A few projects I'm proud of...

Oussama Zekri[1]

A selection of projects I am proud of, and a few points of interest

[1] Research Student, oussama.zekri@ens-paris-saclay.fr

June 6, 2024

école normale supérieure paris—saclay

A few projects I'm proud of...

Summary

1. Maths project : Consistent error bounds
   - Projection algorithm
   - Douglas-Rachford splitting algorithm
   - Main results
2. Code project : Convolutional Kernel Networks
   - CKN but... from scratch !
   - Blogpost incoming
3. Both : SGD through LLMs ICL
   - LLMs understand the convergence of SGD
   - Estimating the trans. kernel of SGD
   - Experiments
4. My affinities, interests and usefulness

Math-oriented project : Consistent error bounds

The Convex Feasibility Problem

The (pretty simple) problem of interest

Let $C_1, ..., C_m$ be closed convex sets included in a finite-dimensional v.s. $\mathcal{E}$ and $C = \cap_{i=1}^m C_i \neq \emptyset$.

Find $x \in C$ **(CFP)**

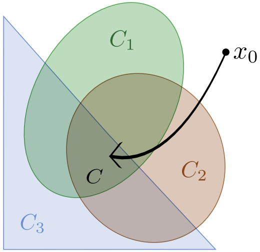
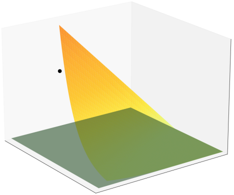
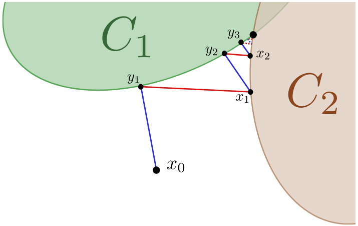


Figure: Illustration of CFP in the case of $m = 3$ sets

Projection algorithm

The first idea that comes to mind : Cyclic projections.





Classic idea for solving the problem : We project alternatively between the C_i sets.

Cyclic projection algorithm (1933)

Require: x_0 and N

1: **for** $k = 0$ to N **do**

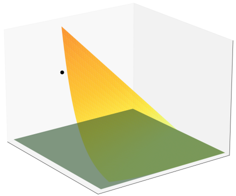
2: $\quad y_{k+1} = p_{C_1}(x_k)$

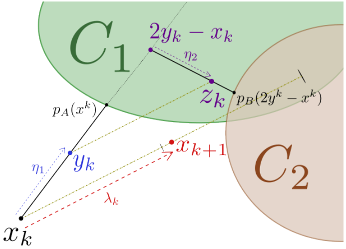
3: $\quad x_{k+1} = p_{C_2}(y_{k+1})$

4: **end for**

Douglas-Rachford splitting algorithm

A slightly more sophisticated algorithm...





We introduce $f(x) = \text{dist}^2(x, C_1)$ and $g(x) = \text{dist}^2(x, C_2)$. We replace projections by proximal operators :

$$\text{prox}_{\eta f}(x) = \frac{1}{2\eta + 1} x + \frac{2\eta}{2\eta + 1} p_{C_1}(x)$$

(Damped)Douglas-Rachford algorithm (1956)

Require: $x_0, N, \eta_1, \eta_2,$ and $(\lambda_k)_k \in (0, 1]$

1: **for** $k = 0$ to N **do**
2: $\quad y_k = \text{prox}_{\eta_1 f}(x_k)$
3: $\quad z_k = \text{prox}_{\eta_2 g}(2y_k - x_k)$
4: $\quad x_{k+1} = x_k + 2\lambda_k(z_k - y_k)$
5: **end for**

Damped Douglas-Rachford

One iteration of the Damped Douglas-Rachford algorithm.

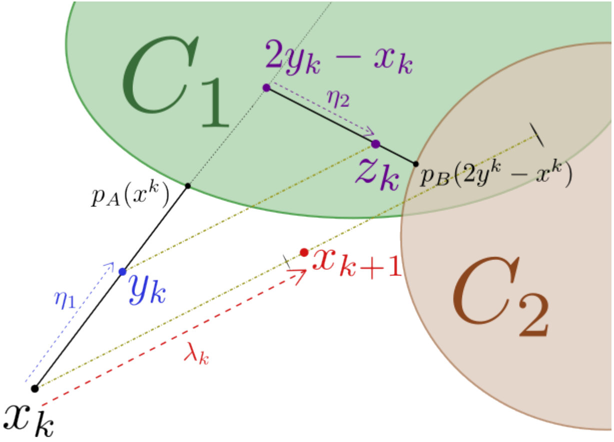


Figure: One iteration of the Damped Douglas-Rachford algorithm

Main results

Consistent error bounds

- Let $C_1, \ldots, C_m$ be closed convex sets included in a finite-dimensional e.v. $\mathcal{E}$ and $C = \cap_{i=1}^m C_i \neq \emptyset$. Find $x \in C$ (CFP)

- (strict) Consitent error bound function for $C_1, \ldots, C_m$[1] : $\Phi : \mathbb{R}^+ \times \mathbb{R}^+ \mapsto \mathbb{R}^+$ such that
  1. $\forall x \in \mathcal{E},\ \text{dist}(x, C) \leq \Phi\left(\max_{1 \leq i \leq m} \text{dist}(x, C_i), \|x\|\right)$
  2. $\forall b \in \mathbb{R}^+, \Phi(., b)$ is monotone (increasing) nondecreasing on $\mathbb{R}^+$, right-continuous at 0 and satisfies $\Phi(0, b) = 0$
  3. $\forall a \in \mathbb{R}^+, \Phi(a, .)$ is monotone nondecreasing on $\mathbb{R}^+$

[1] T. Liu & B.F. Lourenço, Convergence analysis under consistent error bounds

Assumption 1

Let $\{x^k\} \subseteq \mathcal{E}$ be a sequence such that the following conditions hold.

(i) *Fejér monotonicity condition.* For any fixed $c \in C$, it holds that

$$\|x^{k+1} - c\| \leq \|x^k - c\| \quad \forall\ k. \tag{1}$$

(ii) *Sufficient decrease condition.* There exist some positive integer ℓ and nonnegative sequence $\{a_k\}$ with $\sum_{k=0}^{\infty} a_k = \infty$ such that

$$\operatorname{dist}^2(x^k, C) \geq \operatorname{dist}^2(x^{k+\ell}, C) + a_k \max_{1 \leq i \leq m} \operatorname{dist}^2(x^k, C_i) \quad \forall\ k. \tag{2}$$

Main results

The big (and ugly ?) Theorem !

Proposition

Let Assumption 1 holds. Then $\{x^k\}$ converges to some point in C.

Theorem

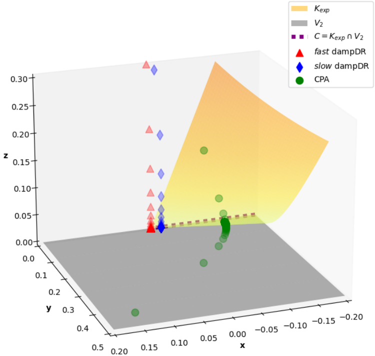
Suppose that Assumption 1 holds. Let Φ be a strict consistent error bound function for $C_1, \ldots, C_m$. Let $\Phi^{\spadesuit}_{\widehat{\kappa}}$ be defined as in Definition 2 b with $\widehat{\kappa}$ such that $\widehat{\kappa} \geq \|x^0\| + 2\,\mathrm{dist}(0,\, C)$. Then, the convergence of $\{x^k\}$ is either finite or

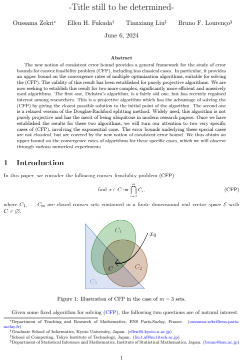
$$\mathrm{dist}(x^k,\, C) \leq \sqrt{(\Phi^{\spadesuit}_{\widehat{\kappa}})^{-1}\left(\Phi^{\spadesuit}_{\widehat{\kappa}}(\mathrm{dist}^2(x^0,\, C)) - \sum_{i=0}^{b_k - 1} a_{k_0 + i\ell}\right)} \quad \forall\ k \geq 2\ell \tag{3}$$

holds for any integer $k_0 \in [0,\, \ell - 1]$ and $b_k := \frac{k-\ell-(k \bmod \ell)}{\ell}$.

Paper incoming

to be submitted really soon to Optimization Letters





-Title still to be determined-

Oussama Zekri[*] Ellen H. Fukuda[†] Tianxiang Liu[‡] Bruno F. Lourenço[§]

June 6, 2024

Abstract

The new notion of consistent error bound provides a general framework for the study of error bounds for convex feasibility problem (CFP), including less classical cases. In particular, it provides an upper bound on the convergence rates of multiple optimization algorithms, suitable for solving the (CFP). The validity of this result has been established for purely projective algorithms. We are now seeking to establish this result for two more complex, significantly more efficient and massively used algorithms. The first one, Dykstra's algorithm, is a fairly old one, but has recently regained interest among researchers. This is a projective algorithm which has the advantage of solving the (CFP) by giving the closest possible solution to the initial point of the algorithm. The second one is a relaxed version of the Douglas-Rachford splitting method. Widely used, this algorithm is not purely projective and has the merit of being ubiquitous in modern research papers. Once we have established the results for these two algorithms, we will turn our attention to two very specific cases of (CFP), involving the exponential cone. The error bounds underlying these special cases are not classical, but are covered by the new notion of consistent error bound. We thus obtain an upper bound on the convergence rates of algorithms for these specific cases, which we will observe through various numerical experiments.

1 Introduction

In this paper, we consider the following convex feasibility problem (CFP)

$$\text{find } x \in C := \bigcap_{i=1}^{m} C_i, \tag{CFP}$$

where $C_1, \ldots, C_m$ are closed convex sets contained in a finite dimensional real vector space $\mathcal{E}$ with $C \neq \varnothing$.

Figure 1: Illustration of CFP in the case of $m = 3$ sets.

Given some fixed algorithm for solving (CFP), the following two questions are of natural interest.

[*]Department of Teaching and Research of Mathematics, ENS Paris-Saclay, France. (oussama.zekri@ens-paris-saclay.fr)
[†]Graduate School of Informatics, Kyoto University, Japan. (ellen@i.kyoto-u.ac.jp)
[‡]School of Computing, Tokyo Institute of Technology, Japan. (liu.t.af@m.titech.ac.jp)
[§]Department of Statistical Inference and Mathematics, Institute of Statistical Mathematics, Japan. (bruno@ism.ac.jp)

Joint work with Ellen H. Fukuda, Bruno F. Lourenço and Tianxiang Liu, realized during the 2nd year internship at the Maths department of ENS Paris-Saclay.

Bibliography

T. Liu and B. F. Lourenço.
Convergence analysis under consistent error bounds.
Found Comput Math, 2022.
arXiv:2008.12968.

J. Douglas and H. H. Rachford.
On the Numerical Solution of Heat Conduction Problems in Two and Three Space Variables.
JSTOR, 1956.

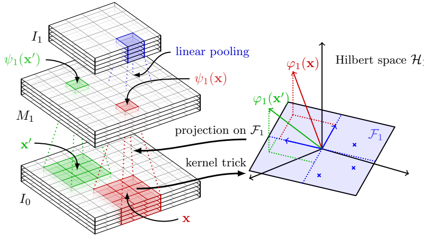
B. F. Svaiter.
On weak convergence of the Douglas-Rachford method.
SIAM Journal on Control and Optimization, 2011.

Code-oriented project : Convolutional Kernel Networks

CKN but... from scratch !

Recoding the whole architecture from scratch

To win the challenge of J. Mairal's course, we decided to implement the CKN. The only requirement of the course was to use Kernel Methods and to code everything from scratch **(no ML librairies !)**.



This implies many challenges, such as

- Recoding automatic differentiation (track correctly the gradients, computational graph etc...)
- In-depth understanding of how the architecture works.
- Harder : Parallelizing the code (with CUDA !)

Blog : logB project.

Our new blog, with Ambroise Odonnat.

A blogpost about this work is being finalized.

The blogpost : https://logb-research.github.io/blog/2024/ckn/

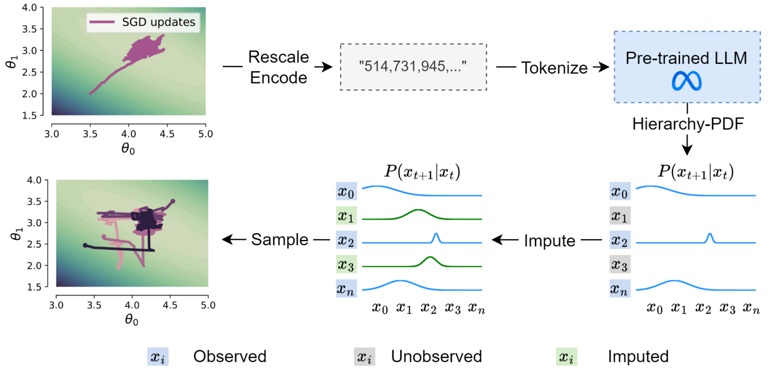
Bibliography

J. Mairal, P. Koniusz, Z. Harchaoui and C. Schmid.
Convolutional Kernel Networks.
NIPS, 2014.
`arXiv:1406.3332`.

J. Mairal.
End-to-End Kernel Learning with Supervised Convolutional Kernel Networks.
NIPS, 2016.

A. Bietti.
Foundations of deep convolutional models through kernel methods.
PhD Thesis, Université Grenoble Alpes, 2019.

A clever mix : Understanding SGD through LLMs ICL abilities



Procedure: ICL for dynamics learning

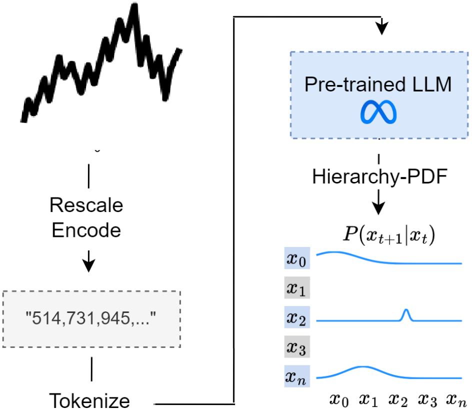
Input: time serie $(x_i)_{i \leq t}$, LLM M, precision k

1. Rescale and encode the time serie with k digits

$$\hat{x}_t = "x_1^1 x_1^2 ... x_1^k, ..."$$

2. Call $M(\hat{x}_t)$
3. Extract the digits logits $(0, 1, 2, 3, \ldots, 9)$
4. Build the next state probability distribution using the Hierarchy-PDF algorithm in [Liu et al., 2024]

Return: predicted transition rules for the observed states: $\{P(X_{i+1}|X_i = x_i)\}_{i \leq t}$



Training set $x = (x_1, \dots, x_N)$ of N i.i.d samples,

$$\min_{\theta} F(\theta), \quad F(\theta) = \frac{1}{N} \sum_{i=1}^{N} f(x_i, \theta), \tag{4}$$

where $\theta \in \mathbb{R}^d$. Minibatch SGD updates :

$$\theta^{t+1} = \theta^t - \gamma_t \nabla \tilde{f}_t(\theta^t) \tag{5}$$

where θ^t denotes the parameters after t iterations, and $\nabla \tilde{f}_t(\theta^t) = \frac{1}{m} \sum_{x \in B_k} \nabla_\theta f(x, \theta^t)$ where B_t is a minibatch of size m of training examples selected randomly.

LLMs understand the convergence of SGD

Overparametrized vs. underparametrized regime

$$\theta^{t+1} = \theta^t - \gamma \nabla \tilde{f}_t(\theta^t) \tag{6}$$

If $\gamma_t = \gamma$, θ^t from (6) form a homogeneous Markov chain that converge to a unique stationary distribution π_γ.

- In the *overparametrized* regime(i.e. when $d >> N$), $\pi_\gamma = \delta_{\tilde{\theta}^*}$ where $\tilde{\theta}^*$ is a specific optimum.
- In the *underparametrized* regime (i.e. when $d << N$), π_γ is a distribution with a strictly positive variance, e.g. $\mathcal{N}(\theta^*, \gamma^{1/2})$ where θ^* is an optimum.

LLMs understand the convergence of SGD

Overparametrized vs. underparametrized regime

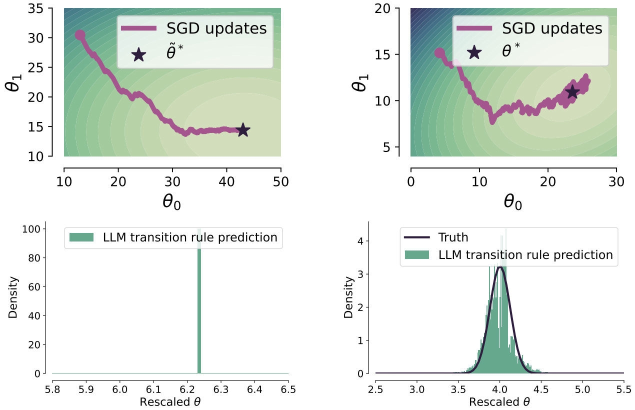


Figure: **Top Left** and **Top Right**, a run of SGD in the overparameterized and underparameterized regimes, respectively. **Bottom Left** and **Bottom Right**, transition probabilities predicted by LLM in overparameterized and underparameterized regimes.

Estimating the transition kernel of SGD

For each parameter $\theta_i, i \in \{1, \dots, d\}$, we consider the one-hot discretized state vector Θ_i^t at time t. Then, we can write

$$\Theta_i^{t+1} = \sum_{j=1}^{d} \lambda_{i,j} P^{(i,j)} \Theta_j^t$$

where $\forall i, j, \lambda_{i,j} \geq 0, \sum_{j=1}^{d} \lambda_{i,j} = 1$ and $P^{(i,j)} = P(\theta_j | \theta_i)$.
Then, the discretized transition kernel of SGD can be seen as a matrix

$$Q = \begin{pmatrix} \lambda_{1,1} P^{(1,1)} & \dots & \lambda_{1,d} P^{(1,d)} \\ \vdots & \ddots & \vdots \\ \lambda_{d,1} P^{(d,1)} & \dots & \lambda_{d,d} P^{(d,d)} \end{pmatrix}$$

which satisfies $\Theta^{t+1} = Q\Theta^t$.

Algorithm

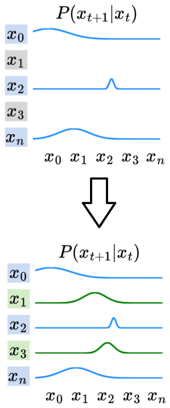
Estimating $P^{(i,i)}$

Input: time serie $(\theta_i^{t+1})_{t\geq 0}$, LLM M, precision k, regularization ε

1. Fill $s < 10^k$ rows of the 10^k rows of $P^{(i,i)}$ with Procedure(θ_i^{t+1}, M, k), denoted as $(P_1^{(i,i)}, \dots, P_s^{(i,i)})$
2. Fill the remaining $10^k - s$ rows of $P^{(i,i)}$ with debiased Sinkhorn barycenter of regularization parameter ε :

for $j = 1$ **to** $s-1$ **do**
  if empty rows between $P_j^{(i,i)}$ and $P_{j+1}^{(i,i)}$ **then**
    Compute debiased Sinkhorn barycenter between $P_j^{(i,i)}$ and $P_{j+1}^{(i,i)}$, with regularization parameter ε
    Fill the empty rows
  end if
end for

Return: Estimated matrix $P^{(i,i)}$



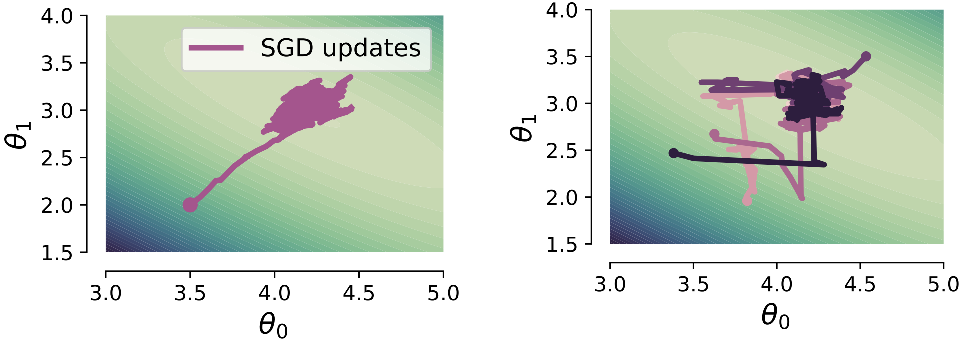


Figure: We optimize F defined in (4) with $f(x_i, \theta) = \frac{1}{2}(\langle x_i, \theta \rangle_{\mathbb{R}^2} - y)^2$ for $d = 2$ and $N = 100$. **Left.** A full SGD run used to learn Q. **Right.** Starting from different initial points, simulating the SGD thanks to Q.

Experiments

Non-convex case

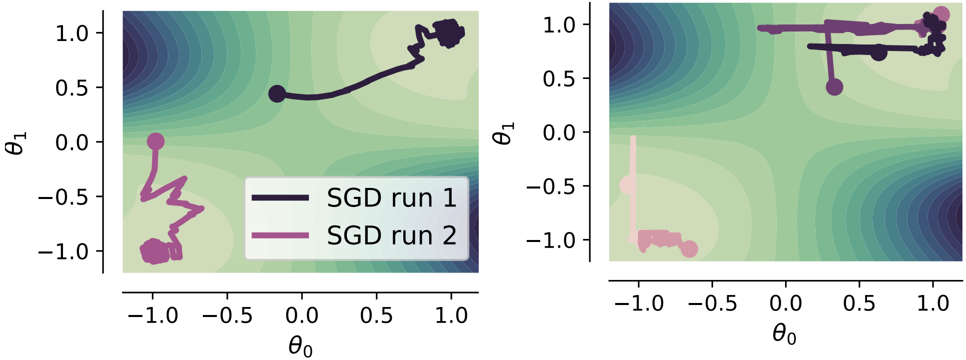


Figure: We optimize F defined in (4) with $f(x_i, \theta) = \frac{1}{2}(\theta_0 \sin(\theta_1 x_i) - y)^2$ for $d = 2$ and $N = 100$. **Left.** A full SGD run used to learn Q. **Right.** Starting from different initial points, simulating the SGD thanks to Q.

Paper submitted

submitted really soon to ICL workshop at ICML 2024

Can LLMs predict the convergence of Stochastic Gradient Descent?

Anonymous Authors[1]

Abstract

Large-language models are notoriously famous for their impressive performance across a wide range of tasks. One surprising example of such impressive performance is a recently identified capacity of LLMs to understand the governing principles of dynamical systems satisfying the Markovian property. In this paper, we seek to explore this direction further by studying the dynamics of stochastic gradient descent in convex and non-convex optimization. By leveraging the theoretical link between the SGD and Markov chains, we show a remarkable zero-shot performance of LLMs in predicting the local minima to which SGD converges for previously unseen starting points. On a more general level, we inquire about the possibility of using LLMs to perform zero-shot randomized trials for larger deep learning models used in practice.

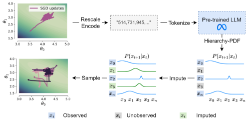


Figure 1. Overview of the proposed approach. After having run SGD on a given optimization problem, we tokenize the obtained iterates and feed them to an LLM of choice. We further use the logits to fill the transition kernel of the Markov chain underlying the SGD with probabilities $P(x_i|x_j)$, while imputing those of its elements that were not observed. Finally, we use the estimate transition kernel to do forecasting for previously unseen inputs.

Joint work with Abdelhakim Benechehab and Ievgen Redko, started 2 months ago during the 3rd year internship at the Maths department of ENS Paris-Saclay.

What I am good at, what interests me and how can I be useful?

Affinity and interests

Because of my background at ENS Saclay, and my personal preferences, **I'm more competent and confident when I'm doing maths**, with some theory. But I also really enjoy coding for my research.

Topics of interest include :

- **Theoretical fields with application to ML** especially Optimization, Kernel Methods and Optimal Transport.
- Differential programming, theory of deep learning and neural nets in general...
- **"Applied" Large-scale ML :** LLMs, ICL...

How can I be useful

I did 2 years of oral exams in preparatory class at LLG, in maths. I am also, for this year 2024, a **member of the HEC entrance exam jury** : I have corrected 300 papers in the written exams, and I will provide 2 weeks of oral exams, on my own exercises.

This could be useful, as I could be a **teaching assistant**, which could help finance part of the potential year.

Thank you for your attention !

Thank you for your attention !

@oussamazekri_

My website : www.oussamazekri.fr