‘Regression anytime’ with brute force SVD truncation

Christian Bender
(joint work with Nikolaus Schweizer, Tilburg)

Saarland University

Annecy, June 2022

Outline

Aim of the talk

- The aim of the talk is to convince you that simulation based least-squares regression can work for solving backward SDEs in moderate dimensions, if the number of simulated paths is proportional to the number of basis functions (up to a log-factor).

Time discretization

- Typical situation: Dependence on ω in the coefficients of a BSDE driven by a Bm W is via a stochastic differential equation, which can be discretized by an Euler scheme X_i.
- Then typical time discretization schemes with step size $h > 0$ boil down to alternating between
  1. Solving numerically a regression problem with Malliavin weight of the form

$$m(x) = E[\beta_{i+1} y_{i+1}(X_{i+1}) | X_i = x].$$

     where

$$\beta_{i+1} \in \left\{ 1, \frac{W_{(i+1)h} - W_{ih}}{h} \right\}$$

  2. Applying a nonlinear deterministic function.
- In this talk, we focus on the analysis of one regression step.

Setting

- Two time steps:

$$X_1 := X \quad \text{'now'}$$

where X is an $\mathbb{R}^D$-valued random variable whose density has a Gaussian tail estimate.

$$X_2 = X_1 + b(X_1)h + \sigma(X_1)\sqrt{h}W \quad \text{'later'}$$

where b, σ are bounded deterministic functions, W is a vector of D independent standard normal random variables independent of X_1, $h > 0$.

- Regression problem:

$$m(x) = E\left[\frac{W}{\sqrt{h}} y(X_2) \middle| X_1 = x\right],$$

where y is of class $\mathcal{C}_b^{Q+1}$ for $Q \geq 3$.

Setting

- Recall:

$$m(x) = E\left[\frac{W}{\sqrt{h}} y(x + b(x)h + \sigma(x)\sqrt{h}W)\right]$$

- Integration by parts yields

$$m(x) = \sigma(x)^\top E[\nabla y(x + b(x)h + \sigma(x)\sqrt{h}W)]$$

- Thus, by a Taylor expansion,

$$m(x) = \sigma(x)^\top \nabla y(x) + O(h)$$

Empirical regression

Regression now:

- D Empirical (simulation-based) regressions of $\frac{W^{(d)}}{\sqrt{h}} y(X_2)$ on basis functions that depend on X_1 (i.e. 'now').
- Standard approach in statistical learning, but with simulated data instead of empirical data.
- See e.g. Lemor, Gobet, Warin (2006) in the context of BSDE numerics.

Regression later:

- Empirical regression of $y(X_2)$ on basis functions depending on X_2 (i.e. 'later') plus closed-form expressions for the conditional expectations of the weighted basis functions.
- Exploits that one knows (in principle) the distribution of the simulated data.
- See e.g. Glasserman, Yu (2004), B., Steiner (2012), Beutner, Schweizer, Pelsser (2013).

Regression anytime

- Choose basis functions that depend on (X_1, X_2) ('anytime'), cp. the stochastic grid bundling method of Oosterlee and co-authors.
- Step 1: Simulate L independent copies $(X_{1,l}, X_{2,l})$ of (X_1, X_2)
- Step 2: Choose K basis functions $\eta_1(x_1, x_2), \dots \eta_K(x_1, x_2)$. We always choose basis functions in product form

$$\eta_k(x_1, x_2) = \eta_k^{now}(x_1)\eta_k^{later}(x_2)$$

and assume that

$$x \mapsto E[W\eta_k^{later}(x + \Sigma W)] =: \tilde{\eta}_k^{later}(x; \Sigma)$$

is available in closed form for every $D \times D$-matrix Σ (take e.g. polynomials).

Regression anytime

- Step 3: Perform an empirical regression of $y(X_2)$ on the basis functions, i.e. define

$$\hat{y}^L(x_1, x_2) = \sum_{k=1}^{K} \hat{\alpha}_k \eta_k^{now}(x_1) \eta_k^{later}(x_2)$$

where $\hat{\alpha}$ is a minimizer in $\mathbb{R}^K$ of

$$\frac{1}{L} \sum_{l=1}^{L} \left(y(X_{2,l}) - \sum_{k=1}^{K} \alpha_k \eta_k^{now}(X_{1,l}) \eta_k^{later}(X_{2,l}) \right)^2$$

Regression anytime

- Step 4: Define

$$\begin{aligned}
\hat{m}^L(x) &:= E\left[\frac{W}{\sqrt{h}}\hat{y}^L(X_1, X_2)\middle| X_1 = x, (X_{1,l}, X_{2,l})_{l=1,\dots L}\right] \\
&= \sum_{k=1}^{K} \hat{\alpha}_k \eta_k^{now}(x) E\left[\frac{W}{\sqrt{h}}\eta_k^{later}(X_2)\middle| X_1 = x\right] \\
&= \sum_{k=1}^{K} \hat{\alpha}_k \eta_k^{now}(x) \frac{1}{\sqrt{h}}\tilde{\eta}_k^{later}(x + b(x)h, \sigma(x)\sqrt{h})
\end{aligned}$$

as estimator for the regression function m.

Regression anytime – Removing the weight

- Removing the weight from the error analysis: By Hölder's inequality:

$$
\begin{aligned}
& E[|m(X_1) - \hat{m}^L(X_1)|^2] \\
= \; & E\left[\left|E\left[\frac{W}{\sqrt{h}}(y(X_2) - \hat{y}^L(X_1, X_2))\right| X_1, (X_{1,l}, X_{2,l})_{l=1,\dots L}\right]\right|^2\Bigg] \\
\leq \; & \frac{D}{h} E[|y(X_2) - \hat{y}^L(X_1, X_2)|^2]
\end{aligned}
$$

Regression anytime – SVD truncation

- Consider the empirical regression matrix

$$A = (\eta_k^{now}(X_{1,l})\eta_k^{later}(X_{2,l}))_{l=1,\ldots,L;\, k=1,\ldots,K}$$

and recall that

$$\hat{\alpha} = A^{\dagger} \begin{pmatrix} y(X_{2,1}) \\ \vdots \\ y(X_{2,L}) \end{pmatrix}$$

where $A^{\dagger}$ denotes the pseudoinverse of A

- Without stabilization the convergence properties of the empirical regression may deteriorate due to rare samples that lead to a very ill-conditioned empirical regression matrix.

Regression anytime – SVD truncation

- Stabilization is usually achieved by truncating the estimator

$$\min\left\{\max\left\{-C, \sum_{k=1}^{K} \hat{\alpha}_k \eta_k^{now}(x_1)\eta_k^{later}(x_2)\right\}, C\right\}$$

  for some sufficiently large constant, say $C \geq \sup_x |y(x)|$.
- Convergence analysis for truncated least-squares estimators can be found in the textbook by Györfi et al. (2002) in the presence of noise and in Cohen, Davenport, Leviatan (2013) in a noiseless setting with orthonormal basis functions.
- However, closed-form computations of the conditional expectation in our setting require linearity of the estimator in the basis functions, which is destroyed by truncation.

Regression anytime – SVD truncation

- Way-out: Set the estimator to zero, if the smallest singular value $s_{\min}(A)$ of the empirical regression matrix ist too close to zero, cp. the conditioned least-squares estimator of Cohen and Migliorati (2017).
- By slight abuse of notation:

$$\hat{y}^L(x_1, x_2) := \sum_{k=1}^{K} \hat{\alpha}_k \eta_k(x_1, x_2)$$

where

$$\hat{\alpha}_k := (A^\top A)^{-1} A^\top \begin{pmatrix} y(X_{2,1}) \\ \vdots \\ y(X_{2,L}) \end{pmatrix} \mathbf{1}_{\{s^2_{\min}(A) \geq L\tau\}}$$

for some threshold $\tau > 0$.

Regression anytime – SVD truncation

- Statistical error decays exponentially in the sample size L and depends on a sup-bound of the basis functions

$$\sup_{(x_1,x_2)} \sum_{k=1}^{K} |\eta_k(x_1, x_2)|^2$$

and on the smallest and largest eigenvalues $\lambda_{\min}(R)$ and $\lambda_{\max}(R)$ of

$$R = (E[\eta_k(X_1, X_2)\eta_\kappa(X_1, X_2)])_{k,\kappa=1,\ldots,K}$$

- Note

$$\frac{1}{L} s^2_{\min}(A) \to \lambda_{\min}(R)$$

almost surely as $L \to \infty$.

- So the threshold τ must be a strict lower bound of $\lambda_{\min}(R)$.

Basis choice

- Recall: We wish to approximate

$$m(x) = E\left[\frac{W}{\sqrt{h}} y(X_2) \middle| X_1 = x\right]$$

  up to order, say, $O(h)$, where X_2 is one step of an Euler scheme with step size h starting at X_1.
- Then, $\hat{y}^L$ must approximate y to the order $O(h^{3/2})$.
- We need to identify an 'anytime'-function basis such that
  1. it is generically applicable to the Euler scheme setting (not tailored to the coefficients b, σ);
  2. closed-form expression of the conditional expectations of the 'later' basis functions is available;
  3. the projection error is of order $O(h^{3/2})$;
  4. the eigenvalues of $R = R_h$ and the sup-norm of the basis functions can be controlled to match the statistical error.

The RawBfst algorithm – Overview

Algorithm:

- Truncate the domain of X_1 in accordance with the Gaussian tail bound.
- Decompose the truncated domain into cubes $(\Gamma_i)_{i \in I}$ of diameter $\sim h^{3/(2Q+2)}$, $Q \geq 3$.
- Basis functions of the form

$$\eta(X_1, X_2) = \mathbf{1}_{\Gamma_i}(X_1)\mathcal{P}(X_2)$$

  where $\mathcal{P}$ are Legendre polynomials of degree up to Q, scaled to be orthonormal w.r.t the uniform distribution on Γ_i.
- Change the sampling distribution of X_1 to a (stratified) uniform distribution on the cubic grid (via importance sampling) and truncate the Gaussian innovations in the sampling scheme for X_2.
- Run 'Regression anytime' with SVD truncation based on a sample of size L to compute $\hat{y}_L$ and $\hat{m}_L$.

The RawBfst algorithm – Convergence

Theorem

Suppose $y \in \mathcal{C}_b^{Q+1}(\mathbb{R}^D)$ *for some* $Q \geq 3$. *Compute* $\hat{m}_L$ *via RawBfst with*

$$\begin{aligned} L &= L_h = \lceil 2\, c_{1,\text{paths}}\ \log(h^{-1}) \rceil \cdot |I| \\ \tau &\in \left(0,\quad 1 - \left(\frac{c^*_{\text{paths}}(Q,D)}{c_{1,\text{paths}}}\right)^{1/2}\right) \end{aligned}$$

$$c_{1,\text{paths}} > c^*_{\text{paths}}(Q,D) := \frac{2}{3} + \frac{8}{3} \sum_{\mathbf{j} \in \mathbb{N}_0^D; |\mathbf{j}|_1 \leq Q} \prod_{d=1}^{D} (2j_d + 1).$$

Then there is a constant $C > 0$ *such that for small* h

$$E\left[\left|E\left[\frac{W}{\sqrt{h}} y(X_2) \middle| X_1\right] - \hat{m}_L(X_1)\right|^2\right] \leq C \log(h^{-1})^{D/2}\ h^2.$$

The RawBfst algorithm – Convergence

Remarks:

- The cost to achieve a root-mean-squared error of the order h is up to a log-factor of the order

$$|I| \sim h^{-3D/(2Q+2)}$$

- Ignoring log-factors the convergence behaviour in the number of samples is

$$L^{-\frac{2(Q+1)}{3D}}$$

- It beats the Monte-Carlo rate of $1/2$ for computing a single expectation, if the smoothness-to-dimension ratio $(Q+1)/D$ exceeds $3/4$.
- In practice, the algorithm can only be applied in moderate dimensions and for moderate polynomial degrees.

Numerical illustration

- Test example from Gobet et al. (2016):

$$
\begin{aligned}
X_t &= W_t \quad \text{D-dim. Brownian motion} \\
Y_t &= Y_1 + \int_t^1 \left(\sum_{d=1}^{D} Z_s^{(d)} \right) \left(Y_s - \frac{1}{D} - \frac{1}{2} \right) ds - \int_t^1 Z_s dW_s \\
Y_1 &= \frac{\exp\{1 + \sum_{d=1}^{D} W_1^{(d)}\}}{1 + \exp\{1 + \sum_{d=1}^{D} W_1^{(d)}\}}
\end{aligned}
$$

- Closed form solution available: $Y_0 = 1/2$.
- We apply the time-discretization scheme by Fahim et al. (2011).

Numerical illustration

- We calibrate the RawBfst algorithm to achieve a convergence rate of $1/2$ in the time step h in accordance with the Euler discretization of Y – applying heuristics for the error propagation over the time steps.
- Dimension: 5
- Total number of cubes: $\sim h^{-(5/4+1)}$,
- number of basis functions per cube: 56 (degree up to 3)
- Number of samples per cube: $2 \cdot 4320 \log(0.5\, h^{-1})$
- Comparison: Calibration of the 'regression now'-algorithm of Gobet et al. (2016) with the same number of cubes requires $\sim h^{-3}$ samples per cube (but with a lower polynomial degree).
- Sample: one D-dimensional uniform or Gaussian random variable.

Numerical illustration

h^{-1}	mean	standard deviation
10	0.486427	$5.01 \cdot 10^{-4}$
20	0.493735	$2.52 \cdot 10^{-4}$
30	0.497602	$1.34 \cdot 10^{-4}$
40	0.499836	$8.33 \cdot 10^{-5}$
50	0.501483	$8.01 \cdot 10^{-5}$
60	0.501333	$8.01 \cdot 10^{-5}$
70	0.501016	$5.77 \cdot 10^{-5}$

Table: Mean and standard deviation of the approximation for Y_0 across 20 runs of the algorithm.

Numerical illustration

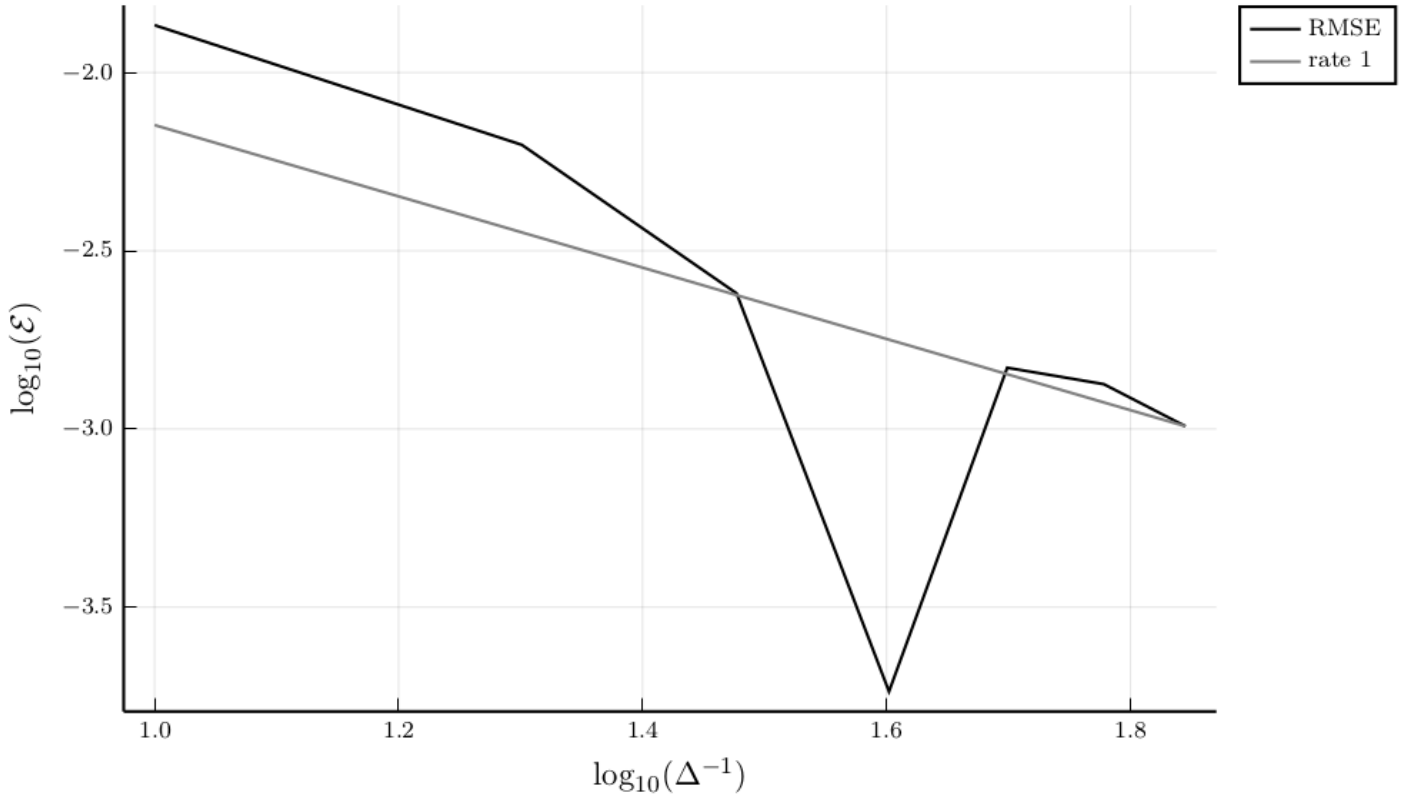

Figure: Approximation errors against time step size ($\Delta := h$) in a $\log_{10}$-$\log_{10}$-plot.

Numerical illustration

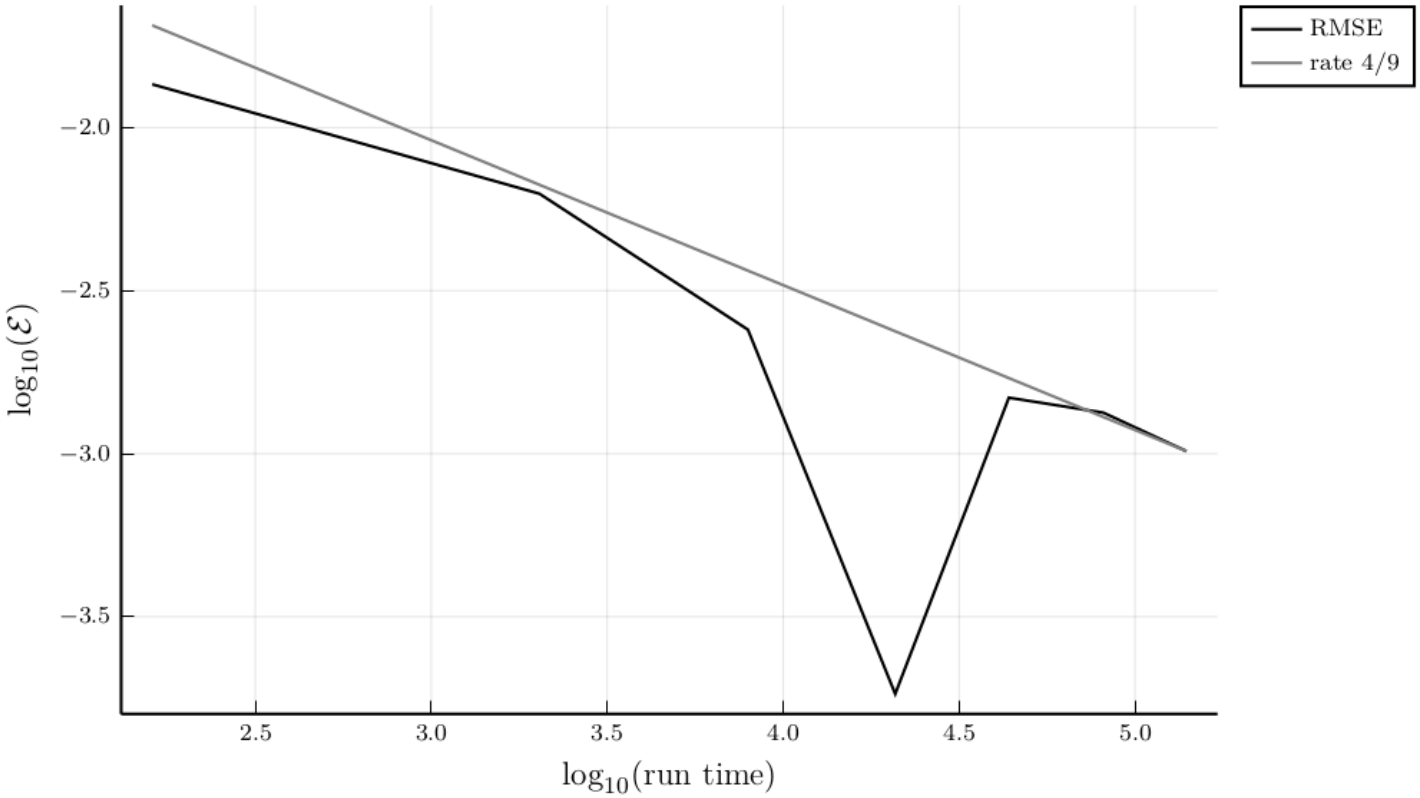

Figure: Approximation errors against run time in a $\log_{10}$-$\log_{10}$-plot. Run times are for a Julia 1.4.2 implementation on a Windows desktop PC with an Intel Core i7-6700 CPU with 3.4GHz.

Some references

- Bender, C. and Steiner, J. (2012). Least squares Monte Carlo for BSDEs. In: Carmona, R. et al. (eds.) *Numerical Methods in Finance*, Springer, Berlin, pp. 257–289.
- Beutner, E., Schweizer, J. and Pelsser, A. (2013). Fast convergence of regress-later estimates in least squares Monte Carlo. arXiv.
- Bouchard, B. and Touzi, N. (2004). Discrete-time approximation and Monte Carlo simulation of backward stochastic differential equations. *Stochastic Process. Appl.* **111** 175–206.
- Cohen, A., Davenport, M. A. and Leviatan, D. (2013). On the stability and accuracy of least squares approximations. *Found. Comput. Math.* **13** 819–834.
- Cohen, A. and Migliorati, G. (2017). Optimal weighted least-squares methods. *SMAI J. Comput. Math.* **3** 181–203.
- Fahim, A., Touzi, N. and Warin, X. (2011). A probabilistic numerical method for fully nonlinear parabolic PDEs. *Ann. Appl. Probab.* **21** 1322–1364.
- Glasserman, P. and Yu, B. (2004). Simulation for American options: Regression now or regression later? In: Niederreiter, H. (ed.) *Monte Carlo and Quasi-Monte Carlo Methods 2002*, Springer, Berlin, pp. 213–226.

Some references

- Gobet, E., López-Salas, J. G., Turkedjiev, P. and Vázquez, C. (2016). Stratified regression Monte-Carlo scheme for semilinear PDEs and BSDEs with large scale parallelization on GPUs. *SIAM J. Sci. Comput.* **38** C652–C677.
- Györfi, L., Kohler, M., Krzyzak, A. and Walk, H. (2002). *A Distribution-Free Theory of Nonparametric Regression.* Springer, New York.
- Jain, S. and Oosterlee, C. W. (2015). The stochastic grid bundling method: efficient pricing of Bermudan options and their Greeks. *Appl. Math. Comput.* **269** 412–431.
- Lemor, J.-P., Gobet, E. and Warin, X. (2006). Rate of convergence of an empirical regression method for solving generalized backward stochastic differential equations, *Bernoulli* **12** 889–916.
- Tropp, J. A. (2012). User-friendly tail bounds for sums of random matrices. *Found. Comput. Math.* **12** 389–434.
- Zhang, J. (2004). A numerical scheme for BSDEs. *Ann. Appl. Probab.* **14** 459–488.

Thank you...

... for your attention!

This talk was based on

- Bender, C. and Schweizer, N. (2021) 'Regression Anytime' with Brute-Force SVD Truncation. *Ann. Appl. Probab.*, **31**, 1140–1179.

Regression anytime – SVD truncation

Theorem

Suppose that the basis functions η_k are bounded. Let

$$\lambda_* \leq \lambda_{\min}(R) \leq \lambda_{\max}(R) \leq \lambda^*$$

and $\tau = (1-\epsilon)\lambda_*$ for some $\epsilon \in (0,1)$. Then,

$$\begin{aligned} & E\left[|y(X_2) - \hat{y}_L(X_1, X_2)|^2\right] \\ \leq \quad & \left(1 + \frac{\lambda^*}{\lambda_*(1-\epsilon)}\right) \inf_{\alpha \in \mathbb{R}^K} E\left[|y(X_2) - \alpha^\top \eta(X_1, X_2)|^2\right] \\ & + 2K \exp\left\{\frac{-3\epsilon^2 L}{6m\lambda^*/\lambda_*^2 + 2\epsilon(m/\lambda_* + \lambda^*/\lambda_*)}\right\} E[|y(X_2)|^2], \end{aligned}$$

Extends related results by Cohen and co-authors beyond the case of orthonormal basis functions.

Regression anytime – SVD truncation

Remarks:

- For a fixed function basis, the statistical error converges exponentially in the number of samples L.
- The key step is to estimate the SVD truncation probability by a matrix Bernstein inequality, see e.g. Tropp (2012).
- The result is not distribution free, but depends on the distribution of (X_1, X_2) via the eigenvalues $\lambda_{\min}(R)$, $\lambda_{\max}(R)$.
- Optimal rates (up to log-factors) for some interpolation problems with random design can be derived from this result.
- The choice of the truncation threshold τ is a trade-off between projection error and statistical error.