Perspectives on stochastic gradient descent

Jonas Latz

Department of Mathematics, University of Manchester

SIAG/UQ Early Career Prize, SIAM Conf. Uncertainty Quantification, Trieste, Italy, March 1st 2024.

Perspectives on stochastic gradient descent

Stochastic gradient descent (SGD) is a randomised algorithm for the optimisation of large sums of strongly convex functions.

Perspective 1: In modern machine learning, stochastic gradient descent is often used with a so-called constant learning rate, then:

- the algorithm doesn't converge to a minimiser, but
- acts as an implicit regulariser

Study the regularisation properties of stochastic gradient descent.

Perspective 2: Stochastic gradient descent is an iterative algorithm of the form

$$\theta_k \leftarrow F(\theta_{k-1}) \qquad (k \in \mathbb{N}),$$

i.e., the algorithm generates a discrete-time dynamical system $(\theta_k)_{k=0}^{\infty}$.

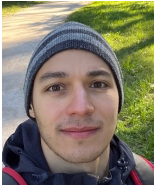
Propose a continuous-time variant of stochastic gradient descent $(\theta(t))_{t\geq 0}$ to analyse the constant learning rate setting.

A continuous-time variant of stochastic gradient descent

Initial development:

L. 2021: **Analysis of stochastic gradient descent in continuous time**, Stat. Comput. 31, 39.

Significant developments since then:



Matei Hanu

Kexin Jin

Chenguang Liu

Alessandro Scagliotti

Claudia Schillings

Carola-Bibiane Schönlieb

Hanu, L., Schillings 2023: **Subsampling in ensemble Kalman inversion**, Inv. Probl. 39, 094002.
Jin, L., Liu, Scagliotti 2022: **Losing momentum in continuous-time stochastic optimisation**, preprint.
Jin, L., Liu, Schönlieb 2023: **A Continuous-time Stochastic gradient descent Method for Continuous Data**, JMLR 24(274):1–48.
Jin, Liu, L. 2024: **Subsampling error in Stochastic Gradient Langevin Diffusions**, AISTATS.
L. 2022: **Gradient flows and randomised thresholding: sparse inversion and classification**, Inv. Probl. 38, 124006.

Funding: Engineering and Physical Sciences Research Council, Swindon, UK

Credit for photo: Nick Saffell (Carola-Bibiane Schönlieb)

Outline

Outline

Problem setting

Consider a minimisation problem of the form

$$\min_{\theta \in X} \bar{\Phi}(\theta) := \frac{1}{N} \sum_{i=1}^{N} \Phi_i(\theta),$$

where $X := \mathbb{R}^n$ and the Φ_i are sufficiently smooth $(i = 1, \ldots, N)$.

Optimisation in data science

- $\bar{\Phi}$ is some kind of potential ($\in$ {negative log-likelihood, loss function, misfit}) given with respect to a large data set
- the Φ_i then represent the potential of a (small) data subsample

Optimisation in data science, e.g.,

Image reconstruction

Supervised learning

Optimisation in data science, e.g.,

Image reconstruction

Figure: *Image of Galaxy M100 from Hubble before (left) and after fixing its mirror (right).* [NASA, Hubble's Mirror Flaw]

Use a variational approach to deblur an image by solving

$$\min_{\theta \in X} \frac{1}{N} \sum_{i=1}^{N} \underbrace{(C_i\theta - y_i)^2 + \mathrm{Reg}(\theta)}_{=\Phi_i(\theta)},$$

where $(C_i)_{i=1}^N$ are the rows of a kernel matrix, θ is the reconstructed image, y is the blurry image, and $\mathrm{Reg} : X \to \mathbb{R}$ is an appropriate regulariser.

Optimisation in data science, e.g.,

Supervised learning

Given a pair of random variables $(x, y) \sim \pi_{x,y}$.

- Learn how to predict y given x, i.e. find f:

$$f(x) \approx y$$

Supervised learning: approximate f using a parametric model $\widehat{f}(\cdot;\theta)$ and sampled data $(x_1, y_1), \ldots, (x_n, y_n) \sim \pi_{x,y}$ by minimising

$$\min_{\theta \in X} \frac{1}{N} \sum_{i=1}^{N} \underbrace{\|\widehat{f}(x_i;\theta) - y_i\|^2}_{=\Phi_i(\theta)}.$$

Image classification. [Krizhevsky 2009]

Training data, e.g., from the CIFAR10 dataset:

$(x_1, y_1) = ($ , ‘cat’),

$(x_2, y_2) = ($ , ‘frog’),

$(x_3, y_3) = ($ , ‘airplane’),...

Optimisation in data science, e.g.,

Supervised learning

Given a pair of random variables $(x, y) \sim \pi_{x,y}$.

- Learn how to predict y given x, i.e. find f:

$$f(x) \approx y$$

Supervised learning: approximate f using a parametric model $\widehat{f}(\cdot;\theta)$ and sampled data $(x_1, y_1), \ldots, (x_n, y_n) \sim \pi_{x,y}$ by minimising

$$\min_{\theta \in X} \frac{1}{N} \sum_{i=1}^{N} \underbrace{\|\widehat{f}(x_i;\theta) - y_i\|^2}_{=\Phi_i(\theta)}.$$

Image classification. [Dechter; 1986]

Deep neural networks have been particularly successful at imaging tasks. Here,

$$\begin{aligned} \widehat{f}(x;\theta) &= f^{(K)} \\ f^{(k)} &= \sigma(W^{(k)} f^{(k-1)} + b^{(k)}) \qquad (k = 1, ..., K) \\ f^{(0)} &= x \end{aligned}$$

with $\theta = (W^{(1)}, b^{(1)}, \ldots, W^{(K)}, b^{(K)})$ and σ being an activation function.

Gradient descent and stochastic gradient descent

How do we solve the following optimisation problem?

$$\min_{\theta \in X} \bar{\Phi}(\theta) := \frac{1}{N} \sum_{i=1}^{N} \Phi_i(\theta)$$

Gradient descent (GD)

[Cauchy; 1847]

for $k = 1, 2, \ldots$:

$$\theta_k \leftarrow \theta_{k-1} - \eta_k \nabla \bar{\Phi}(\theta_{k-1}), \qquad \nabla \bar{\Phi}(\theta_{k-1}) := \underbrace{\tfrac{1}{N} \textstyle\sum_{i=1}^{N} \nabla \Phi_i(\theta_{k-1})}_{(N \text{ gradient evaluations})}$$

- converges if $\bar{\Phi}$ has a minimiser and is convex and if the "step size" η_k is sufficiently small

Gradient descent and stochastic gradient descent

How do we solve the following optimisation problem?

$$\min_{\theta \in X} \bar{\Phi}(\theta) := \frac{1}{N} \sum_{i=1}^{N} \Phi_i(\theta)$$

Stochastic gradient descent (SGD) [Robbins + Monro; 1951]

for $k = 1, 2, \ldots$:

$$\theta_k \leftarrow \theta_{k-1} - \eta_k \nabla \Phi_{\boldsymbol{i}_k}(\theta_{k-1}), \qquad \underbrace{\boldsymbol{i}_k \sim \mathrm{Unif}(I)}_{(= \text{“subsampling”})} .$$

- converges if $\Phi_1, \ldots, \Phi_N$ are strongly convex and “learning rate” $\eta_k \downarrow 0$ $(k \to \infty)$ slowly

Outline

Stochastic gradient descent in machine learning

Consider again the supervised learning problem

$$\min_{\theta \in X} \frac{1}{N} \sum_{i=1}^{N} \|\widehat{f}(x_i; \theta) - y_i\|^2.$$

Problems in supervised learning.

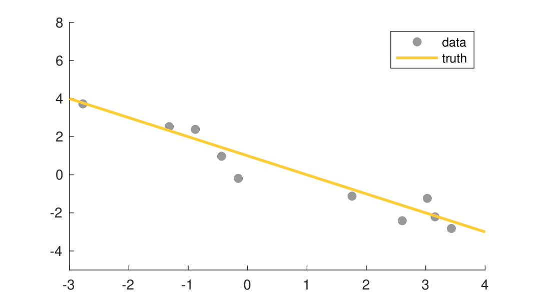
- very large data sets ($N \gg 1$) $\rightarrow$ SGD can help with that.
- depending on the choice of $\widehat{f}$, the target function may be non-convex
  - target functions in deep learning are usually non-convex $\rightarrow$ SGD might struggle. [Du+al; 2017]
- solving this problem may overfit the data
  - machine learning models tend to be highly flexible and overparameterised
  - models may fit the noisy training data and generalise badly to unseen data $\rightarrow$ Let's see!

Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}}$, $\varepsilon^{(1)}_{\text{noise}}, \ldots \sim \mathrm{N}(0,1)$ iid.



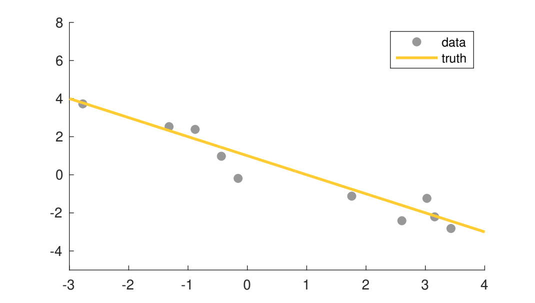
Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis.



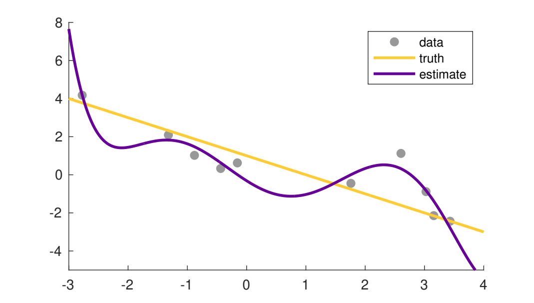
Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0, 1)$ iid.

Construct model $\widehat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis.



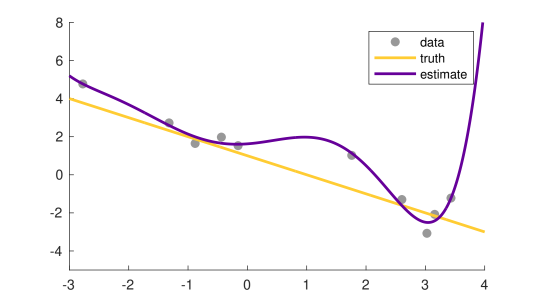
Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0, 1)$ iid.

Construct model $\widehat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis.



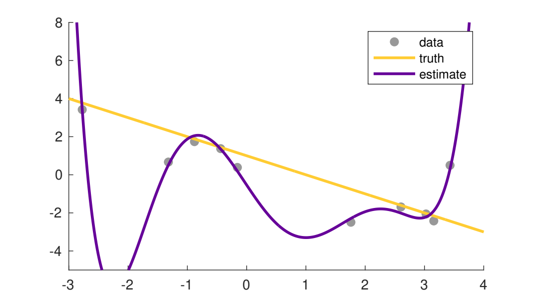
Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let $(x_1, y_1), \dots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis.



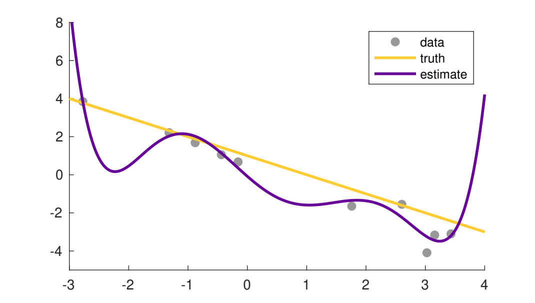
Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let $(x_1, y_1), \dots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \dots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis.



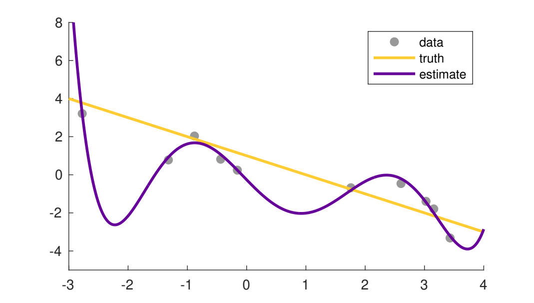
Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0, 1)$ iid.

Construct model $\widehat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis.



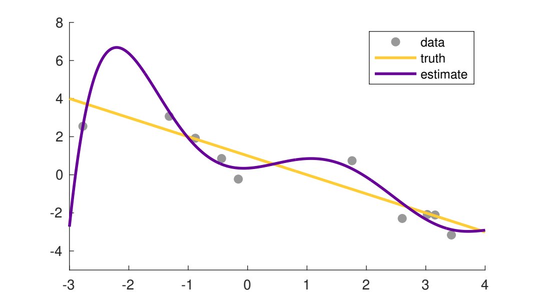
Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let $(x_1, y_1), \dots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \dots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis.



Overfitting

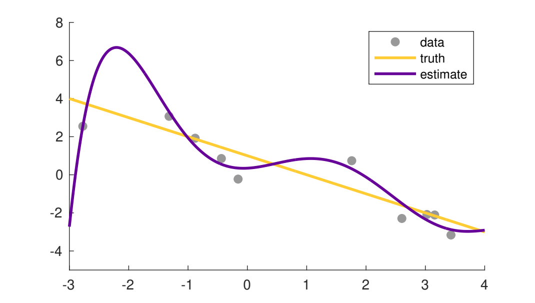
- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis.

Truth is badly estimated + estimated model is unstable with respect to resampling the noise.
$\rightarrow$ overfitting



Regularisation

- The problem of overfitting in learning is related to that of ill-posedness in inverse problems
  - ill-posedness describes the non-existence, non-uniqueness or instability of estimates with respect to changes in the observational data
  - Instability:

Inversion: instability usually measured with {Lipschitz, Hölder, -} continuity with respect to data
Learning: study bias-variance tradeoff: instability $\Rightarrow$ large variance in training with respect to different data sets

- Ill-posedness in inverse problems can often be cured with regularisation
  - Variational regularisation: enforce additional information by modifying the target function

$$\min_{\theta} \bar{\Phi}(\theta) + \mathrm{Reg}(\theta)$$

  - Bayesian approach: being aware of uncertainties usually leads to stability/well-posedness

- Overfitting can sometimes be addressed by regularisation [Mohri+al.; 2018]

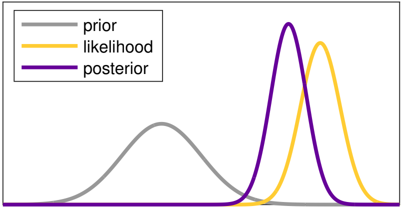
The Bayesian approach

Optimisation may not actually be the best way to learn a data set

- Fitting the model to the noisy data overfits the model
- Uncertainties remain in the model and are not quantified

Bayesian approach

- Consider the parameter θ to be uncertain and model knowledge/assumptions/... re θ with a so-called **prior** $\pi_{\mathrm{prior}} = \mathbb{P}(\theta \in \cdot)$
- Use model and data to learn about θ by conditioning – obtain the **posterior**



$$\pi_{\mathrm{post}} = \mathbb{P}(\theta \in \cdot | y_i = \widehat{f}(x_i;\theta) + \varepsilon_{\mathrm{noise}}^{(i)}, i = 1, \ldots, m)$$

- Predictions of the trained model will be random/uncertain $\pi_{\mathrm{post}}(\widehat{f}(x;\theta) \in \cdot)$
- The posterior is usually stable with respect to perturbations in the data $\rightarrow$ well-posed

[Dashti+Stuart 2017] [Hosseini; 2017] **[L.; 2020, 2023]** [Sprungk; 2020] [Stuart; 2010] [Sullivan; 2017] ,...

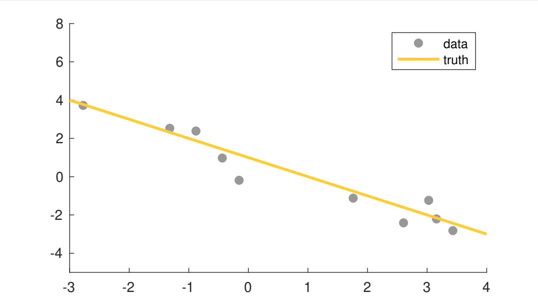
The Bayesian approach

- Predictions of the trained model will be random/uncertain $\pi_{\mathrm{post}}(f(x;\theta) \in \cdot)$

Bayesian polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}, \varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0, 1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and additionally enforce sparsity with prior $\pi_{\mathrm{prior}} = \mathrm{N}(0, \mathrm{diag}(2^{-1}, \ldots, 2^{-7}))$.



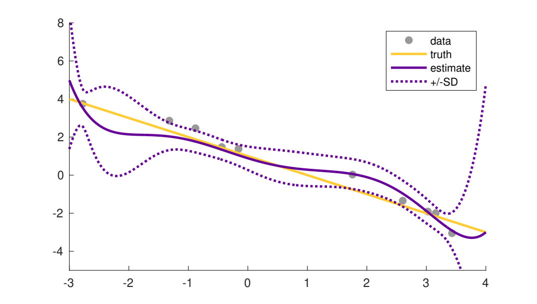
The Bayesian approach

- Predictions of the trained model will be random/uncertain $\pi_{\mathrm{post}}(f(x;\theta) \in \cdot)$

Bayesian polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and additionally enforce sparsity with prior $\pi_{\mathrm{prior}} = \mathrm{N}(0, \mathrm{diag}(2^{-1}, \ldots, 2^{-7}))$.



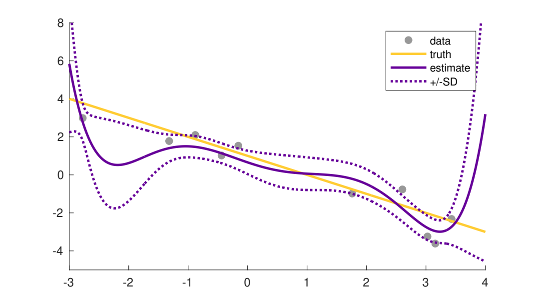
The Bayesian approach

- Predictions of the trained model will be random/uncertain $\pi_{\text{post}}(f(x;\theta) \in \cdot)$

Bayesian polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}}$, $\varepsilon^{(1)}_{\text{noise}}, \ldots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and additionally enforce sparsity with prior $\pi_{\text{prior}} = \mathrm{N}(0, \mathrm{diag}(2^{-1}, \ldots, 2^{-7}))$.



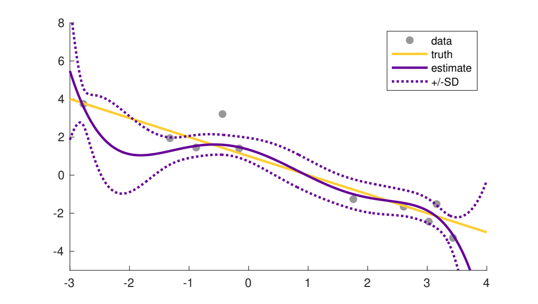
The Bayesian approach

- Predictions of the trained model will be random/uncertain $\pi_{\text{post}}(f(x;\theta) \in \cdot)$

Bayesian polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}}$, $\varepsilon^{(1)}_{\text{noise}}, \ldots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and additionally enforce sparsity with prior $\pi_{\text{prior}} = \mathrm{N}(0, \mathrm{diag}(2^{-1}, \ldots, 2^{-7}))$.



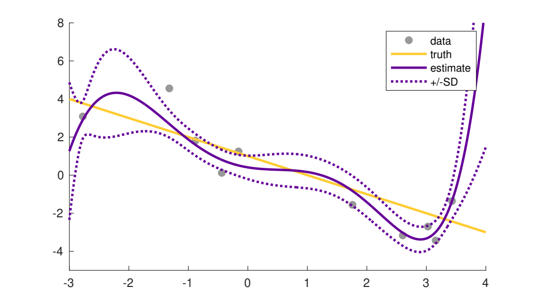
The Bayesian approach

- Predictions of the trained model will be random/uncertain $\pi_{\text{post}}(f(x;\theta) \in \cdot)$

Bayesian polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)}$, $\varepsilon_{\text{noise}}^{(1)}, \ldots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and additionally enforce sparsity with prior $\pi_{\text{prior}} = \mathrm{N}(0, \mathrm{diag}(2^{-1}, \ldots, 2^{-7}))$.



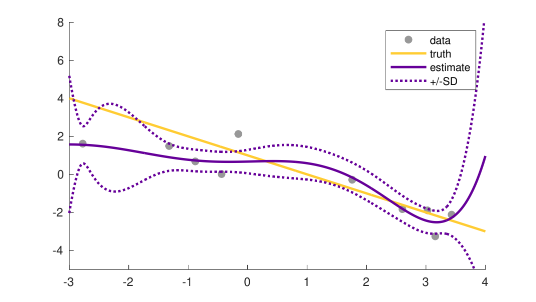
The Bayesian approach

- Predictions of the trained model will be random/uncertain $\pi_{\mathrm{post}}(f(x;\theta) \in \cdot)$

Bayesian polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and additionally enforce sparsity with prior $\pi_{\mathrm{prior}} = \mathrm{N}(0, \mathrm{diag}(2^{-1}, \ldots, 2^{-7}))$.



The Bayesian approach

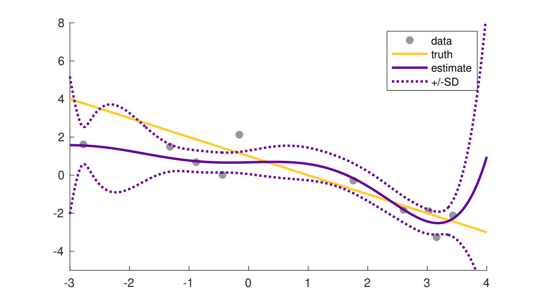
- Predictions of the trained model will be random/uncertain $\pi_{\mathrm{post}}(f(x;\theta) \in \cdot)$

Bayesian polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon_{\mathrm{noise}}^{(i)}$, $\varepsilon_{\mathrm{noise}}^{(1)}, \ldots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and additionally enforce sparsity with prior $\pi_{\mathrm{prior}} = \mathrm{N}(0, \mathrm{diag}(2^{-1}, \ldots, 2^{-7}))$.

Stable solution!
$\rightarrow$ no overfitting



Stochastic gradient descent and implicit regularisation

- Regularisation of supervised learning problems is difficult
  - Correct choice of regularisers or priors is unclear in general (polynomial regression is easy)
  - Bayesian learning is computationally expensive
- **Idea:** Use stochastic gradient descent with a constant learning rate.
  - Markov chain Monte Carlo sampling from the posterior; often through a 'noisy gradient descent'

$$\theta_k \leftarrow \theta_{k-1} - \eta_k \nabla \bar{\Phi}(\theta_{k-1}) + \eta_k \nabla \log \pi_{\mathrm{prior}}(\theta_{k-1}) + \sqrt{\eta_k}\xi_k, \qquad \xi_1, \xi_2, \ldots \sim \mathrm{N}(0, \mathrm{Id}) \text{ iid.} \quad \text{(ULA)}$$

    SGD is also a 'noisy gradient descent', maybe it can act as an approximate MCMC sampler?
  - high variability in the model with respect to training data often suggests overfitting; SGD leads to robustness with respect to small data sets at a time.

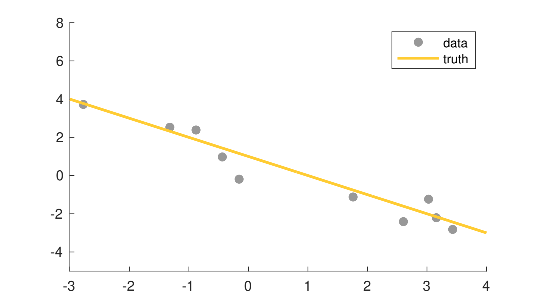
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say η_k constant)?

SGD-regularised Polynomial regression

Let $(x_1, y_1), \dots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \dots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$.



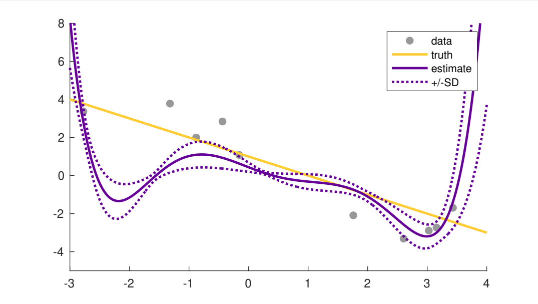
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say η_k constant)?

SGD-regularised Polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0, 1)$ iid.

Construct model $\widehat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$.



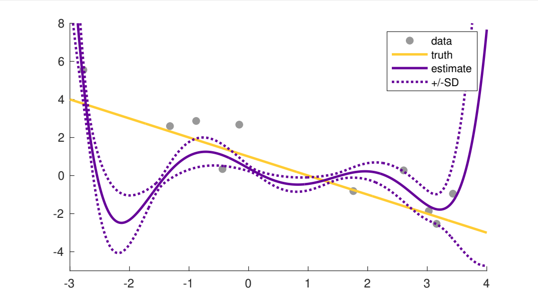
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say η_k constant)?

SGD-regularised Polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0, 1)$ iid.

Construct model $\widehat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$.



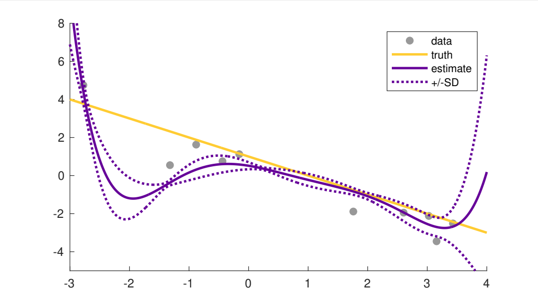
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say η_k constant)?

SGD-regularised Polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}}$, $\varepsilon^{(1)}_{\text{noise}}, \ldots \sim \mathrm{N}(0, 1)$ iid.

Construct model $\widehat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$.



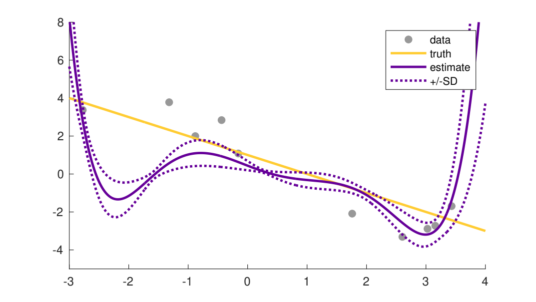
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say η_k constant)?

SGD-regularised Polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$.



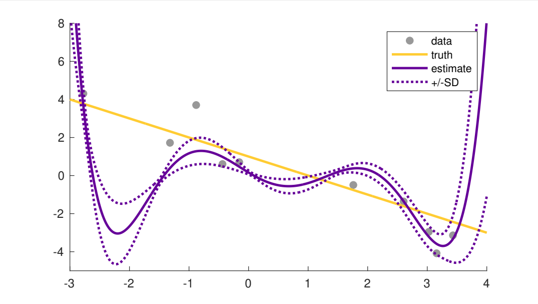
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say η_k constant)?

SGD-regularised Polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}, \varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$.



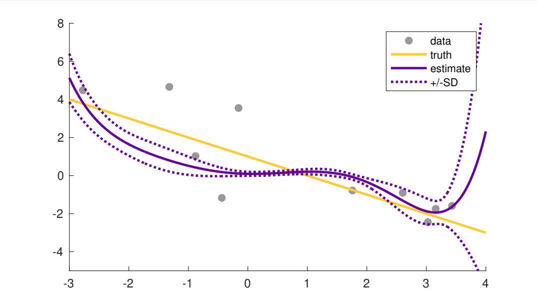
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say η_k constant)?

SGD-regularised Polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}}$, $\varepsilon^{(1)}_{\text{noise}}, \ldots \sim \mathrm{N}(0, 1)$ iid.

Construct model $\widehat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$.



Stochastic gradient descent and implicit regularisation

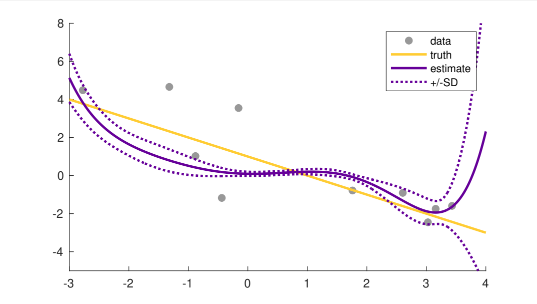
- Can we just apply SGD in a non-convergent regime (say η_k constant)?

SGD-regularised Polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\mathrm{noise}}$, $\varepsilon^{(1)}_{\mathrm{noise}}, \ldots \sim \mathrm{N}(0,1)$ iid.

Construct model $\widehat{f}(x;\theta) := \sum_{i=0}^{6} \theta_i \mathrm{H}_i(x)$ on a Hermite basis and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$.

Not perfect, but not terrible and easy to obtain!



Stochastic gradient descent and implicit regularisation

- Regularisation of supervised learning problems is difficult
  - Correct choice of regularisers or priors is unclear in general (polynomial regression is easy)
  - Bayesian learning is computationally expensive

- **Idea:** Use stochastic gradient descent with a constant learning rate.
  - Markov chain Monte Carlo sampling from the posterior; often through a 'noisy gradient descent'

    $$\theta_k \leftarrow \theta_{k-1} - \eta_k \nabla \bar{\Phi}(\theta_{k-1}) + \eta_k \nabla \log \pi_{\mathrm{prior}}(\theta_{k-1}) + \sqrt{\eta_k}\xi_k, \qquad \xi_1, \xi_2, \ldots \sim \mathrm{N}(0, \mathrm{Id}) \text{ iid.} \quad \text{(ULA)}$$

    SGD is also a 'noisy gradient descent', maybe it can act as an approximate MCMC sampler?
  - high variability in the model with respect to training data often suggests overfitting; SGD leads to robustness with respect to small data sets at a time.
- Indeed, this is a rather popular way of regularisation in machine learning; part of implicit regularisation.

- **Questions**
  - What actually happens when we apply SGD with constant learning rate? [Dieuleveut+al.; 2020]
  - Is there a stationary regime? What do we know about it? Is it a posterior? [Mandt+al.; 2017]

Outline

Algorithms in continuous time

- Iterative algorithms can be understood as discrete-in-time dynamical systems

$$\xi_k \leftarrow F(\xi_{k-1}) \quad (k \in \mathbb{N}), \qquad \xi_0 \in X$$

- We can sometimes find continuous-in-time dynamical systems, e.g.,

$$\dot{\xi}(t) = G(\xi(t)) \quad (t \geq 0), \qquad \xi(0) = \xi_0$$

that behave analogous to the iterative algorithms

Example. (Gradient descent and gradient flows)

Gradient descent

$$\zeta_k \leftarrow \zeta_{k-1} - \eta_k \nabla \bar{\Phi}(\zeta_{k-1}) \qquad (k \in \mathbb{N})$$

is a forward Euler discretisation of the ordinary differential equation

$$\dot{\zeta}(t) = -\nabla \Phi(\zeta(t)) \qquad (t \geq 0)$$

which is a so-called gradient flow.

Algorithms in continuous time

It is sometimes easier or more appropriate to analyse algorithms in continuous time

- certain numerical artefacts that appear in the discrete setting are not particularly interesting: stiffness, step size restrictions,... [Iserles; 2012]
- certain effects can are more or only visible in a continuous setting: ill-posedness of deconvolution,... [Bredies+Lorenz; 2018]
- continuous time adds additional regularity

Continuous time allows us to compare algorithms to physical/biological processes

- Gradient flows appear everywhere, e.g., the heat equation $\dot{u} = \triangle u$ [Santambrogio; 2017]
- Certain classification methods behave like partial differential equations that describe phase separation [Budd+van Gennip; 2020] **[Budd+van Gennip+L.; 2021]**

More examples: Ensemble Kalman inversion [Schillings+Stuart; 2017] [Blömker+al.; 2019], data assimilation [Law+al. 2015] [de Wiljes+al. 2018], continuum limits of graphs [Trillos+Sanz-Alonso; 2018], MCMC [Ottobre+al.; 2019], image reconstruction [Rudin+al.; 1992] [Schönlieb; 2015], data science [Kreusser+Wolfram; 2020]

Diffusion limit of SGD

Predominant model for SGD in continuous time: Diffusion process

- Idea: $\eta_k = \eta \approx 0 \Rightarrow$ gradient error is approximately Gaussian (CLT)
- Hence, $(\theta_k)_{k=1}^{\infty}$ can be represented by a diffusion process

$$\dot{\theta}_{\mathrm{sde}}(t) = -\nabla\bar{\Phi}(\theta_{\mathrm{sde}}(t)) + \sqrt{\eta}\Sigma(\theta_{\mathrm{sde}}(t))^{1/2}\dot{\mathrm{W}}_t \quad (t \geq 0), \qquad \theta(0) = \theta_0.$$

[Hu+al.; 2019] [Li+al.; 2016, 2017, 2019] [Mandt+al.; 2015, 2016, 2017] [Wojtowytsch; 2024] ,...

- for large η_k, the paths of $(\theta_k)_{k=1}^{\infty}$ are very different from a diffusion
  - preasymptotic phase and constant η_k not explained
- diffusion does not actually explain subsampling in a continuous-time model
  - does not represent the discrete nature of the potential selection
  - needs access to $\bar{\Phi}$

Outline

Observations and fundamental idea

- the update

$$\theta_k \leftarrow \theta_{k-1} - \eta \nabla \Phi_{\boldsymbol{i}_k}(\theta_{k-1}) \qquad \text{(discrete)}$$

  is a forward Euler discretisation of the gradient flow

$$\dot{\theta}(t) = -\nabla \Phi_{\boldsymbol{i}_k}(\theta(t)) \qquad \text{(continuous)}$$

- learning rate η has two different meanings
  (i) η is the step size of the gradient flow discretisation
  (ii) η determines the length of the time interval with which we switch the Φ_i

Idea.

Obtain a continuous time model for SGD, by

(i) let the step size go to 0, i.e. replace (discrete) by (continuous).

(ii) switch the potentials in the gradient flow at a rate of $1/\eta$

Switching of the potentials

control the switching of the potentials by a continuous-time Markov process (CTMP) $(\boldsymbol{i}(t))_{t\geq 0}$ on $I := \{1, ..., N\}$ ("index process")

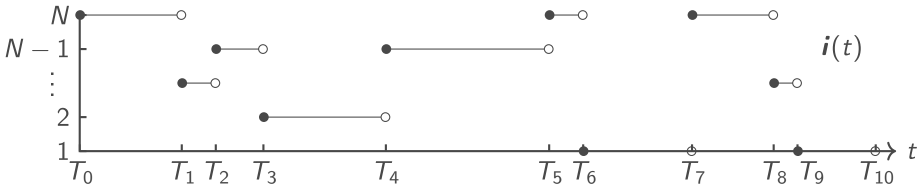


Figure: *Cartoon of a CTMP*

CTMPs 101

- $(\boldsymbol{i}(t))_{t\geq 0}$ is piecewise constant
- randomly jumps from one state to another after a random waiting time $\Delta \sim \pi_{\mathrm{wt}}(\cdot|t_0)$

Stochastic gradient process

CTMP $(\boldsymbol{i}(t))_{t\geq 0}$ representing a constant learning rate $\eta_\bullet \equiv \eta > 0$

- constant learning rates are popular in practice
- $\pi_{\mathrm{wt}}(\cdot|t_0)$ is constant in time (indeed this will be an exponential distribution)

$(\boldsymbol{i}(t))_{t\geq 0}$ has constant transition rate matrix $A \in \mathbb{R}^{N\times N}$: $A_{i,j} := \begin{cases} \frac{1}{(N-1)\eta}, & \text{if } i \neq j, \\ -\frac{1}{\eta}, & \text{if } i = j. \end{cases}$

Definition. [L.; 2021]

We define the **stochastic gradient process with constant learning rate (SGPC)** by $(\theta(t))_{t\geq 0}$, which satisfies

$$\dot\theta(t) = -\nabla\Phi_{\boldsymbol{i}(t)}(\theta(t)) \quad (t \geq 0), \qquad \theta(0) = \theta_0.$$

$(\theta(t))_{t\geq 0}$ and $(\xi(t))_{t\geq 0}$ are almost surely well-defined, if

Assumption [Lipschitz]. *For $i \in I$: $\Phi_i \in C^1(X, \mathbb{R})$ and $\nabla\Phi_i$ is Lipschitz continuous.*

Stochastic gradient process

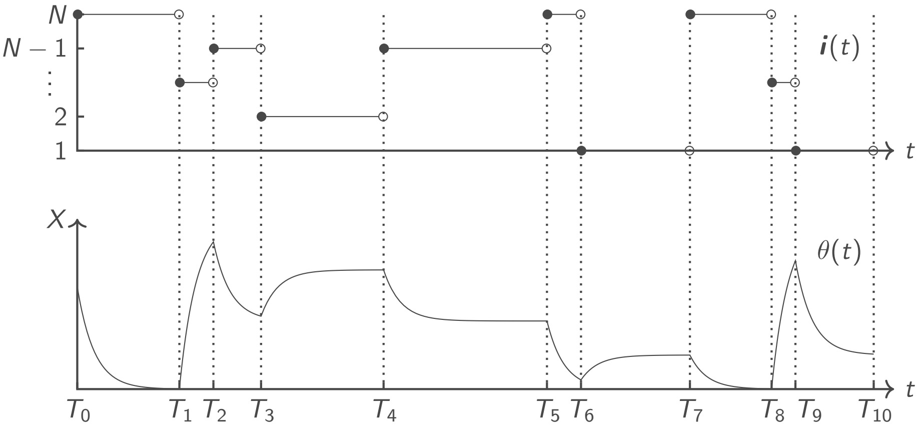


Figure: *Cartoon of SGPC*

Properties

- $(\boldsymbol{i}(t), \theta(t))_{t\geq 0}$ is a piecewise-deterministic Markov process: essentially an ODE with a right-hand side that changes at random points in time
- the choice of the transition rate matrix of $(\boldsymbol{i}(t), \theta(t))_{t\geq 0}$ leads to subsampling at rate $1/\eta$
  - mean waiting time $\mathbb{E}[T_i - T_{i-1}] = \eta$
- $(\theta(t))_{t\geq 0}$ approximates the gradient flow $(\zeta(t))_{t\geq 0}$ for small η **[L.; 2021]**

Short learning rate ($\eta \downarrow 0$)

Example. Let $\Phi_1(\theta) := (\theta - 1)^2/2$ and $\Phi_2(\theta) := (\theta + 1)^2/2$. $\Rightarrow \bar{\Phi}(\theta) = (\theta^2 + 1)/2$.

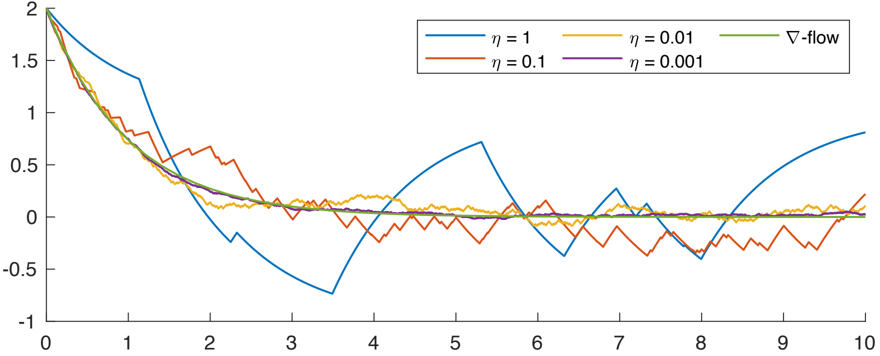


Figure: *Exemplary realisations of SGPC and plot of precise gradient flow. Discretisation with* `ode45`*.*

Convergence proof in **[L.; 2021]** using techniques from [Kushner; 1984].

Properties

- $(\boldsymbol{i}(t), \theta(t))_{t\geq 0}$ is a piecewise-deterministic Markov process: essentially an ODE with a right-hand side that changes at random points in time
- the choice of the transition rate matrix of $(\boldsymbol{i}(t), \theta(t))_{t\geq 0}$ leads to subsampling at rate $1/\eta$
  - mean waiting time $\mathbb{E}[T_i - T_{i-1}] = \eta$
- $(\theta(t))_{t\geq 0}$ approximates the gradient flow $(\zeta(t))_{t\geq 0}$ for small η **[L.; 2021]**
- stochastic gradient flow has a biological interpretation [Kussell+Leibler; 2005]
  - clonal populations that live in randomly changing environments use diversified bet-hedging strategies that follow similar dynamics

Outline

Longtime behaviour ($t \to \infty$)

What happens with $\mathbb{P}(\theta(t) \in \cdot)$ as $t \to \infty$?

- Stability? Stationary measures?
- Speed of convergence?
- Characterisation of stationary measures?
- Implicit regularisation?
- Posteriors?

Preliminaries

Wasserstein distance

Let $q \in (0,1]$. Consider Wasserstein distance between $\pi, \pi' \in \mathrm{Prob}(X)$:

$$\mathrm{W}(\pi, \pi') := \inf_{H \in \mathrm{Coup}(\pi,\pi')} \int_{X \times X} \min\{1, \|\theta - \theta'\|_2^q\} H(\mathrm{d}\theta, \mathrm{d}\theta'),$$

$$\mathrm{Coup}(\pi, \pi') := \{G \in \mathrm{Prob}(X^2) : \quad G(\cdot \times X) = \pi, \quad G(X \times \cdot) = \pi'\}.$$

Assumption [Smooth]. *For any $i \in I$, let $\Phi_i \in C^2(X; \mathbb{R})$ and let $\nabla\Phi_i, \mathrm{H}\Phi_i$ be continuous and bounded on bounded subsets of X.*

Assumption [Convex]. *There are some $\kappa_i \in \mathbb{R}$, with*

$$\langle \theta_0 - \theta_0', \nabla\Phi_i(\theta_0) - \nabla\Phi_i(\theta_0')\rangle \geq \kappa_i \|\theta_0 - \theta_0'\|^2 \qquad (\theta_0, \theta_0' \in X, i \in I),$$

with $\kappa_1 + \cdots + \kappa_N > 0$.

Longtime behaviour ($t \to \infty$)

Theorem. [L.; 2021]

Let Assumptions `[Smooth]` and `[Convex]` hold. Then, $(\theta(t), \boldsymbol{i}(t))_{t>0}$ has a unique stationary measure π_{C} on $(X \times I, \mathcal{B}X \otimes 2^I)$. Moreover, there exist $\kappa', c > 0$ and $q \in (0,1]$, with

$$\mathrm{W}(\pi_{\mathrm{C}}(\cdot \times I), \mathbb{P}(\theta(t) \in \cdot | \theta_0, i_0)) \leq c \exp(-\kappa' t) \left(1 + \sum_{i \in I} \int_X \|\theta_0 - \theta'\|^q \pi_{\mathrm{C}}(\mathrm{d}\theta' \times \{i\})\right)$$

$$(i_0 \in I, \theta_0 \in X).$$

Longtime behaviour ($t \to \infty$)

Theorem. [L.; 2021]

Let Assumptions [Smooth] and [Convex] hold. Then, $(\theta(t), \boldsymbol{i}(t))_{t>0}$ has a unique stationary measure π_{C} on $(X \times I, \mathcal{B}X \otimes 2^I)$. Moreover, there exist $\kappa', c > 0$ and $q \in (0,1]$, with

$$\mathrm{W}(\pi_{\mathrm{C}}(\cdot \times I), \mathbb{P}(\theta(t) \in \cdot | \theta_0, i_0)) \leq c \exp(-\kappa' t) \left(1 + \sum_{i \in I} \int_X \|\theta_0 - \theta'\|^q \pi_{\mathrm{C}}(\mathrm{d}\theta' \times \{i\})\right) \quad (i_0 \in I, \theta_0 \in X).$$

- convergence with exponential speed
- proof based on results by [Benaïm+al.; 2012] [Cloez+Hairer; 2015]
- convexity assumption can be weakened (needs Hörmander Bracket condition)
- SGD with constant stepsize is safe to use in 'more-convex-than-not' settings and converges very quickly to its stationary regime

Longtime behaviour ($t \to \infty$)

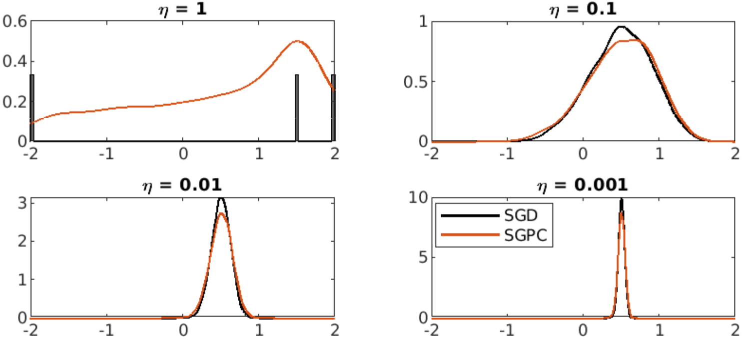


Figure: *Kernel density estimates of* $\mathbb{P}(\theta(10) \in \cdot | \theta(0) = -1.5) \approx \pi_{\mathrm{C}}$ *(SGPC) and* $\mathbb{P}(\theta_{10/\eta} \in \cdot | \theta_0 = -1.5)$ *(SGD) based on* $\eta \in \{1, 0.1, 0.01, 0.001\}$ *using 10,000 samples each.* [**Example.** *Let* $N := 3$, *i.e.* $I := \{1, 2, 3\}$, *and* $X := \mathbb{R}$. *We define the potentials* $\Phi_1(\theta) := \frac{1}{2}(\theta + 2)^2$, $\Phi_2(\theta) := \frac{1}{2}(\theta - 1.5)^2$, $\Phi_3(\theta) := \frac{1}{2}(\theta - 2)^2$ $(\theta \in X)$. *Here,* $\mathrm{argmin} \bar{\Phi} = \{0.5\}$.]

Stationary distributions and implicit regularisation

- π_C might be a good representation for the implicit regularisation achieved by SGD
- It appears as if $\pi_C \to \delta(\cdot - \theta^*)$, as $\eta \downarrow 0$, where $\theta_* \in \operatorname{argmin} \bar{\Phi}$. Indeed, we can show:

Corollary.

Let Assumptions `[Smooth]` and `[Convex2]` hold. Then, $\lim_{\eta \downarrow 0} \mathrm{W}(\pi_\mathrm{C}(\cdot \times I), \delta(\cdot - \theta_*)) = 0$.

Assumption `[Convex2]`. *There is a* $\kappa > 0$, *with*

$$\langle \theta_0 - \theta_0', \nabla\Phi_i(\theta_0) - \nabla\Phi_i(\theta_0') \rangle \geq \kappa \|\theta_0 - \theta_0'\|^2 \qquad (\theta_0, \theta_0' \in X, i \in I).$$

- the corollary above is a simple application of Proposition 4(ii) in **[L.; 2021]**
- the result shows that η controls the strength of the regularisation
  - decreasing η over time corresponds to the classical SGD setting
  - we can also do that in the stochastic gradient process

Implicit regularisation and posteriors

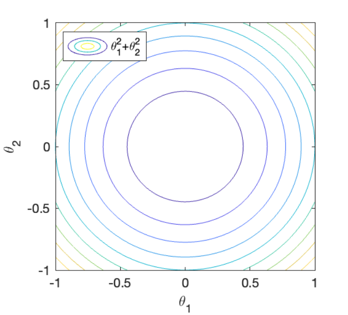
- π_C behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

[Benaim+al.; 2015]

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \underbrace{\|\theta - \mathbf{1}\|^2}_{:=\Phi_1} + \frac{1}{2} \underbrace{\|\theta + \mathbf{1}\|^2}_{:=\Phi_1}$$



Implicit regularisation and posteriors

- π_C behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

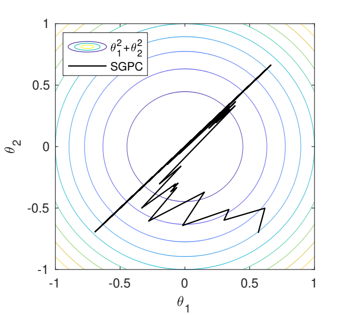
[Benaim+al.; 2015]

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \underbrace{\|\theta - \mathbf{1}\|^2}_{:=\Phi_1} + \frac{1}{2} \underbrace{\|\theta + \mathbf{1}\|^2}_{:=\Phi_1}$$

Employ SGPC with $\eta = 10$.



Implicit regularisation and posteriors

- π_C behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

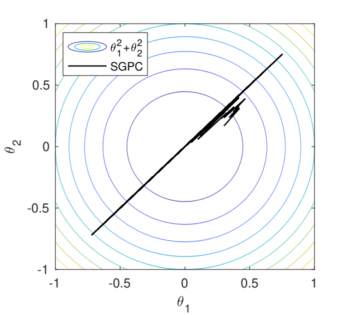
[Benaim+al.; 2015]

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \underbrace{\|\theta - \mathbf{1}\|^2}_{:=\Phi_1} + \frac{1}{2} \underbrace{\|\theta + \mathbf{1}\|^2}_{:=\Phi_1}$$

Employ SGPC with $\eta = 10$.



Implicit regularisation and posteriors

- π_C behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

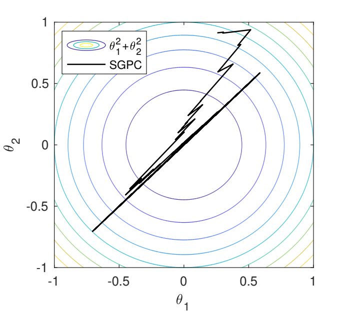
[Benaim+al.; 2015]

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \underbrace{\|\theta - \mathbf{1}\|^2}_{:=\Phi_1} + \frac{1}{2} \underbrace{\|\theta + \mathbf{1}\|^2}_{:=\Phi_1}$$

Employ SGPC with $\eta = 10$.



Implicit regularisation and posteriors

- π_C behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

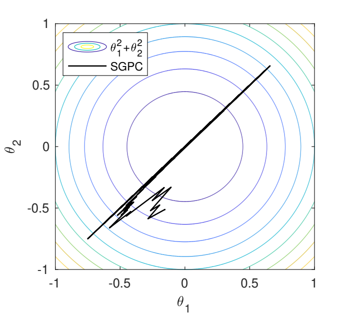
[Benaim+al.; 2015]

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \underbrace{\|\theta - \mathbf{1}\|^2}_{:=\Phi_1} + \frac{1}{2} \underbrace{\|\theta + \mathbf{1}\|^2}_{:=\Phi_1}$$

Employ SGPC with $\eta = 10$.



Implicit regularisation and posteriors

- π_C behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

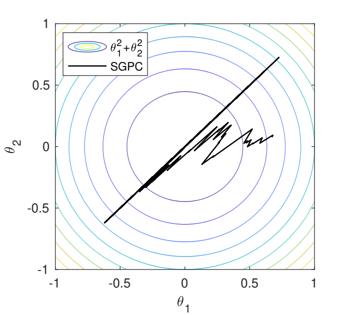
[Benaim+al.; 2015]

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \underbrace{\|\theta - \mathbf{1}\|^2}_{:=\Phi_1} + \frac{1}{2} \underbrace{\|\theta + \mathbf{1}\|^2}_{:=\Phi_1}$$

Employ SGPC with $\eta = 10$.



Implicit regularisation and posteriors

- π_C behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

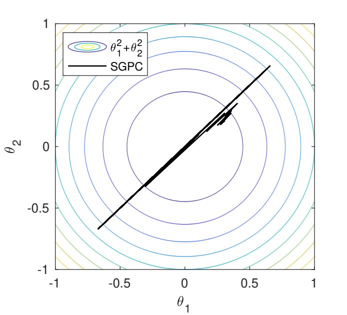
[Benaim+al.; 2015]

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \underbrace{\|\theta - \mathbf{1}\|^2}_{:=\Phi_1} + \frac{1}{2} \underbrace{\|\theta + \mathbf{1}\|^2}_{:=\Phi_1}$$

Employ SGPC with $\eta = 10$.



Implicit regularisation and posteriors

- π_C behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

[Benaim+al.; 2015]

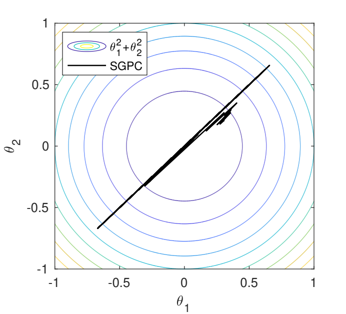
SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \underbrace{\|\theta - \mathbf{1}\|^2}_{:=\Phi_1} + \frac{1}{2} \underbrace{\|\theta + \mathbf{1}\|^2}_{:=\Phi_1}$$

Employ SGPC with $\eta = 10$.

π_C is concentrated on a subspace and rather different from the measure we would associate with this optimisation problem.



Implicit regularisation and posteriors

- π_C behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace [Benaim+al.; 2015]
- Can we turn this into a posterior?
  - In principle, yes. The Stochastic Gradient Langevin Dynamics combines SGD and ULA by adding white noise to SGD. It approximates $\pi \propto \exp(-\bar{\Phi})$. [Welling+Teh; 2011]
  - In continuous time, we obtain the Stochastic Gradient Langevin Diffusion **[Jin, Liu, L.; 2024]**

$$\mathrm{d}\theta(t) = -\nabla\Phi_{\boldsymbol{i}(t)}(\theta(t))\mathrm{d}t + \sqrt{2}\mathrm{d}W_t,$$

Outline

Conclusions

today:

- Implicit regularisation is a vital tool in machine learning; the stochastic gradient descent algorithm can be used as such an implicit regulariser
- the stochastic gradient process is a natural continuous-time variant of stochastic gradient descent
- in convex settings, stochastic gradient processes are stable and converge quickly to their stationary regime.
- the stationary regime may explain implicit regularisation; the strength of regularisation is controlled by the learning rate parameter

related results:

- mildly non-convex/non-smooth optimisation **[L.; 2022]**
- subsampling in particle-based optimisation **[Hanu+L.+Schillings; 2023]**
- subsampling with continuous data and other sampling patterns **[Jin+Liu+L.+Schönlieb; 2023]**


MANCHESTER
1824
The University of Manchester
Jonas Latz
web:latzplacian.org
twitter/latzplacian