_N^{D,\nu_X}}{N}}$	$F_N^{\nu_X}$	$\sqrt{\frac{v_N^{F,\nu_X}}{N}}$	$C_N^{\nu_X}$	$\sqrt{\frac{v_N^{C,\nu_X}}{N}}$
Lin. regr.	4.00031	0.00038	1.00011	0.00015	3.00033	0.00023	5.00014	0.00049
Poly. regr.	0.99991	0.00006	1.00011	0.00015	0.00003	0.00012	5.00014	0.00049
NN tanh	0.99998	0.00006	1.00011	0.00015	0.00010	0.00012	5.00014	0.00049
NN ReLU	1.00019	0.00006	1.00011	0.00015	0.00031	0.00012	5.00014	0.00049
NN LSE	0.99999	0.00006	1.00011	0.00015	0.00011	0.00012	5.00014	0.00049

[7] To fit the numbers reported in this appendix into the tables, we had to round them. The numerical results in Sect. 4 were computed from slightly more precise approximations of U_N^{ν}, D_N^{ν}, F_N^{ν}, C_N^{ν} and their standard errors.

Table 8 Additional numerical results for the non-polynomial example (4.6) regressed under $\mathbb{P}$

	$U_N^{\nu_X}$	$\sqrt{\frac{v_N^{U,\nu_X}}{N}}$	$D_N^{\nu_X}$	$\sqrt{\frac{v_N^{D,\nu_X}}{N}}$	$F_N^{\nu_X}$	$\sqrt{\frac{v_N^{F,\nu_X}}{N}}$	$C_N^{\nu_X}$	$\sqrt{\frac{v_N^{C,\nu_X}}{N}}$
Lin. regr.	41.57801	0.00210	36.17079	0.00262	5.40722	0.00171	5.40705	0.00187
Lin. regr., add. feature	37.89611	0.00214	36.17079	0.00262	1.72471	0.00172	5.40705	0.00187
Poly. regr.	36.67335	0.00202	36.17079	0.00262	0.50198	0.00168	5.40705	0.00187
Poly. regr., add. feature	36.40040	0.00205	36.17079	0.00262	0.22893	0.00169	5.40705	0.00187
NN tanh	36.17214	0.00205	36.17079	0.00262	0.00044	0.00169	5.40705	0.00187
NN tanh, add. feature	36.17204	0.00205	36.17079	0.00262	0.00030	0.00169	5.40705	0.00187
NN ReLU	36.17202	0.00205	36.17079	0.00262	0.00029	0.00169	5.40705	0.00187
NN ReLU, add. feature	36.17177	0.00205	36.17079	0.00262	0.00001	0.00169	5.40705	0.00187
NN LSE	36.17160	0.00205	36.17079	0.00262	−0.00013	0.00169	5.40705	0.00187
NN LSE, add. feature	36.17166	0.00205	36.17079	0.00262	−0.00007	0.00169	5.40705	0.00187

Table 9 Additional numerical results for the non-polynomial example regressed under $\mathbb{P}^\nu$

	U_N^ν	$\sqrt{\frac{v_N^{U,\nu}}{N}}$	D_N^ν	$\sqrt{\frac{v_N^{D,\nu}}{N}}$	F_N^ν	$\sqrt{\frac{v_N^{F,\nu}}{N}}$	C_N^ν	$\sqrt{\frac{v_N^{C,\nu}}{N}}$
Lin. regr.	39.93609	0.00212	39.84922	0.00271	0.08475	0.00169	11.05717	0.00209
Lin. regr., add. feature	39.92639	0.00212	39.84922	0.00271	0.07508	0.00169	11.05717	0.00209
Poly. regr.	39.85559	0.00212	39.84922	0.00271	0.00446	0.00169	11.05717	0.00209
Poly. regr., add. feature	39.85515	0.00211	39.84922	0.00271	0.00406	0.00169	11.05717	0.00209
NN tanh	39.85126	0.00212	39.84922	0.00271	0.00012	0.00169	11.05717	0.00209
NN tanh, add. feature	39.85131	0.00212	39.84922	0.00271	0.00018	0.00169	11.05717	0.00209
NN ReLU	39.85132	0.00212	39.84922	0.00271	0.00018	0.00169	11.05717	0.00209
NN ReLU, add. feature	39.85139	0.00212	39.84922	0.00271	0.00026	0.00169	11.05717	0.00209
NN LSE	39.85125	0.00212	39.84922	0.00271	0.00012	0.00169	11.05717	0.00209
NN LSE, add. feature	39.85132	0.00212	39.84922	0.00271	0.00018	0.00169	11.05717	0.00209

Table 10 Additional numerical results for the max-call option (4.9)

	$U_N^{\nu_X}$	$\sqrt{\frac{v_N^{U,\nu_X}}{N}}$	$D_N^{\nu_X}$	$\sqrt{\frac{v_N^{D,\nu_X}}{N}}$	$F_N^{\nu_X}$	$\sqrt{\frac{v_N^{F,\nu_X}}{N}}$	$C_N^{\nu_X}$	$\sqrt{\frac{v_N^{C,\nu_X}}{N}}$
Lin. regr.	6.40323	0.00252	6.39782	0.00313	0.00552	0.00086	2.70728	0.00119
Lin. regr., add. feature	6.40265	0.00252	6.39782	0.00313	0.00494	0.00086	2.70728	0.00119
Poly. regr.	6.40049	0.00252	6.39782	0.00313	0.00280	0.00086	2.70728	0.00119
Poly. regr., add. feature	6.40038	0.00253	6.39782	0.00313	0.00267	0.00086	2.70728	0.00119
NN tanh	6.39743	0.00252	6.39782	0.00313	−0.00028	0.00086	2.70728	0.00119
NN tanh, add. feature	6.39749	0.00252	6.39782	0.00313	−0.00023	0.00086	2.70728	0.00119
NN ReLU	6.39782	0.00252	6.39782	0.00313	0.00010	0.00086	2.70728	0.00119
NN ReLU, add. feature	6.39743	0.00252	6.39782	0.00313	−0.00029	0.00086	2.70728	0.00119
NN LSE	6.39769	0.00252	6.39782	0.00313	−0.00002	0.00086	2.70728	0.00119
NN LSE, add. feature	6.39754	0.00252	6.39782	0.00313	−0.00018	0.00086	2.70728	0.00119

Table 11 Additional numerical results for the binary option (4.11) regressed under $\mathbb{P}$

	$U_N^{\nu_X}$	$\sqrt{\frac{v_N^{U,\nu_X}}{N}}$	$D_N^{\nu_X}$	$\sqrt{\frac{v_N^{D,\nu_X}}{N}}$	$F_N^{\nu_X}$	$\sqrt{\frac{v_N^{F,\nu_X}}{N}}$	$C_N^{\nu_X}$	$\sqrt{\frac{v_N^{C,\nu_X}}{N}}$
Lin. regr.	24.36567	0.00099	24.34948	0.00554	0.01644	0.00322	25.87654	0.00565
Lin. regr., add. feature	24.36471	0.00099	24.34948	0.00554	0.01553	0.00322	25.87654	0.00565
Poly. regr.	24.35543	0.00100	24.34948	0.00554	0.00631	0.00322	25.87654	0.00565
Poly. regr., add. feature	24.35385	0.00101	24.34948	0.00554	0.00483	0.00322	25.87654	0.00565
NN tanh	24.34906	0.00101	24.34948	0.00554	−0.00003	0.00322	25.87654	0.00565
NN tanh, add. feature	24.34813	0.00101	24.34948	0.00554	−0.00088	0.00322	25.87654	0.00565
NN ReLU	24.34869	0.00101	24.34948	0.00554	−0.00038	0.00322	25.87654	0.00565
NN ReLU, add. feature	24.34803	0.00101	24.34948	0.00554	−0.00098	0.00322	25.87654	0.00565
NN LSE	24.34972	0.00101	24.34948	0.00554	0.00059	0.00322	25.87654	0.00565
NN LSE, add. feature	24.34824	0.00101	24.34948	0.00554	−0.00077	0.00322	25.87654	0.00565

Table 12 Additional numerical results for the binary option (4.11) regressed under $\mathbb{P}^{\nu}$

	U_N^{ν}	$\sqrt{\frac{v_N^{U,\nu}}{N}}$	D_N^{ν}	$\sqrt{\frac{v_N^{D,\nu}}{N}}$	F_N^{ν}	$\sqrt{\frac{v_N^{F,\nu}}{N}}$	C_N^{ν}	$\sqrt{\frac{v_N^{C,\nu}}{N}}$
Lin. regr.	21.20332	0.00221	21.18294	0.00528	0.02149	0.00279	46.90835	0.00644
Lin. regr., add. feature	21.20026	0.00226	21.18294	0.00528	0.01841	0.00279	46.90835	0.00644
poly. regr.	21.19633	0.00226	21.18294	0.00528	0.01455	0.00279	46.90835	0.00644
Poly. regr., add. feature	21.19377	0.00226	21.18294	0.00528	0.01198	0.00279	46.90835	0.00644
NN tanh	21.18645	0.00225	21.18294	0.00528	0.00466	0.00279	46.90835	0.00644
NN tanh, add. feature	21.18624	0.00224	21.18294	0.00528	0.00442	0.00279	46.90835	0.00644
NN ReLU	21.18635	0.00225	21.18294	0.00528	0.00454	0.00279	46.90835	0.00644
NN ReLU, add. feature	21.18625	0.00224	21.18294	0.00528	0.00443	0.00279	46.90835	0.00644
NN LSE	21.18736	0.00225	21.18294	0.00528	0.00555	0.00279	46.90835	0.00644
NN LSE, add. feature	21.18625	0.00224	21.18294	0.00528	0.00442	0.00279	46.90835	0.00644

References

1. Acerbi, C., Tasche, D.: On the coherence of expected shortfall. J. Bank. Financ. **26**, 1487–1503 (2002)
2. Åström, K.J.: Introduction to Stochastic Control Theory. Mathematics in Science and Engineering, vol. 70. Academic Press, New York, London (1970)
3. Bain, A., Crisan, D.: Fundamentals of Stochastic Filtering, vol. 60. Springer (2008)
4. Bally, V.: Approximation scheme for solutions of BSDE. In: Pitman Research Notes in Mathematics Series, vol. 364. Longman (1997)
5. Bauer, D., Reuss, A., Singer, D.: On the calculation of the solvency capital requirement based on nested simulations. ASTIN Bull. **42**, 453–499 (2012)
6. Beck, C., Becker, S., Cheridito, P., Jentzen, A., Neufeld, A.: Deep learning based numerical approximation algorithms for stochastic partial differential equations and high-dimensional nonlinear filtering problems. arXiv:2012.01194 (2020)
7. Beck, C., Becker, S., Cheridito, P., Jentzen, A., Neufeld, A.: Deep splitting method for parabolic PDEs. SIAM J. Sci. Comput. **43**(5), A3135–A3154 (2021)
8. Becker, S., Cheridito, P., Jentzen, A.: Pricing and hedging American-style options with deep learning. J. Risk Financ. Manag. **13**(7), 158, 1–12 (2020)
9. Björck, Å.: Numerical Methods for Least Squares Problems. SIAM, Philadelphia (1996)
10. Bouchard, B., Touzi, N.: Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations. Stoch. Process. Appl. **11**(2), 175–206 (2004)
11. Broadie, M., Cao, M.: Improved lower and upper bound algorithms for pricing American options by simulation. Quant. Finance **8**, 845–861 (2008)
12. Broadie, M., Glasserman, P.: A stochastic mesh method for pricing high-dimensional American options. J. Comput. Financ. **7**, 35–72 (2004)
13. Broadie, M., Yiping, D., Moallemi, C.C.: Efficient risk estimation via nested sequential simulation. Manage. Sci. **57**, 1172–1194 (2011)
14. Broadie, M., Yiping, D., Moallemi, C.C.: Risk estimation via regression. Oper. Res. **63**, 1077–1097 (2015)
15. Bru, B., Heinich, H.: Meilleures approximations et médianes conditionnelles. Ann. l'IHP Probab. Stat. **21**, 197–224 (1985)
16. Carriere, J.F.: Valuation of the early-exercise price for options using simulations and nonparametric regression. Insurance Math. Econom. **19**, 19–30 (1996)
17. Chatterjee, S., Hadi, A.S.: Regression Analysis by Example. Wiley (2015)
18. Cheridito, P., Ery, J., Wüthrich, M.V.: Assessing asset-liability risk with neural networks. Risks **8**(1), 16, 1–17 (2020)
19. Chevance, D.: Numerical methods for backward SDEs. In: Numerical Methods in Finance, vol. 232 (1997)
20. Draper, N.R., Smith, H.: Applied Regression Analysis. Wiley, New York (1998)
21. Fahim, A., Touzi, N., Warin, X.: A probabilistic numerical method for fully nonlinear parabolic PDEs. Ann. Appl. Probab. **21**(4), 1322–1364 (2011)
22. Föllmer, H., Schied, A.: Stochastic Finance. De Gruyter Textbook (2016)
23. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: Bayesian Data Analysis. CRC Press (2013)
24. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)
25. Gobet, E., Turkedjiev, P.: Linear regression MDP scheme for discrete backward stochastic differential equations under general conditions. Math. Comput. **85**, 1359–1391 (2006)
26. Gobet, E., Lemor, J.-P., Warin, X.: A regression-based Monte Carlo method to solve backward SDEs. Ann. Appl. Probab. **15**, 2172–2202 (2005)
27. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning, vol. 1. MIT Press, Cambridge (2016)
28. Gordy, M.B., Juneja, S.: Nested simulation in portfolio risk measurement. Manage. Sci. **56**, 1833–1848 (2010)
29. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer (2009)
30. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning, vol. 37, pp. 448–456 (2015)
31. Jazwinski, A.H.: Stochastic Processes and Filtering Theory. Courier Corporation (2007)
32. Karatzas, I., Shreve, S.E.: Methods of Mathematical Finance. Springer (1998)
33. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv:1412.6980 (2014)

34. Lee, S.-H., Glynn, P.W.: Computing the distribution function of a conditional expectation via Monte Carlo: discrete conditioning spaces. ACM Trans. Model. Comput. Simul. **13**, 238–258 (2003)
35. Longstaff, F.A., Schwartz, E.S.: Valuing American options by simulation: a simple least-squares approach. Rev. Financ. Stud. **14**, 113–147 (2001)
36. Ryan,T.P.: Modern regression methods. Wiley Series in Probability and Statistics, 2nd edn. Wiley, Hoboken (2009)
37. Tsitsiklis, J.N., Van Roy, B.: Regression methods for pricing complex American-style options. IEEE Trans. Neural Netw. **12**, 694–703 (2001)