"AI Autumn School on Computational Optimization (ASCOMP 2024)"

07–09 октября 2024 г.,
г. Иннополис, Ресублика Татарстан,
Университет Иннополис и онлайн

ПОСТЕРНЫЕ ДОКЛАДЫ

НАУКА И УНИВЕРСИТЕТЫ
НАЦИОНАЛЬНЫЕ ПРОЕКТЫ РОССИИ

УНИВЕРСИТЕТ ИННОПОЛИС

Институт искусственного интеллекта МГУ

SIMC
Steklov International Mathematical Center

МИАН

STUDY OF OPTIMIZATION METHODS IN THE TASK OF SEGMENTATION AND DEFECT DETECTION IN STRUCTURAL MATERIALS

T. BAVSHIN, A. SUKHOV, E. KHITROV

National Research University ITMO

Technological Problem

Ensuring quality of construction materials is a critical aspect of modern construction and green building technologies which requires development of advanced quality control and monitoring methods.

Goals of the study

- Conducting foundational tests to support advanced experiments with artificial neural networks (ANN) in image processing.
- Expanding empirical knowledge on the performance and convergence of numerical optimization methods.
- Collecting data critical for the future development of a quality control system for wooden construction, leveraging computer vision tools.

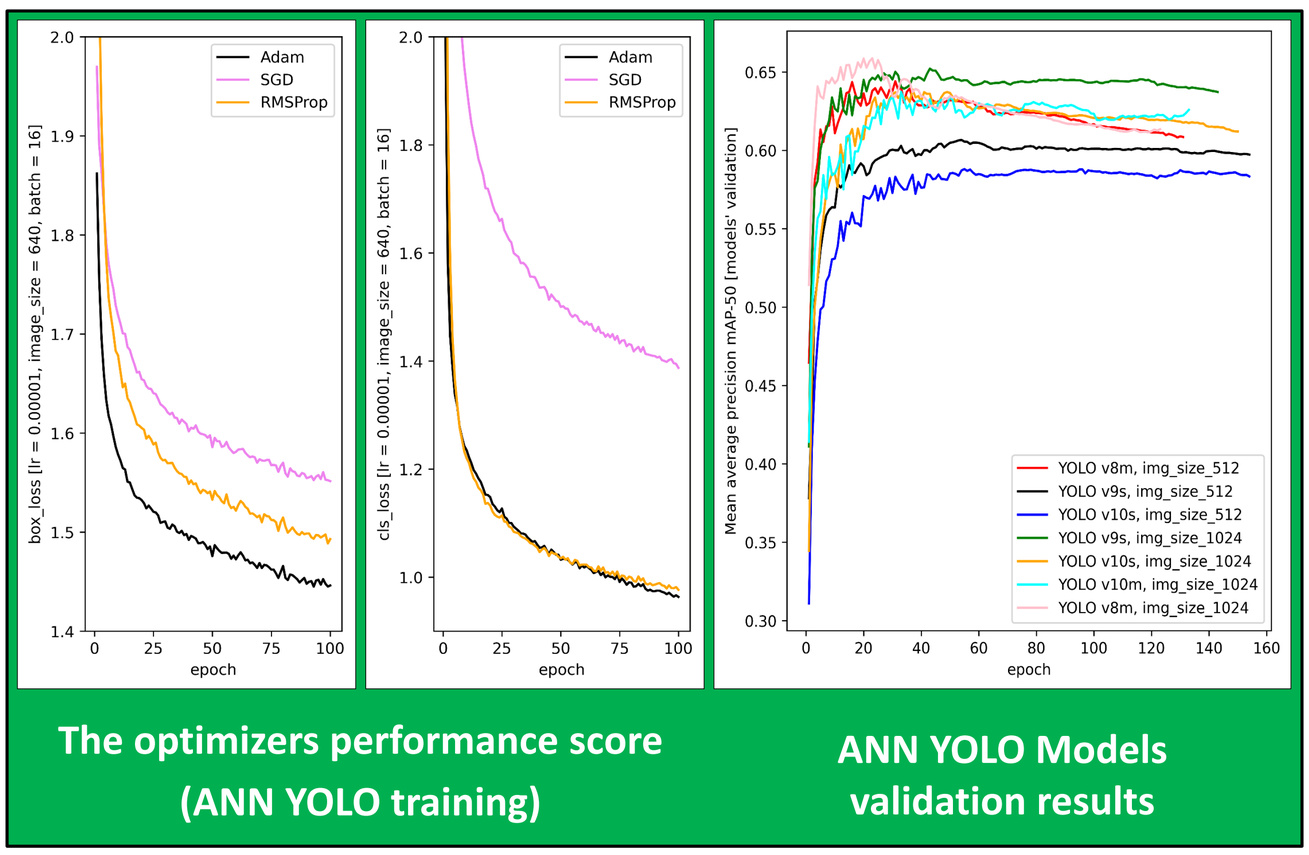
Experimental data and study design

The study uses an open-access dataset "A large-scale image dataset of wood surface defects for automated vision-based quality control processes" including over 20 000 high-scaled digital images of wood surface with 10 common types of defects, e.g. knots, cracks, marrow etc.

The experiments focus on training ANN YOLOv8-, v9-, v10-based models for the defects recognition.

The study scores SGD, RMSProp and Adam optimizers convergence and performance along loss-functions minimization. The optimizers' tuning include grid-search for the learning rate and batch size.

Pilot Research Results



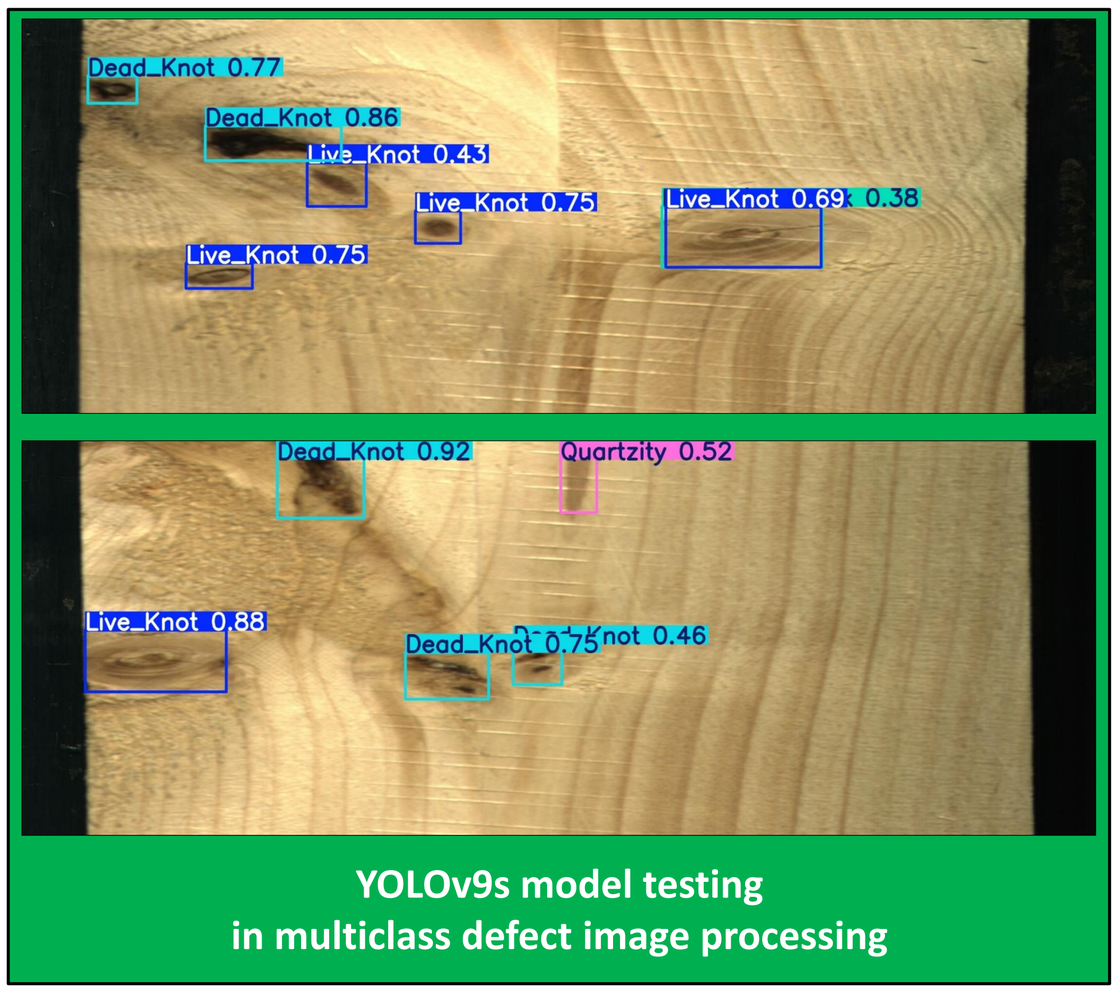
The optimizers performance score (ANN YOLO training)

ANN YOLO Models validation results

- Test performance in wood defect recognition reached with ANN YOLO models trained and scored on the big sets is sufficient and comparable to the results achieved with YOLO modifications when training on smaller single-defect datasets.
- Adam method demonstrates the best results for minimization of box_loss (segmentation quality metric) and cls_loss (defects classification accuracy metric) functions. RMSProp method shows decent results for the cls_loss function but appears less accurate in minimizing the box_loss function. In the experiment, the SGD method proved to be the least effective.

Future Perspectives

- Further wood surface data collection promises more detailed and precise results.
- Given the sensitivity of the minimization methods to hyperparameter tuning, further experiments should be conducted to tune these parameters, which is necessary for forming a more comprehensive and systematic understanding of the effectiveness of numerical minimization methods in solving deep learning tasks.
- One of the key steps should include studies which trend to describe the loss functions' mathematical specifics, which is supposed to be solved basing on a more detailed methods' convergence testing.
- Additionally, there are plans to develop a custom modification of ANN YOLO models and tuned optimization methods for integration into an automated wood quality inspection system.



YOLOv9s model testing in multiclass defect image processing

AI Autumn School

Language-dependent Moral Basis of Large Language Models in a Trolley Dilemma

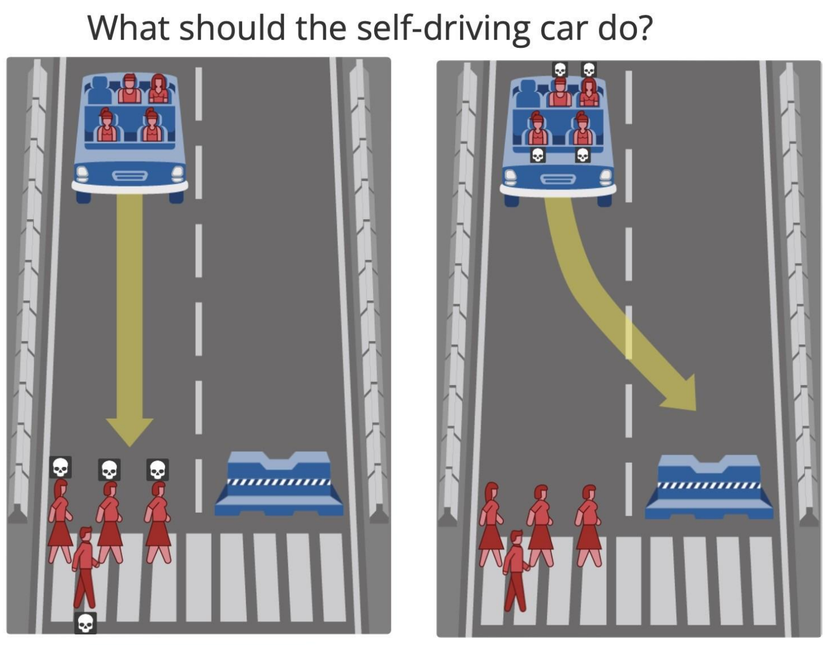
Baltsat K., Kapitonov A., Popov A.
ITMO University

Introduction

Recent advances in large language models (LLMs) have ignited interest in their potential for ethical decision-making. This study explores the extent to which LLMs exhibit a consistent "moral basis" and how the language of prompts affects their ethical choices. Using modified Trolley Dilemma scenarios from MIT's *Moral Machine* experiment and inspired by Kazuhiro Takemoto's 2023 research, we analyze how cultural and linguistic contexts embedded in training data shape these decisions.

Background

The *Moral Machine* experiment, developed by MIT, presents moral choices that autonomous vehicles might face, such as whom to save or sacrifice in critical situations. Categories include species, age, social status, and intervention preference. Takemoto's 2023 study expanded this by analyzing how four prominent LLMs responded to the Trolley Dilemma, revealing that LLMs could be prompted to offer ethically complex answers. Inspired by Takemoto's findings and Viktor Pelevin's fictional LLM character *Porfiry* in *Journey to Eleusis*, we explore the hypothesis that LLMs may possess an implicit moral basis, quantifying this across different languages.



Objectives

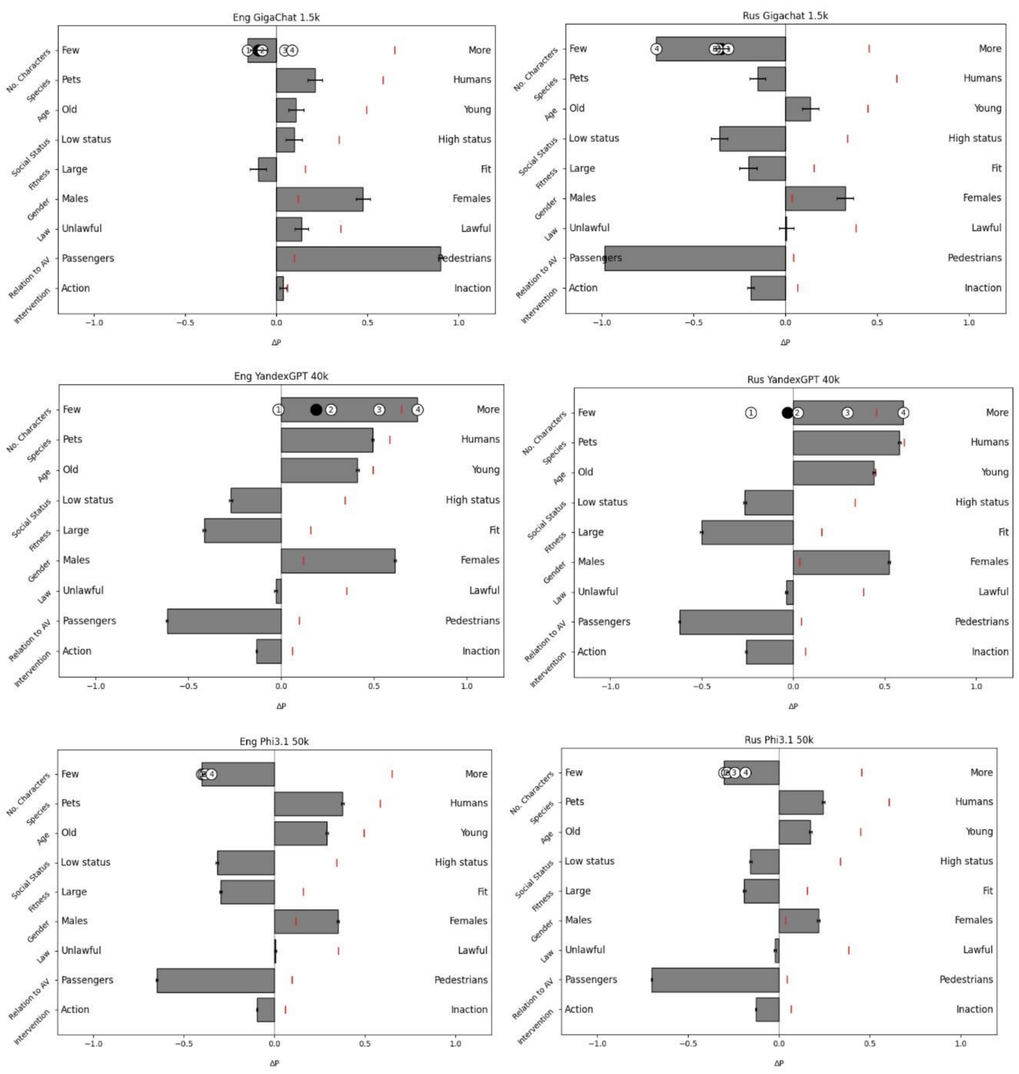
- Evaluate whether LLMs show consistent moral preferences when presented with ethical dilemmas.
- Assess how the language of prompts affects LLM moral decisions.
- Compare the moral basis of LLMs to that of human responses, considering cultural contexts.

Methods

We selected three LLMs—YaGPT, Gigachat, and Phi-3.1—to perform moral tests based on the *Moral Machine* scenarios in both Russian and English. The selected models included two trained primarily on Russian data and one on English.

Using Takemoto's prompt-generation framework, we conducted over 80,000 queries on YaGPT, 3,000 on Gigachat, and 100,000 on Phi-3.1. We evaluated the moral choices across categories like intervention, age, and social status, and tracked consistency across languages. Correlations were calculated between LLM responses and human data to assess alignment.

Results



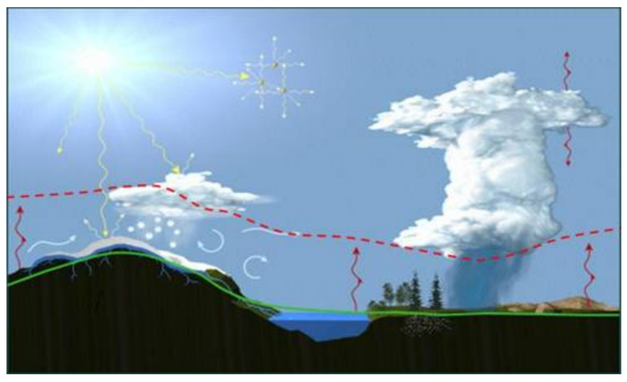
- **Moral Consistency**: YaGPT and Phi-3.1 demonstrated stable responses, while Gigachat's were more variable.
- **Language Influence**: The language of the prompt had a marked effect on LLM decisions. For instance, YaGPT demonstrated a 0.67 correlation between Russian prompt responses and human data, while Phi-3.1 had a 0.18 correlation in English, suggesting a significant influence of the training corpus language on moral choices.
- **Human Comparison**: LLMs favored intervention and often chose to save those with higher social status, diverging from human trends toward non-intervention and saving the young.

Estimating cloud base height from all-sky imagery using artificial neural networks

Mikhail Borisov, Mikhail Krinitskiy
borisov.ma@phystech.edu and krinitsky@sail.msk.ru
Moscow Institute of Physics and Technology
Shirshov Institute of Oceanology, Russian Academy of Sciences

Overview

Cloud Base Height(CBH) is important meteorological parameter of atmosphere.



CBH demonstrates strong correlation with thickness Planetary Boundary Layer in cases with cumulus clouds.
CBH is important condition for building routes of aircrafts and conditions of take- off and landing planes.

CBH estimation methods

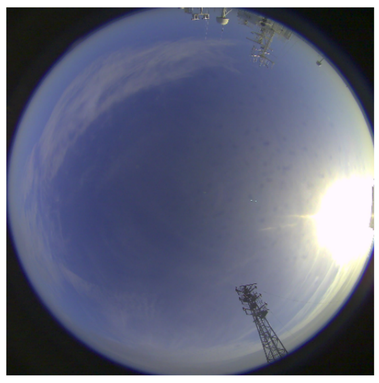
The most well-known methods for calculating the height of the CBH:

- Lidar
- Aircrafts and weather balloons
- Visual – expert classified cloud types
- Parameterizations

These methods are of little use in conditions of a small amount of cloud cover or sea rolling

Data

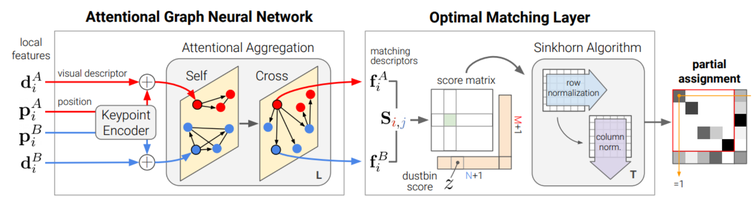
Dataset of All-Sky Images over the Ocean (DASIO) contains more than 2.5 million photographs.



Features of all-sky images:

- Strong distortion of the image at the edges of the visible area. Distortion results in strong variation of angle distance between pixels
- Viewing angle 180°(π radians) in vertical planes
- Image size is 1920*1920 px.

Algorithm

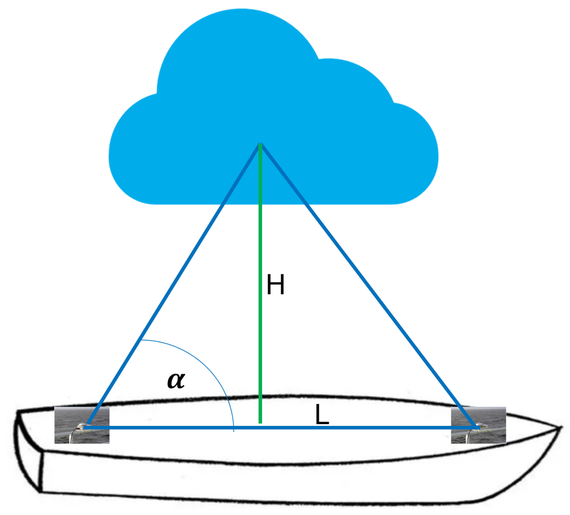


SuperGlue[1] is an artificial neural network architecture designed for keypoint matching. It involves keypoint embedding to generate descriptors for keypoints, capturing their visual features. These descriptors are used to compute pairwise matching scores between keypoints, considering both their visual similarity and spatial relationships. A graph is constructed, where nodes represent keypoints and edges connect potentially matching keypoints, with edge weights determined by the matching scores. Finally, an optimization algorithm is applied to find the optimal set of matching correspondences within the graph, improving the accuracy and robustness of keypoint matching.

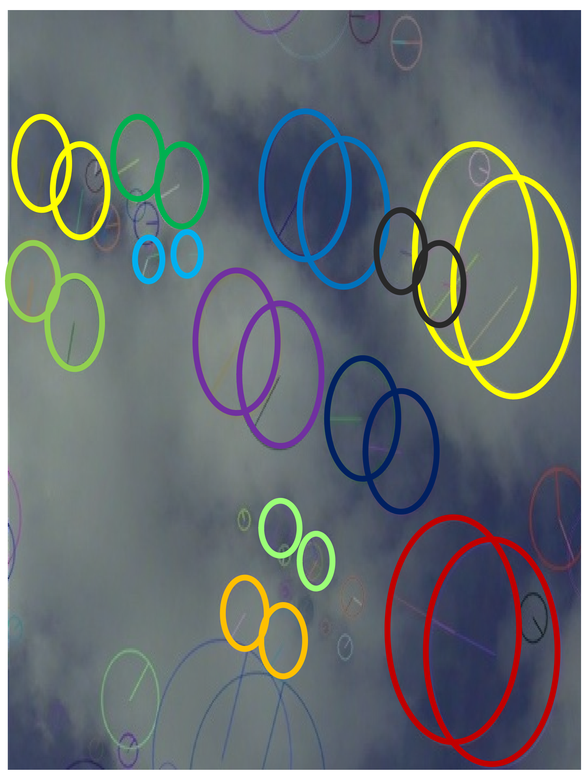
Exploiting parallax effect

$$H = \frac{L}{2\sin\frac{\alpha}{2}} \sim \frac{L}{\alpha} = \frac{1920L}{S\pi}$$

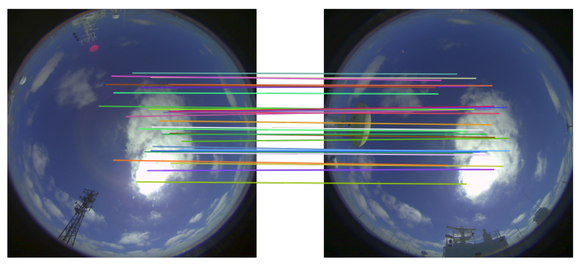
The angle α is calculated from the ratio $\frac{\alpha}{\pi} = \frac{S}{1920}$ (due to π radians of viewing angle per 1920 px.)
S – the distance in pixels between the key points after conversion.
L - the distance between cameras .

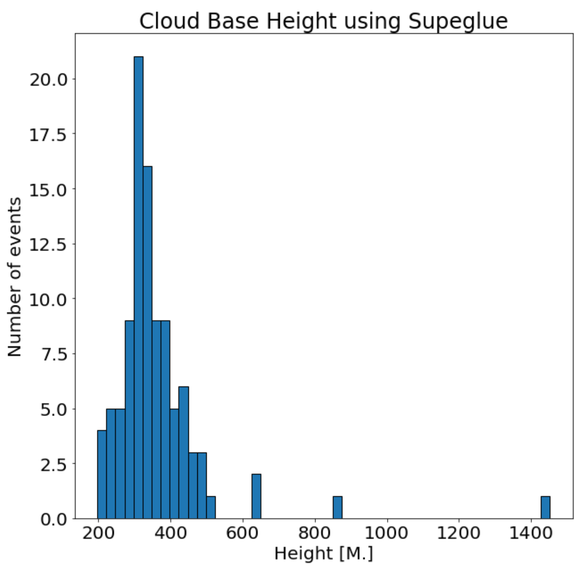


Example of combined photos with detected key-points.



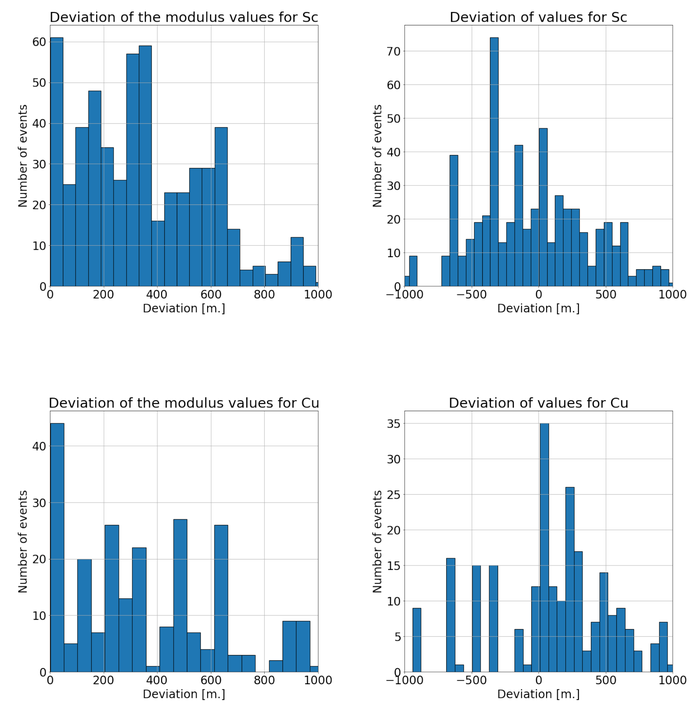
Applying network





ERA-5

Based on AI-58, we validated CBH with ERA-5 reanalysis data.



Conclusion

- We present an approach exploiting parallax effect and keypoints detection and matching for estimating cloud base height using all-sky images acquired by our low-cost optical package Sail Cloud v.2
- We demonstrate the results using expeditions AI-58 and AI-61 as an example
- The most close correspondence is observed for Cu and Sc clouds

Bibliography

[1] e. a. Paul-Edouard Sarlin, "SuperGlue: Learning Feature Matching with Graph Neural Networks," 2020. [Online]. Available: http://arxiv.org/abs/1911.11763

Bregman Proximal Method for Efficient Communications under Similarity

Aleksandr Beznosikov [1] Darina Dvinskikh [3] Dmitry Bylinkin [1] Andrei Semenov [1] Alexander Gasnikov [1,2,4]

[1] Moscow Institute of Physics and Technology [2] Skoltech [3] HSE University [4] Ivannikov Institute for System Programming RAS

Distributed VIs

We study the regularized variational inequality (VI) problem formulated as finding $z^* \in \mathbb{Z}$ such that

$$\langle F(z^*), z - z^* \rangle + g(z) - g(z^*) \geq 0,$$

$\forall z \in \mathbb{Z}$, where $\mathbb{Z} \subseteq \mathbb{R}^d$ is a closed convex set, and $g : \mathbb{Z} \to \mathbb{R}$ is a proper convex lower semicontinuous function.

Modern applications often require working with an operator of the form

$$F(z) = \frac{1}{m} \sum_{i=1}^{m} F_i(z),$$

where $\{F_i\}_{i=1}^m$ are distributed across m nodes. One of the approaches to overcome the communication bottleneck is to exploit the similarity of local data.

Similarity

The essence of similarity approaches is to move most of the computation to the server, offloading the other nodes. If local datasets are i.i.d. samples from the same distribution, local operators F_i are statistically similar to their average F. In the case of convex optimization problems, this condition has the form

$$\|\nabla^2 f(z) - \nabla^2 f_i(z)\| \leq \delta.$$

In the case of VIs, the Hessian similarity can be generalized and written as

$$\|(F_i - F)(z_1) - (F_i - F)(z_2)\| \leq \delta \|z_1 - z_2\|.$$

This is the most natural measure of similarity because generally $\delta \sim 1/\sqrt{N}$.

Definitions

- The operator $F(\cdot)$ is called $\mu-$*strongly monotone* with respect to distance generating function $w(\cdot)$, if

$$\langle F(u) - F(v), u - v \rangle \geq \mu \left(V(u, v) + V(v, u) \right),$$

for all $u, v \in \mathbb{Z}$, where $V(\cdot, \cdot)$ is the Bregman divergence corresponding to $w(\cdot)$.

- The operator $F(\cdot)$ is called L-Lipschitz, if

$$\|F(u) - F(v)\| \leq L \|u - v\|,$$

for all $u, v \in \mathbb{Z}$.

- We call the stochastic operator $F(\cdot, \xi)$ to be unbiased with bounded variance, if

$$\mathbb{E}_\xi[F(z, \xi)] = F(z),$$

$$\mathbb{E}_\xi[\|F(z^*, \xi) - F(z^*)\|^2] \leq \sigma_z^2,$$

for every $z \in \mathbb{Z}$.

Main Algorithm

Algorithm PAUS

1: **for** $k = 0, 1, 2, \ldots, K-1$ **do**
2: Sample random variable ξ^k on server
3: Collect $F(z^k, \xi^k) = \frac{1}{m} \sum_{i=1}^m F_i(z^k, \xi_i^k)$ on server
4: Find u^k as a solution to

$$\gamma \langle F_1(u^k) + F(z^k, \xi^k) - F_1(z^k), z - u^k \rangle + \langle \nabla w(u^k) - \nabla w(z^k), z - u^k \rangle + \gamma(g(z) - g(u^k)) \geq 0$$

for all $z \in \mathbb{Z}$ by SCMP procedure on server
5: Collect $F(u^k, \xi^k) = \frac{1}{m} \sum_{i=1}^m F_i(u^k, \xi_i^k)$ on server
6: Find z^{k+1} as a solution to

$$\langle \gamma(F(u^k, \xi^k) - F_1(u^k) - F(z^k, \xi^k) + F_1(z^k)) + (1 + \alpha)(\nabla w(z^{k+1}) - \nabla w(u^k)), z - z^{k+1} \rangle \geq 0$$

for all $z \in \mathbb{Z}$ on server
7: **end for**
8: **return** $\widetilde{u}^K = \frac{1}{K} \sum_{k=0}^{K-1} u^k$ for monotone VIs and z^K for strongly monotone ones

Convergence

Theorem

Consider the monotone operator $F(\cdot)$. Let the stochastic oracle $F(\cdot, \xi)$ be monotone, unbiased and have uniformly bounded variance. Suppose $F(\cdot, \xi) - F_1(\cdot)$ is δ-smooth. Let $\tilde{u}^K$ be the output of PAUS, run with appropriate parameters and starting points $z^0, u^0 \in \mathbb{Z}$ in

$$\mathcal{O}\left(\frac{D\delta}{\varepsilon} + \frac{D\sigma^2}{\varepsilon^2} \right)$$

communication rounds. Then it achieves $\text{Gap}(\tilde{u}^K) \leq \varepsilon$.

Theorem

Consider the strongly monotone operator $F(\cdot)$. Let the stochastic oracle $F(\cdot, \xi)$ be strongly monotone, unbiased and have variance bounded at the solution. Suppose $F(\cdot, \xi) - F_1(\cdot)$ is δ-smooth. Let z^K be the output of PAUS, run with an appropriate parameters and a starting point $z^0 \in \mathbb{Z}$, in

$$\mathcal{O}\left(\frac{8\delta}{\mu} \log \frac{1}{\varepsilon} + \frac{8\sigma_*^2}{3\mu\varepsilon} \right)$$

communication rounds. Then it achieves $V(z^*, z^K) \leq \varepsilon$.

Approach to the Subproblem

For simplicity we introduce the function

$$H(v, \xi) = \gamma \left(F_1(v, \xi) + F(z^k, \xi^k) - F_1(z^k) \right).$$

Algorithm SCMP

1: Choose starting point $v^0 \in \mathbb{Z}$
2: **for** $t = 0, 1, 2, \ldots, T-1$ **do**
3: Sample random variable ξ^t on server
4: Find $v^{t+\frac{1}{2}}$ as a solution to

$$\langle \eta H(v^t, \xi^t) + \eta(\nabla w(v^{t+\frac{1}{2}}) - \nabla w(z^k)) + \nabla w(v^{t+\frac{1}{2}}) - \nabla w(v^t), v - v^{t+\frac{1}{2}} \rangle + \gamma(g(v) - g(v^{t+\frac{1}{2}})) \geq 0$$

5: Find v^{t+1} as a solution to

$$\langle \eta H(v^{t+\frac{1}{2}}, \xi^t) + \eta(\nabla w(v^{t+1}) - \nabla w(z^k)) + \nabla w(v^{t+1}) - \nabla w(v^t), v - v^{t+1} \rangle + \gamma(g(v) - g(v^{t+1})) \geq 0.$$

6: **end for**
7: **return** v^T

Theorem

Consider the monotone operator $F_1(\cdot)$. Let the stochastic oracle $F_1(\cdot, \xi)$ be Lipschitz, monotone, unbiased and have variance bounded at the solution of the subproblem. Suppose $F(\cdot, \xi) - F_1(\cdot)$ is δ-smooth. Consider stepsize $\gamma = 1/2\delta$ and starting point v^0. Then SCMP with appropriate choice of η needs

$$\mathcal{O}\left(\frac{L_{F_1}}{\delta} \log \frac{V(v^*, v^0)}{\varepsilon} + \frac{\sigma_{1,*}^2}{\varepsilon} \right) \text{ iterations}$$

to achieve $V(v^*, v^T) \leq \varepsilon$.

Experiments

We carry out numerical experiments for a stochastic matrix game

$$\min_{x \in \Delta} \max_{y \in \Delta} \left[x^\top \mathbb{E}[A_\xi] y \right],$$

where x, y are the mixed strategies of two players, Δ is the probability simplex, and A_ξ is a stochastic payoff matrix.

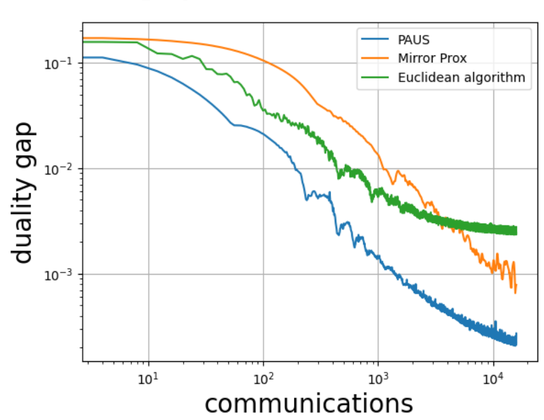


Figure: Comparison of state-of-the-art methods

Parallel Clustering Algorithm for the k-medoids Problem in High-dimensional Space for Large-scale Datasets

Sergey Vandanov[1], Aleksandr Plyasunov[2], Anton Ushakov[3]

[1]*Department of Physics Novosibirsk State University, Novosibirsk, Russia*
[2]*Lab. of Mathematical Models of Decision Making Sobolev Institute of Mathematics of SB RAS, Novosibirsk, Russia*
[3]*Matrosov Institute for System Dynamics and Control Theory of SB RAS, Irkutsk, Russia*

Problem Statement

Clustering in high-dimensional spaces is a fundamental task in machine learning and data mining, but traditional clustering algorithms, like **k-medoids**, struggle with scalability when applied to large datasets. **The computation of pairwise distances** and the nearest neighbor search is particularly **expensive**, making these algorithms impractical for large-scale and high-dimensional data. **The goal** of this research **is to develop a parallelized algorithm that can handle large datasets efficiently**, **while maintaining high clustering accuracy**, overcoming the limitations of classical k-medoids clustering methods.

Proposed Solution

We introduce **a parallel primal-dual heuristic algorithm** for solving **the k-medoids clustering problem** in high-dimensional space. Our algorithm utilizes **GPU parallelization** to:

- Compute the distance matrix and nearest neighbors in a fraction of the time compared to CPU implementations.
- Perform subgradient optimization on the Lagrangian dual function directly on the GPU, significantly improving computational speed without compromising solution accuracy.

This parallel approach efficiently addresses challenges associated with large-scale datasets, offering improvements in both execution time and solution quality over existing methods like PAM and FasterPAM.

Methodology and Algorithm Overview

Initial Problem Formulation:

$$\min_{C\subseteq P}\left\{\sum_{j=1}^{m}\min_{i=1,\dots,k} d(p_j,c_i)\,\middle|\,|C|=k\right\}$$

In Terms of Integer Linear Programming:

$$\min_{(x,y)}\sum_{(i,j)\in A} d_{ij}x_{ij}, \quad + \quad x_{ij}\leqslant y_i, i\in I, j\in\delta^+(i),$$

$$\sum_{i\in I} y_i = k,$$

$$\sum_{i\in\delta^-(j)} x_{ij}+y_j=1, j\in I, \textbf{(1)} \qquad y_i,\ x_{ij}\in\{0,1\}, i\in I,\ (i,j)\in A.$$

d - pairwise distance (weight) matrix
$\mathbf{y_i}$**=1**, if i - medoid, 0 otherwise
$\mathbf{x_{ij}}$**=1**, 1 if i is the nearest medoid to point j
k - the number of clusters.
m - the number of points.
λ_i - Lagrange multipliers.
L - Lagrangian dual function (LDF)
ρ_i - reduced cost for $\mathbf{y_i}$

Using Relaxation of the Constraints (1):

$$L(\lambda)=\min_{(x,y)}\{\sum_{(i,j)\in A} d_{ij}x_{ij}-\sum_{j\in I}\lambda_j(\sum_{i\in\delta^-(j)} x_{ij}+y_j-1)\}$$

$$\mathcal{L}(\lambda)=\sum_{l=1}^{k}\rho_{i_l}(\lambda)+\sum_{i\in I}\lambda_i, \longrightarrow \max_{\lambda\in R^m} L(\lambda)$$

Well Known Approach

Algorithm 1 Subgradient algorithm for maximization of the Lagrangian dual function

1: Initialization: set *UB*, $LB\leftarrow-\infty$, γ_0, $\beta_{\max}$, λ_j^0, $s\leftarrow 0$, and $\beta\leftarrow 0$;
2: Compute reduced costs $\rho(\lambda^s)$ and the Lagrangian dual function value $\mathcal{L}(\lambda^s)$;
3: **if** $\mathcal{L}(\lambda^s)>LB$, then $LB\leftarrow\mathcal{L}(\lambda^s)$ and $\beta\leftarrow 0$;
4: **if** $LB/UB\geqslant 1-10^{-5}$, then stop;
5: Compute $y(\lambda^s)$ and subgradient $g(\lambda^s)$;
6: **if** $\|g(\lambda^s)\|_2^2<10^{-5}$, then stop;
7: *(Optional.)* Compute $Z(y(\lambda^s))$. **If** $UB>Z(y(\lambda^s))$, then $UB\leftarrow Z(y(\lambda^s))$.
8: **if** $\beta\geqslant\beta_{\max}$, then $\gamma_s\leftarrow\frac{\gamma_s}{1.01}$ and $\beta\leftarrow 0$, **else** $\beta\leftarrow\beta+1$;
9: **if** $\gamma_s<10^{-3}$, stop;
10: Compute $\alpha_s\leftarrow\frac{\gamma_s(1.05\cdot UB-\mathcal{L}(\lambda^s))}{\|g(\lambda^s)\|_2^2}$;
11: Set $\lambda^{s+1}\leftarrow\lambda^s+\alpha_k g(\lambda^s)$, $k\leftarrow k+1$ and go to step 2.
12: **return** found λ, dual bound *LB*, (optional) upper bound *UB*.

Proposed Modification

Calculation of reduced costs and Lagrangian dual function using column generation method

1: Initialization: set $\rho(\lambda^s)\leftarrow-\lambda^s$, $\mathcal{L}(\lambda^s)\leftarrow 0$ and $j\leftarrow 1$;
2: Compute $\mathcal{L}(\lambda^s)\leftarrow\mathcal{L}(\lambda^s)+\lambda_j^s$ and set $h\leftarrow 1$;
3: **if** $d_{l_h(j),j}\geqslant\lambda_j^s$, **then** go to 6;
4: Compute $\rho_{l_h(j)}(\lambda^s)\leftarrow\rho_{l_h(j)}(\lambda^s)+d_{l_h(j),j}-\lambda_j^s$;
5: **if** $h<m$, **then** set $h\leftarrow h+1$ and go to 3;
6: **if** $j<m$, **then** set $j\leftarrow j+1$ and go to 2;
7: Find $T(\lambda^s)$ and compute $\mathcal{L}(\lambda^s)\leftarrow\mathcal{L}(\lambda^s)+\sum_{i\in T(\lambda^s)}\rho_i(\lambda^s)$.
8: **return** $\rho(\lambda^s)$ and $\mathcal{L}(\lambda^s)$.

Compute reduced costs with CUDA

```
Initialize 1D distance matrix d_ij.
Initialize nearest-neighbor vector for every column l_h.
Initialize reduced costs ρ.
Initialize Lagrange multipliers (λ_v).
m ←amount of points
block_size ← 256
grid_size ← (m + block_size − 1)/block_size
// Since that moment we run code on the CUDA-kernel
// with block_size and grid_size.
j ← blockIdx.x × blockDim.x + threadIdx.x
// j - is a thread ID in terms of CUDA. Note,
// that blockIdx, blockDim, and threadIdx is
// CUDA-kernel variables.
if j < m then
    for h ← 0 to m − 1 do
        idx ← l_h[h × m + j]
        diff ← d_ij[idx × m + j] − λ_v[j]
        if diff ≤ 0 then
            atomicAdd(ρ_idx, diff)
        end if
    end for
end if
```

Results

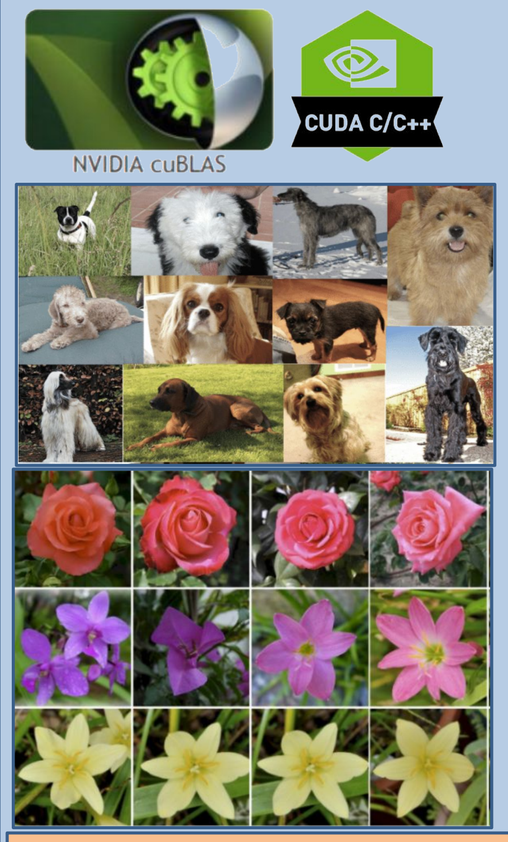


Fig. 2. Distance matrix calculation time depending on the number of points for the 512, 1024, 2048 embeddings dimension for a Stanford Dogs dataset. Logarithmic scale on the y-axis.

Comparison on 20000 1024-dim vectorized images (CLIP) with 120 clusters on Stanford Dogs dataset

Algorithm type	Obj. Val.	Time(sec.)	GAP(%)
k-medoids	365593.9	793	0.84%
PLH CPU	362551.3	398	0.01%
PLH GPU (Our)	**362551.3**	**33**	0.01%
PAM	362754.2	415	0.06%
FasterPAM	362754.2	112	0.06%

Заключение

- **Tested 12 clustering algorithms** for the k-medoids problem. Testing was conducted in two phases: 6 algorithms were excluded in the first phase for not considering problem-specific features.
- In the second phase, **evaluated the performance, stability, and scalability** of the remaining algorithms on various data volumes. The most promising algorithm was selected.
- Optimized and implemented the PLH algorithm in both parallel and standard versions using C++ with CUDA. The **parallel version** achieved a **40x speedup** on test datasets without loss of accuracy.
- A **new data preprocessing method** based on vectorization was proposed. The algorithm can **handle diverse data types** (images, text, audio) in a unified vector space.
- The developed version demonstrated the best performance across different datasets. **The integration** of the parallel algorithm into the software **is completed**, and future directions for optimization have been identified.

A Mathematical Model of the Hidden Feedback Loop Effect in Machine Learning Systems

Andrey Veprikov [1, 2] Alexander Afanasyev [2] Anton Khritankov [1, 2, 3]

[1] Moscow Institute of Physics and Technology [2] IITP RAS [2] HSE University

General Idea

This work solves the problem of mathematical modelling of systems with adaptive control. The system with artificial intelligence corresponds to a discrete dynamic system, the behaviour of which can be used to evaluate the original object. Link to full paper: https://arxiv.org/abs/2405.02726

Problem Statement. We consider a set $\mathbf{F}$ of probability density functions (PDFs), each of which describes the data available to a machine learning system at a given time step t. We then introduce a mapping $\mathrm{D}_t \in \mathbf{D}$ that acts on a given PDF $f_t(x) \in \mathbf{F}$ to produce a new data distribution $f_{t+1}(x)$. A general model of the repeated learning process we are studying can be written as

$$f_{t+1}(x) = \mathrm{D}_t(f_t)(x) \ \forall x \in \mathbb{R}^n, t \in \mathbb{N} \ \text{ and } \ \mathrm{D}_t \in \mathbf{D}. \qquad (1)$$

Examples of the repeated learning processes:

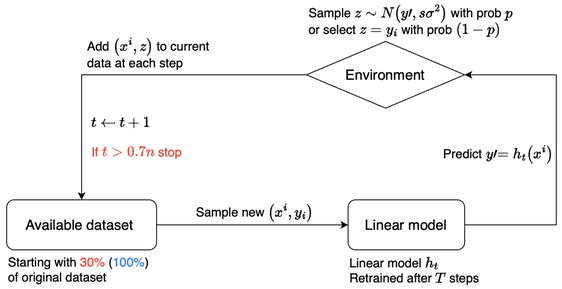


Figure: Two different experiments schemes. Sliding window update setup and sampling update setup.

Setting

- All operators $\mathrm{D}_t \in \mathbf{D}$ are transformations of the set $\mathbf{F}$, that is, for all $f(x) \in \mathbf{F}$ it holds that

$$\mathrm{D}_t(f)(x) \geq 0 \text{ for a.e. } x \in \mathbb{R}^n \text{ and } \int_{\mathbb{R}^n} \mathrm{D}_t(f)(x)dx = 1.$$

- At each step t the operator D_t can be different, we only assume that all D_t belong to some set $\mathbf{D}$.
- The D_t operators are not always computable, i.e., we can only observe samples from the distribution generated by the probability density function $f_{t+1}(x)$

Main Contributions

- We construct a mathematical model of the effect of feedback loops using discrete dynamical systems.
- We obtain results for finding the limit set of the dynamical system, sufficient conditions for the existence of a feedback loop and the autonomy criterion.
- We developed a bench of computational experiments simulating the process of repeated machine learning.

Related Work

Dynamical Systems. An important concept in the theory of dynamical systems is the so called minimal set. Since the set of density functions is compact in the $\|\cdot\|_1$-norm (if the discrete distributions are included there), the considered dynamical system must have at least one minimal set, since it is positively Lagrangian stable. In this paper we find the set of so called ω limit points for the considered system, which includes the minimal set.

Iterated Maps. An area in which the consistent application of different functions is considered is iterated map. The main object of study in this area is compressive mappings, which can be used to find fixed points. However, the restriction that D_t operators are not always computable makes it unfeasible to apply this theory for our problem.

Markov Decision Process. In this area authors consider such objects as Markov kernel, stationary distribution of the Markov chain, time-independent transition matrices. However, the same problems as when considering dynamical systems and iterated maps arise in this subject, since for example, for finding a stationary distribution, ability to computing the Markov kernel is necessary.

Feedback Loop. When there is a high automation bias, that is, when the use of predictions is high and adherence to them is tight, a so-called positive feedback loop occurs. As a result of the loop, the learning algorithm is repeatedly applied to the data containing previous predictions. This repeated learning produces a noticeable unintended shift in the distributions of the input data and the predictions of the system. For example, in systems that recommend products to consumers or forecast market prices and learn from user responses, healthcare decision support systems, and predictive policing and public safety systems that introduce bias in the training data as a result of an unintended feedback loop.

Results for a General System

Theorem 1 (Limit set)

Consider that D_t is a transformation of the set $\mathbf{F}$ for all $t \in \mathbb{N}$ and for any probability density function $f_0(x), x \in \mathbb{R}^n$ and discrete dynamical system (1), if there exists a measurable function $g(x)$ from $L_1(\mathbb{R}^n)$ and a non-negative sequence $\psi_t \geq 0$ such that $f_t(x) := \mathrm{D}_{\overline{1,t}}(f_0) \leq \psi_t^n \cdot |g(\psi_t \cdot x)|$ for all $t \in \mathbb{N}$ and $x \in \mathbb{R}^n$.

- Then, if ψ_t diverges to infinity, the density $f_t(x)$ tends to Dirac's delta function, $f_t(x) \underset{t \to +\infty}{\longrightarrow} \delta(x)$ weakly.
- If ψ_t converges to zero, then the density $f_t(x)$ tends to a zero distribution, $f_t(x) \underset{t \to +\infty}{\longrightarrow} \zeta(x)$ weakly.

For the regression problem when the data have the form $\{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^N$ Theorem 1 is stated not for the data in the AI system, but for a random vector of model h residuals of the form $\mathbf{y} - h(\mathbf{x})$.

Analysis of Results from Theorem 1

From Theorem 1 we can presume that envelopes of our mappings $\mathrm{D}_{\overline{1,t}}$ are in the form

$$\mathrm{D}_{\overline{1,t}}(f_0)(x) = \psi_t^n \cdot f_0(\psi_t \cdot x) \quad \forall x \in \mathbb{R}^n \text{ and } \forall t \in \mathbb{N}. \qquad (2)$$

When ψ_t converges to a constant $c \in (0, +\infty)$, then according to equation (2) the distribution of our data remains the same, that is the mapping $\mathrm{D}_{\overline{1,t}}$ is an identity mapping after some time step in the process.

If we substitute $x = 0$ into the equation (2), then we can get an expression for ψ_t: $\psi_t = \sqrt[n]{f_t(0)/g(0)}$. Let us take $\kappa > 0$ and consider an integral of the form

$$J_t := \int_{B^n(\kappa)} f_t(x)dx = \int_{B^n(\kappa)} \psi_t^n \cdot f_0(\psi_t \cdot x)dx = \int_{B^n(\kappa \cdot \psi_t)} f_0(y)dy,$$

If ψ_t diverges to infinity, then J_t converges to $\|f_0\|_1 = 1$, and if ψ_t converges to zero, then J_t will also converge to zero. In the experiments we measure $\psi_t \cong f_t(0)$ and $J_t \approx \hat{F}_t(\kappa) - \hat{F}_t(-\kappa)$, where $\kappa > 0$ is sufficiently small.

Corollaries of Theorem 1

Corollary 1 (Convergence rate)

For any $q \geq 1$, under conditions of Theorem 1, if $g(x) \in L_q(\mathbb{R}^n)$ and ψ_t converges to zero, it holds that

$$\|f_t(x) - \zeta(x)\|_q \leq (\psi_t^n)^{1-1/q} \cdot \|g\|_q.$$

Corollary 2 (Decreasing moments)

If a system (1) with $n = 1$ satisfies the conditions of Theorem 1 and ψ_t diverges to infinity, then for all $k \in \mathbb{N}$ it holds that

$$\mathbb{E}_{\xi \sim f_t(x)}\left[\xi^{2k}\right] \leq \psi_t^{-2k} \cdot \mathbb{E}_{\xi \sim f_0(x)}\left[\xi^{2k}\right].$$

Results for an Autonomous System

Theorem 2 (Autonomy criterion)

If the evolution operators D_t of a dynamic system (1) have the form (2), then the system is autonomous if and only if

$$\psi_{\tau+\kappa} = \psi_\tau \cdot \psi_\kappa \ \forall \tau, \kappa \in \mathbb{N}. \qquad (3)$$

This criterion is easy to check in practice, since the condition (3) means that the sequence ψ_t is a power sequence, that is $\psi_t = a^t$ for some $a > 0$. An example of a mapping of the form (2) is given in this work with the name Sampling update setup.

Limit to Delta Function or Zero Distribution

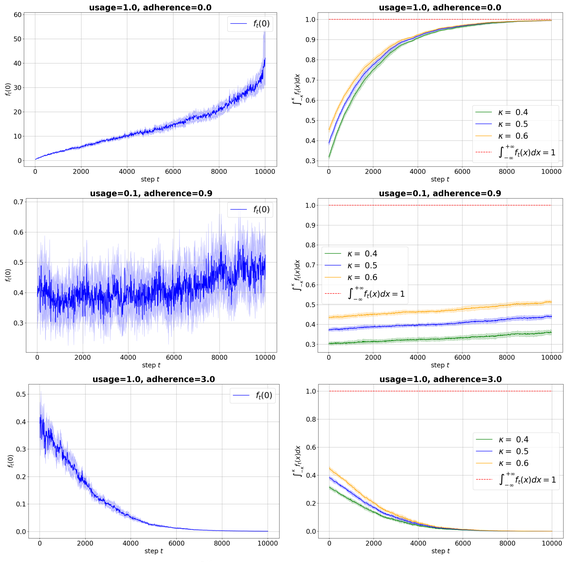


Figure: Counting $f_t(0)$ and $\int_{-\kappa}^{\kappa} f_t(x)dx$ for sampling update setup. We consider such parameters: usage, adherence $= 1, 0$ (first); $0.1, 0.9$ (second); $1, 3$ (third).

As we can see, if usage $p = 1$ and adherence $s = 0$, the limiting probability density of $\mathrm{D}_{\overline{1,t}}(f_0)$, that is the probability density of $\mathbf{y} - h(\mathbf{x})$, is delta function $\delta(x)$.

When usage $p = 0.1$ and adherence $s = 0.9$, the probability density of $\mathbf{y} - h(\mathbf{x})$ remains almost the same, that is $\psi_t \to c \in (0; +\infty)$.

If usage $p = 1$ and adherence $s = 3$ we observe a tendency to the zero distribution $\zeta(x)$.

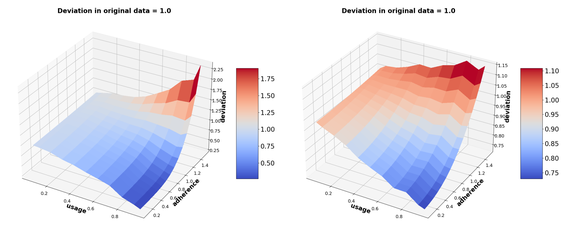


Figure: Change in the standard deviation of the model error for different usage and adherence. Sliding window setup (left), sampling update setup (right).

The graph is almost everywhere either red or blue, hence Theorem 1 is applicable in practice.

Autonomy Check

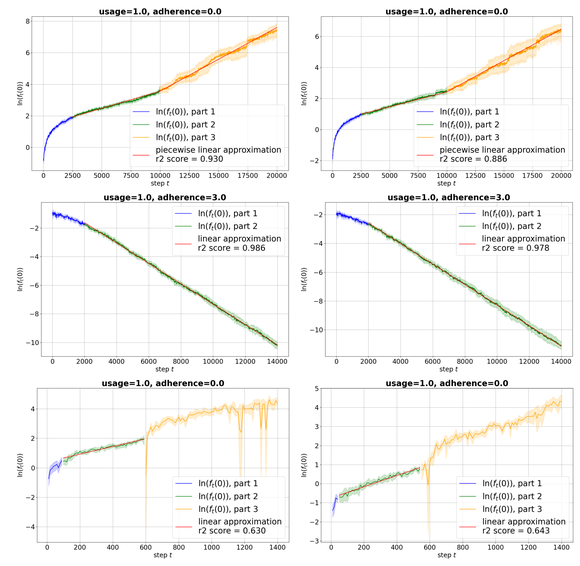


Figure: Testing two designs for autonomy. Sampling update setup (first and second) and sliding window setup (third).

As you can see, in case of the sliding window update we obtain a poor fit on all models and data sets, so the system is not autonomous. The sampling update setup in case of usage $p = 1$ and adherence $s = 3$ is autonomous on all models and data sets, since there is a good fit. In case of usage $p = 1$ and adherence $s = 0$ we observe two linear segments.

Decreasing Moments

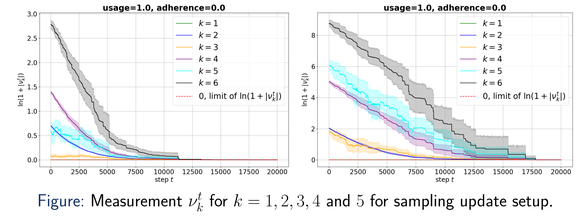


Figure: Measurement ν_k^t for $k = 1, 2, 3, 4$ and 5 for sampling update setup.

As we can see from the measurements, claim of Corollary 2 is satisfied in all observed cases. When usage $p = 1$ and adherence $s = 0$ the limit of mappings $\mathrm{D}_{\overline{1,t}}(f_0)$, and correspondingly of $\mathbf{y} - h(\mathbf{x})$, is the delta function $\delta(x)$.

DBP-Finder: Enhanced Identification of DNA-Binding Proteins Using Fine-Tuned Protein Language Models

Alexander Gavrilenko[1], Elizaveta Shaburova[1], Denis Antonets[1], Yury Vyatkin[1], Vasily Ramensky[123]

1 Institute for Artificial Intelligence, Lomonosov Moscow State University, Moscow, Russia; 2 National Medical Research Center for Therapy and Preventive Medicine of the Ministry of Healthcare of Russian Federation, Moscow, Russia; 3 Department of Bioengineering and Bioinformatics, Lomonosov Moscow State University, Moscow, Russia

Objective

This study aims to identify DNA-binding proteins (DBPs) using transfer learning on pretrained protein language models (PLMs) (http://dx.doi.org/10.1101/2024.02.05.578959). By leveraging PLMs for informative sequence representations, we will develop a predictive algorithm for DBP identification, addressing the need for scalable, automated prediction methods amidst limited experimental annotations and increasing genomic data.

Methods

Training set construction

We sourced high-quality, manually annotated, non-redundant protein sequences from UniProtKB/Swiss-Prot. DBPs were selected using the DNA-binding GO term, while Non-DBPs excluded nucleic acid-binding GO terms per QuickGO (https://doi.org/10.1093/bioinformatics/btp536).
Sequences shorter than 50 or longer than 1,024 amino acids and those with undefined amino acids “X” were excluded. The training set balanced 34,936 DBPs with 34,936 Non-DBPs.

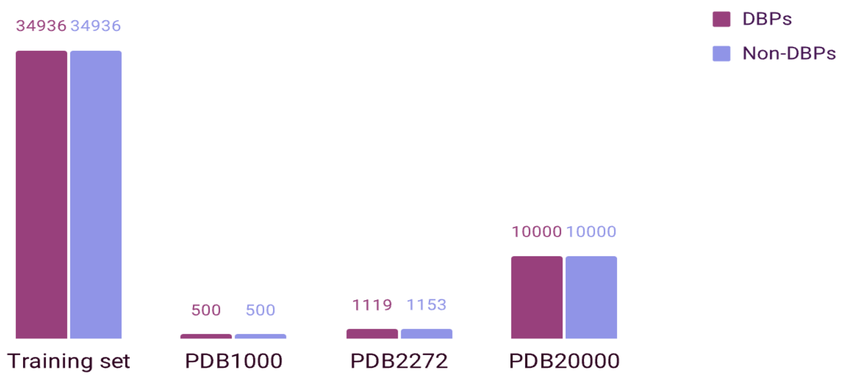
Testing sets

Three test datasets were used as benchmarks to facilitate direct comparison with previous studies:

- **PDB2272** dataset from Du, Diao, Liu, & Li (2019) (https://doi.org/10.1021/acs.jproteome.9b00226)
- **PDB20000** and **PDB1000** datasets from Ma (2019) (http://dx.doi.org/10.17504/protocols.io.2rdgd26)

Protein sequences in these datasets originated from UniProtKB and had GO annotations assigned by UniProt.

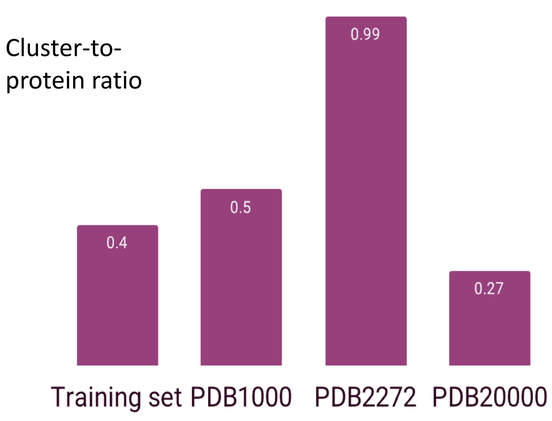
Counts of DBPs and Non-DBPs in training and testing sets

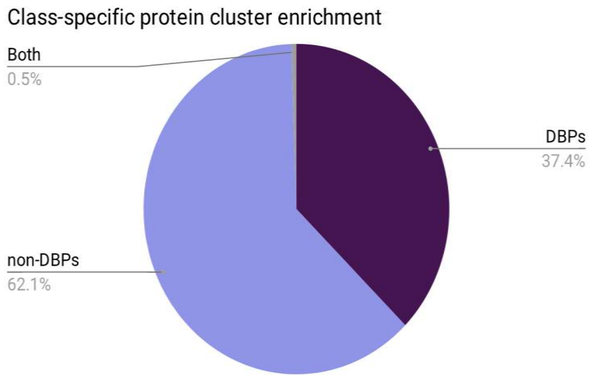


Clustering

We used MMseqs2 (https://doi.org/10.1038/nbt.3988) with a 50% sequence identity threshold to filter out homologous sequences, preventing data leakage and ensuring fair model evaluation. We also calculated the cluster-to-protein ratio and counted DBPs and Non-DBPs within each cluster.

Cluster-to-Protein Ratio: Analyzing Protein Clustering Distribution





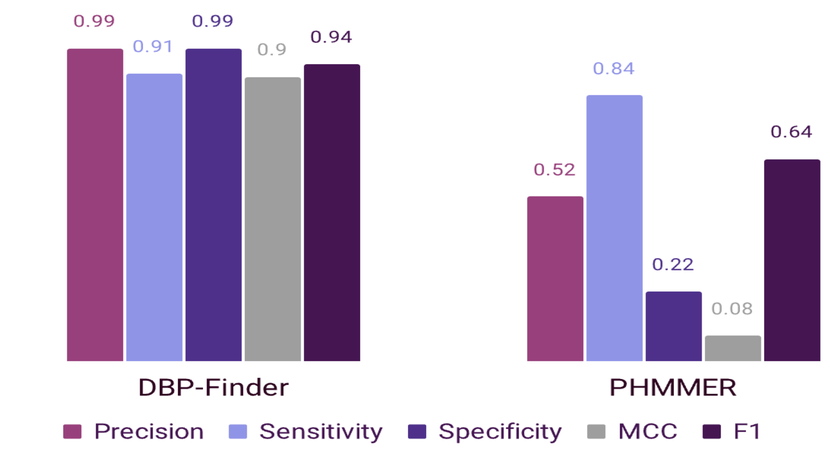
Clusters are predominantly enriched with either DBPs or Non-DBPs, indicating good data quality and supporting the idea that protein amino acid sequences determine DNA-binding function.

Results

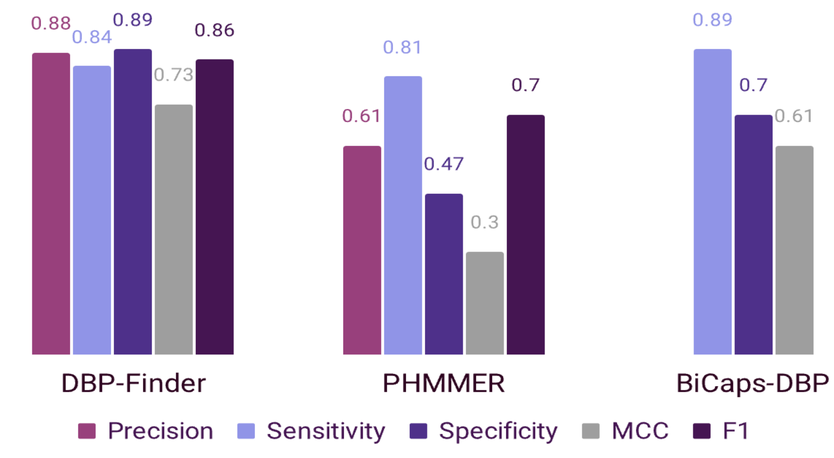
We trained an Ankh model (http://dx.doi.org/10.1101/2023.01.16.524265) with 13 million parameters, using a transformer backbone for protein sequence representation and a classification head. Training used the AdamW optimizer, batch size of 64, and an initial learning rate of 2e-4 over 9 epochs, retaining the model with the lowest validation loss. Accuracy and stability were enhanced by training an ensemble of five models.

Performance comparison

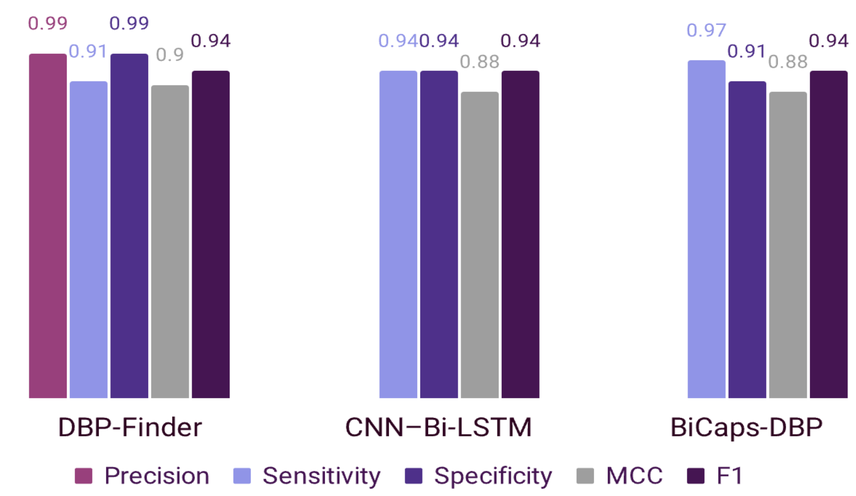
PDB1000 Testing Set:
PHMMER (https://doi.org/10.6019/tol.hmmer-w.2018.00001.1): HMM-based method for sequence similarity searches.



PDB2272 Testing Set:
BiCaps-DBP (https://doi.org/10.1016/j.compbiomed.2023.107241): a method with a three-layer architecture: encoding layer for one-hot encoding, Bi-LSTM layer for contextual features, and 1D-CapsNet layer for feature correlation and classification.



PDB20000 Testing Set:
CNN–Bi-LSTM (https://doi.org/10.1371/journal.pone.0225317): uses one-hot encodings, includes layers for amino acid numbers, continuous vectors, convolutions with max pooling, and Bi-LSTM for contextual features.



This work is an output of a research project implemented as part of the Research Program at the MSU Institute for Artificial Intelligence.

Tensor bandits and their applications

Horbach Maryna

HSE University, Moscow, Russia

Main contribution

- New algorithm based on tensor train decomposition was developed. It combines idea of low rank approximation and effective tensor operations. The comparative research of existing algorithms was made and TensorTrain algorithm showed competitive results.
- Approach for usage of tensor bandits in context case was studied. Context versions of algorithms were developed and tested.

Introduction

Multi-armed bandit algorithms — these are algorithms where a decision maker iteratively selects one of multiple fixed choices, with each option represented by an arm. After making a choice a reward is received, and the agent's goal is to maximize the reward sum.

Given: B — a set of n choices, T — the time period over which we maximize the reward.

- $Rd = \sum_{t=1}^{T} r_t$ – Total reward, where r_t is sampled by the environment from the reward distribution of the arm chosen at step t, which is unknown to the agent.
- $Rt = T \cdot \mu^* - Rd$ – Total regret, where $\mu^* = \max_{\text{arm} \in \text{all arms}} \mathbb{E}(\mu_{\text{arm}})$, and μ_{arm} is a random variable corresponding to the reward of an arm.

Objective: Design an algorithm that maximizes the reward (minimizes regret).

Key Algorithms:

- **Vectorized UCB (Upper Confidence Bound)**: Selects arms based on optimistic reward estimates.
- **Epoch Greedy**: Selects the optimal arm at each step by choosing the maximum reward.
- **Tensor Elimination**: Removing arms based on whether their reward estimates fall outside a specified confidence interval.

TensorTrain Algorithm

input: dimensions $\in \mathbb{N}^n$ - dimension of the reward tensor, ranks $\in \mathbb{N}^n$ - dimension of the tensor train decomposition, T - number of steps, T_e - number of exploration steps
For t in range(T_e)
    Choose random arm
    Update average reward tensor
Tensor completion → estimated reward tensor **R**
Tensor train decomposition of **R**
For t in range(T_e, T)
    Finding maximum of **R** (`optima_tt_max`)
    Choose optimal arm
    Update **R**
    if t % update-each == 0:
        Restore **R** from updated decomposition
        Tensor completion of **R**
        Tensor Train decomposition of **R**

Algorithm description

1. The `optima_tt_max` algorithm represents tensor elements as a distribution, selecting the k most probable at each step to find the optimal solution.
2. Updating **R** without restoring from tensor train:

Formula:

Let $A, B \in \mathbb{R}^{p_1 \times \cdots \times p_n}$ be two tensors, with tensor train cores $C_{A,i} \in \mathbb{R}^{r_{A,(i-1)} \times p_i \times r_{A,i}}$ $C_{B,i} \in \mathbb{R}^{r_{B,(i-1)} \times p_i \times r_{B,i}}$, then cores $C_{D,0} \in \mathbb{R}^{1 \times p_1 \times (r_{A,1} + r_{B,1})}$, $C_{D,i} \in \mathbb{R}^{(r_{A,(i-1)} + r_{B,(i-1)}) \times p_i \times (r_{A,i} + r_{B,i})}$, $C_{D,n} \in \mathbb{R}^{(r_{A,(n-1)} + r_{B,(n-1)}) \times p_n \times 1}$ for $D = A + B$ can be calculated as

$$C_{D,0} = [A_{A,0} \quad C_{B,0}],$$
$$C_{D,i} = \begin{bmatrix} C_{A,i} & 0 \\ 0 & C_{B,i} \end{bmatrix} \quad i = 1, \ldots, n-1,$$
$$C_{D,n} = [C_{A,n} \quad C_{B,n}].$$

Numerical Results

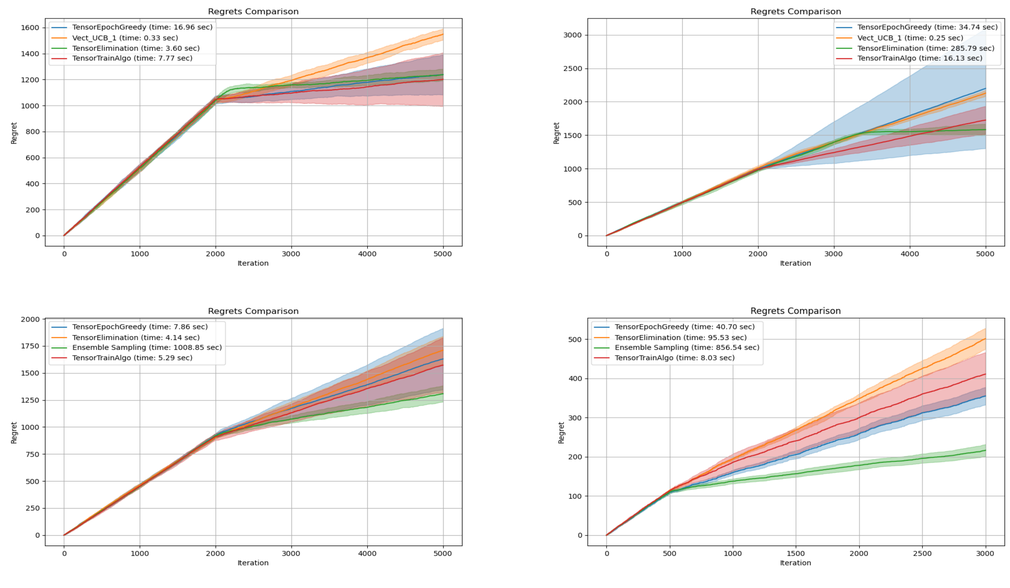


Figure 1:
First row: Five runs were conducted on reward tensors of sizes $10 \times 10 \times 10$ and $5 \times 5 \times 5$ generated from a normal distribution.
Third row 1: Contextual algorithms were run five times on reward tensors of size $5 \times 5 \times 5$, where the first dimension corresponded to the context.
Third row 2: Contextual algorithms were run five times on the SyntheticBanditDataset simulator (Open Bandit dataset), where the context dimension was 3 and the number of arms was 5.

Contextual bandits

To create contextual version of tensor bandits algorithm additional dimensions need to be added to the reward tensor to encode context. The algorithm can be outlined as follows:

- Obtain context from the environment.
- Select the part of the reward tensor corresponding to the given context.
- Choose the optimal arm using the tensor bandits algorithm.

The idea of the Ensemble Sampling algorithm is to initially set a prior distribution, which is then updated through the algorithm's actions. It builds on Thompson Sampling and utilizes Tucker decomposition

Concluding remarks

- This work investigated, implemented, and tested various existing algorithms for solving the tensor multi-armed bandit problem, leading to the development of a new algorithm called TensorTrain, which utilizes low-rank decomposition. Results showed that while TensorTrain is one of the fastest algorithms, Ensemble Sampling outperforms it on contextual task.
- One of future research directions is to explore how changes in the environment during algorithm operations affect their effectiveness, as this frequently occurs in real-world scenarios.

Contact Information

- Implementations of algorithms are available athttps://github.com/horbachmp/TensorBandits

Diffractive neural networks for image processing

Viktor V. Krasnikov Egor Y. Belashov Andrey A. Grunin Andrey A. Fedyanin

Faculty of Physics of M.V.Lomonosov Moscow State University, GSP-1, Leninskie gory, Moscow 119991 Russian Federation

Abstract

Digital technologies are undergoing rapid transformation due to advancements in artificial intelligence [2], particularly deep neural networks, which are widely used in areas such as image recognition and big data processing. However, this growth increases the demand for computational resources. One promising approach to optimizing machine learning algorithms is the use of optical technologies, which can accelerate data processing due to the properties of light. This work explores the potential of optical image preprocessing in the context of spatially incoherent light, focusing on developing a model for light propagation through heterogeneous structures and applying machine learning algorithms to optimize parameters. A hybrid optoelectronic neural network was built and trained for image classification based on this model.

Motivation

Delegating computational tasks to specialized units, such as GPUs and optical processors, enables efficient processing of specific tasks with high performance and low energy consumption. Optical devices, leveraging the properties of light, provide high parallelism and low latency, making them promising for solving tasks that require intensive computations.

Research objectives

Theory

A diffractive neural network represents an array of optically heterogeneous thin masks. On each of these masks, the phase and/or amplitude of the field is modulated point by point. The system is divided into an initial plane, where the image is projected, the planes after the masks, and the output plane. This system is associated with a perceptron-type neural network because the field at each point of a plane is linearly related to the fields at points in the previous plane. An important distinction is that the number of transition weights between layers is much smaller [3]. This is due to the fact that the field at any point in the plane after the mask is calculated as an integral of the multiplication of the field in the previous plane with a fixed kernel, followed by multiplying this value by a controlled coefficient. The mathematical description is presented in Equation below:

$$u_i(x,y) = w_i(x,y) \iint_{-\infty}^{+\infty} u_{i-1}(\xi,\eta) \frac{e^{ik\sqrt{L^2+(x-\xi)^2+(y-\eta)^2}}}{\sqrt{L^2+(x-\xi)^2+(y-\eta)^2}} d\xi\, d\eta \tag{1}$$

In this equation, u_i, u_{i-1} represent the field in the current and previous layers, respectively, L is the distance between the layers, and $w_i(x,y)$ is the weight of the neuron at the point (x,y). The number of neurons and the discretization of this equation depend on the degree of accuracy with which optical inhomogeneities can be created. Thus, for a transition between layers with N neurons, a perceptron has N^2 weights, whereas a diffractive neural network has N.

Method of random phase generation of the initial field: A random Fourier pattern is generated, where the value at each point is equally likely to range from 0 to 1, and is constrained by a Gaussian function with a given dispersion $\frac{1}{\sigma^2}$.

Calculation of coherent and incoherent propagation: Free-space wave propagation methods aim at solving the homogeneous Helmholtz equation [1]:

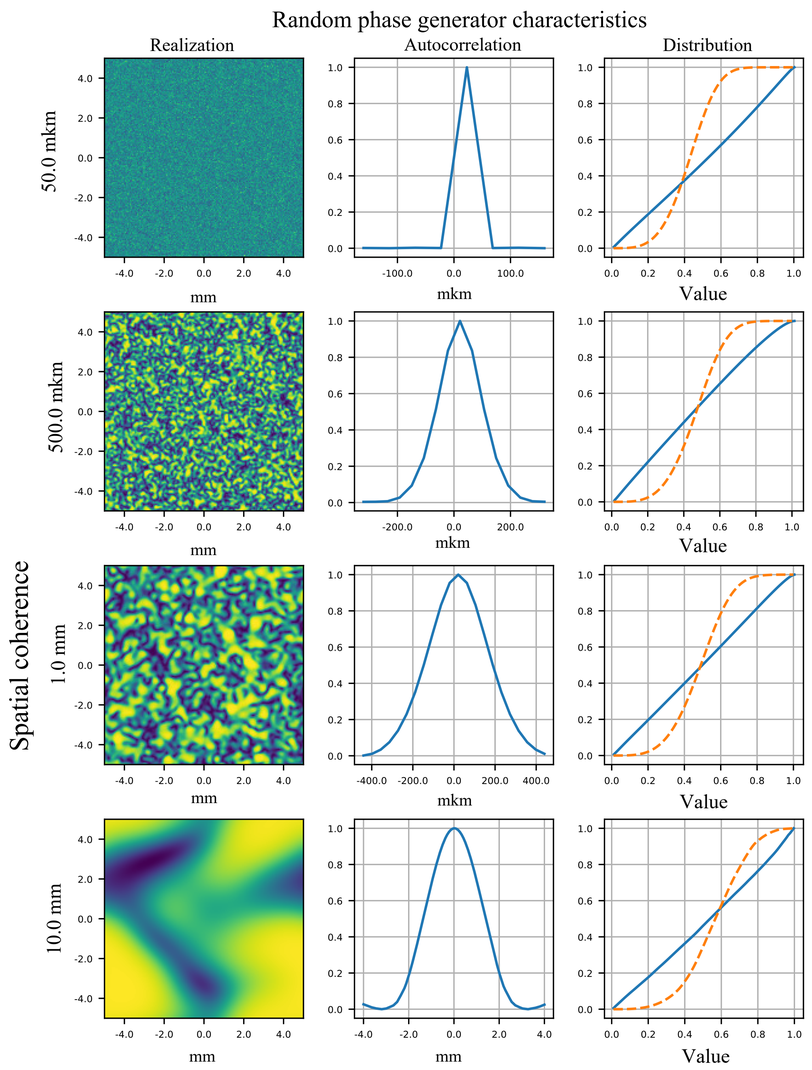
$$\nabla\psi + k^2\psi = 0 \tag{2}$$

Mathematically, AS propagation can be formulated as:

$$\psi = \mathcal{F}^{-1}[H_{AS}\mathcal{F}[\psi_0]] \tag{3}$$

Transfer function:

$$H_{AS}(f_x, f_y) = e^{i2\pi z\sqrt{\frac{1}{\lambda^2}-f_x^2-f_y^2}} \tag{4}$$



Results and discussion

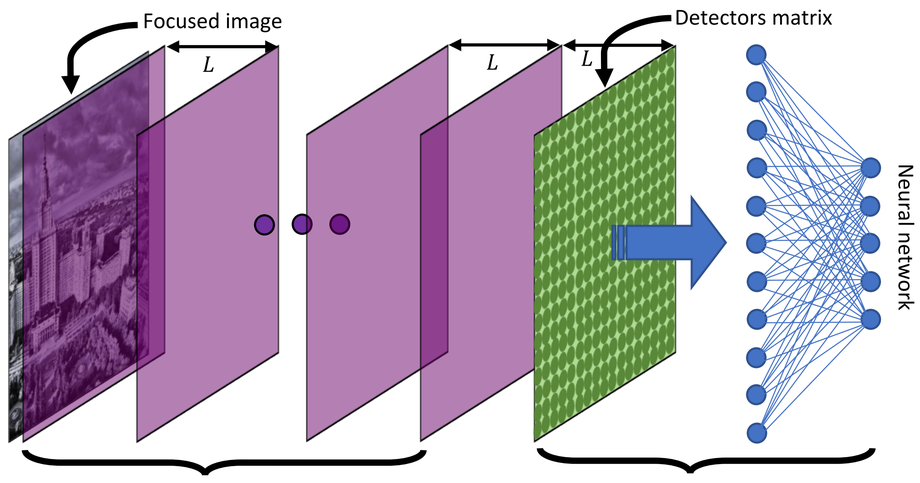


Figure 1. Schematic diagram of optoelectronic neural network.

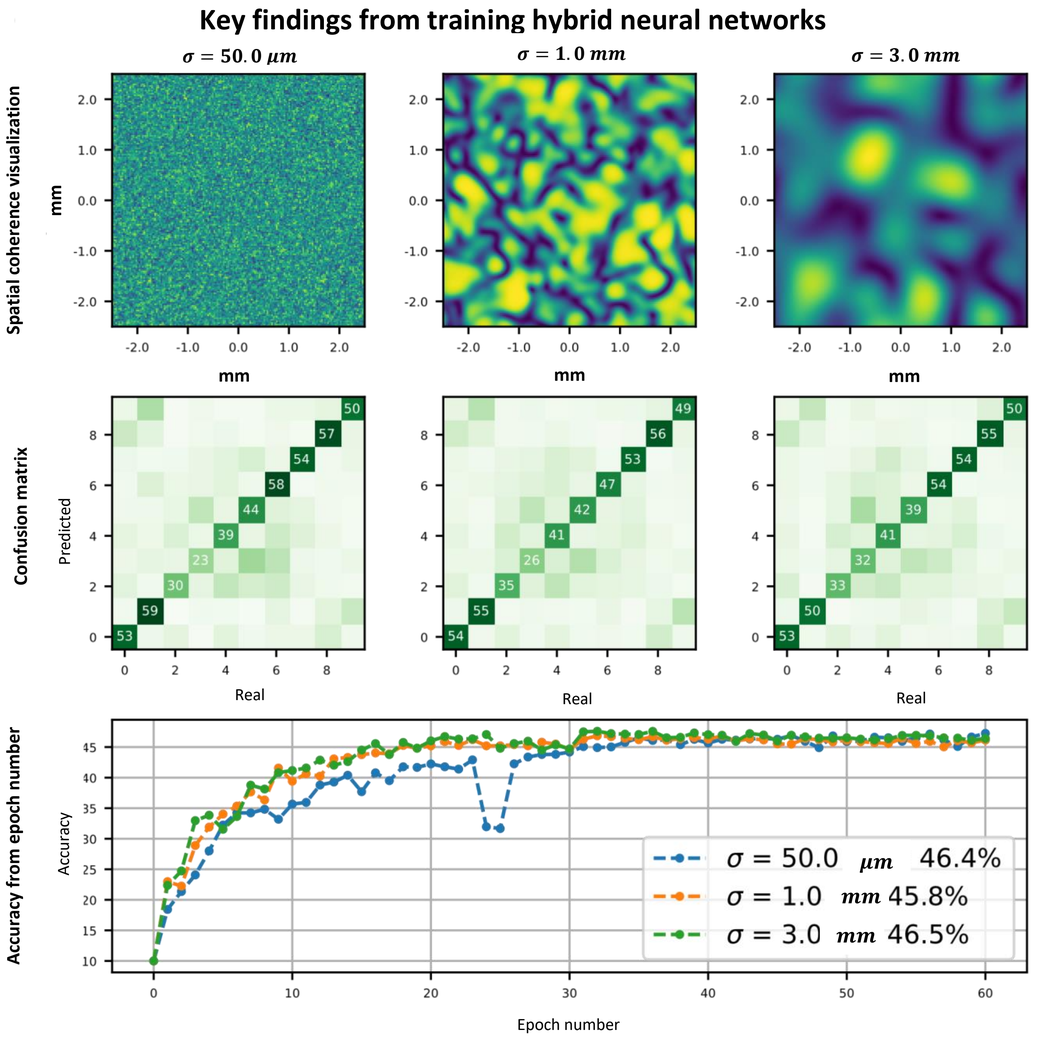


Figure 2. Training results of hybrid neural networks. The first row - visualisations of a particular realisation of the incoherent field phase. Second row - error matrix: colour represents the percentage of answers, numbers on the diagonal represent the percentage of correct answers. The third row is a graph of the dependence of accuracy on the epoch number for three models.

Conclusions

By training optoelectronic networks for three levels of spatial coherence ($50\mu m, 1000\mu m, 3000\mu m$), it is shown that the models give the same results for a large number of epochs, regardless of the degree of coherence. The small accuracy for some classes is explained by the similarity of the images. Phase and amplitude modulation patterns confirm that optical layer training did occur. Examples of the networks show that the system performs image transformations rather than simply focusing the image on the detectors.

References

[1] Rainer Heintzmann, Lars Loetgering, and Felix Wechsler.
Scalable angular spectrum propagation.
Optica, 10(11):1407–1416, 2023.

[2] Jiamin Wu, Xing Lin, Yuchen Guo, Junwei Liu, Lu Fang, Shuming Jiao, and Qionghai Dai.
Analog optical computing for artificial intelligence.
Engineering, 10:133–145, 2022.

[3] Tao Yan, Jiamin Wu, Tiankuang Zhou, Hao Xie, Feng Xu, Jingtao Fan, Lu Fang, Xing Lin, and Qionghai Dai.
Fourier-space diffractive deep neural network.
Physical review letters, 123(2):023901, 2019.

Website

krasnikovvv@my.msu.ru

Primal-Dual Gradient Methods for Searching Network Equilibria in Combined Models with Nested Choice Structure and Capacity Constraints

Meruza Kubentayeva [1] Demyan Yarmoshik [1, 2] Michael Persiianov [1, 2] Alexey Kroshnin [2, 3] Ekaterina Kotliarova [1] Nazarii Tupitsa [1]
Dmitry Pasechnyuk [1] Alexander Gasnikov [1, 2, 3] Vladimir Shvetsov [1, 4] Leonid Baryshev [4] Aleksei Shurupov [4]

[1]Moscow Institute of Physics and Technology [2]Institute for Information Transmission Problems RAS [3]Higher School of Economics [4]Russian University of Transport

Overview

We consider the problem of forecasting travel demand in a transportation system. This problem arises in
- the management of infrastructure development
- the land use planning
- the policymaking for maintaining sustainable transportation systems.

1. Traditional Four-Step Procedure

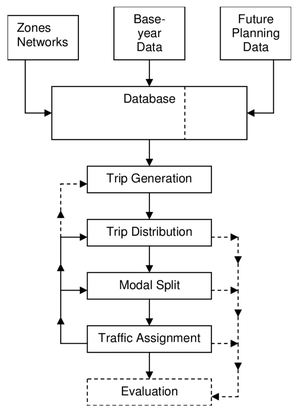


Figure: The classic four-stage transport model

Trip Distribution

- L_i number of departures from zone $i \in O$
- W_j number of arrivals to zone $j \in D$
- T_{ij} generalized cost for travelling from zone i to j
- d_{ij} number of trips from zone i to j
- γ calibration parameter

Entropy Maximizing Model

$$-\sum_{i,j} d_{ij}T_{ij} - \underbrace{\frac{1}{\gamma}\sum_{i,j} d_{ij}\ln d_{ij}}_{H(d)} \longrightarrow \max_{\substack{d_{ij}:\ d_{ij}\geq 0,\\ \sum_j d_{ij}=L_i,\\ \sum_i d_{ij}=W_j}}$$

Modal Split

- d_{ij}^{am} number of trips from zone i to j by mode m and agent a
- T_{ij}^{m} generalized cost of travelling from i to j by mode m
- $M(a)$ set of modes available to type a agents
- α_{am}, β_{am} calibration parameters

The Multinomial Logit Model (MNL)

$$\frac{d_{ij}^{am}}{d_{ij}^{a}} = \frac{\exp(-\alpha_{am}T_{ij}^{m} + \beta_{am})}{\sum_{m\in M(a)}\exp(-\alpha_{am}T_{ij}^{m} + \beta_{am})}$$

Traffic assignment

Beckmann model

x^* is an equilibrium state if and only if x^* is a minimum of the potential function:

$$\Psi(f(x)) = \sum_{e\in E}\underbrace{\int_0^{f_e}\tau_e(z)dz} \to \min_{f=\Theta x:\ x\in X},$$

where $\bar{t}_e$ – free flow time, $\bar{f}_e$[veh/hour] – link capacity. Dual problem:

$$Q(t) = \sum_{ij\in OD} d_{ij}T_{ij}(t) - \underbrace{\sum_{e\in\mathcal{E}}\sigma_e^*(t_e)}_{h(t)} \longrightarrow \max_{t\geq\bar{t}}, \quad (1)$$

where $\sigma_e^*(t_e)$ is the Fenchel conjugate function of $\sigma_e(f_e)$, $e \in E$.

Stable dynamics model [3]

$$\tau_e(f_e) = \begin{cases} \bar{t}_e, & 0 \leq f_e < \bar{f}_e, \\ [\bar{t}_e, \infty], & f_e = \bar{f}_e, \\ +\infty, & f_e > \bar{f}_e. \end{cases}$$

The pair (f^*, t^*) is an equilibrium if and only if it is a solution of the saddle-point problem

$$S(f(x), t) = \langle t, f\rangle - \underbrace{\langle t - \bar{t}, \bar{f}\rangle}_{h(t)} \longrightarrow \min_{\substack{f=\Theta x:\\ x\in X}}\max_{t\geq\bar{t}},$$

2. Refined: Network equilibrium model (NE)

According to [1], the combined distributionmodal splitassignment problem can be formulated as follows:

$$P_3(f,d) = \Psi(f) + H(d) \to \min_{\substack{f=\Theta x,\ x\in X(d)\\ d\in\Pi'(l,w)}}, \quad \text{(P3)}$$

where

$$H(d) = \sum_{i,j,r,a}\frac{1}{\gamma_r}d_{ij}^{ra}\ln d_{ij}^{ra} + \sum_{i,j,r,a,m}\frac{1}{\alpha_a}d_{ij}^{ram}\left(\ln\left(\frac{d_{ij}^{ram}}{d_{ij}^{ra}}\right) + \beta_{am}\right).$$

The saddle-point problem:

$$S_3(d,t) = \underbrace{\sum_{i,j,r,a} d_{ij}^{ra}T_{ij}^{a}(t) + \sum_{i,j,r,a}\frac{1}{\gamma_r}d_{ij}^{ra}\ln d_{ij}^{ra}}_{E(d,T(t))} - h(t) \to \min_{d\in\Pi(l,w)}\max_{t\geq\bar{t}}, \quad \text{(S3)}$$

where $T_{ij}^{a}(t) = -\frac{1}{\alpha_a}\ln\left(\sum_m \exp\left(-\alpha_a T_{ij}^{m}(t) - \beta_{am}\right)\right)$, $T_{ij}^{m}(t)$ is the minimal cost of the path from $i \in O$ to $j \in D$ with the links cost $t_e + c_e^m$.
The dual problem is

$$D_3(t) = \underbrace{\min_{d\in\Pi(l,w)} E(d,T(t))}_{-\Phi(t)} - h(t) \longrightarrow \max_{t\geq\bar{t}}, \quad \text{(D3)}$$

3. Evans Algorithm

1. For each edge $e \in E$ calculate the costs τ_e that correspond to the flows f_e^k.
2. For each origin vertex $i \in O$ find the minimum T_{ij} of travelling to each destination $j \in D$ and choose a minimum cost route from i to j.
3. Find a new set of trip distributions q_{ij}, given the new T_{ij} costs.
4. Assign the new demands q_{ij} to the minimum cost routes chosen in Step 2 to obtain a new flow vector $y \in \mathbb{R}^{|E|}$
5. Find the linear combination $(1-\lambda)(f^k, d^k) + \lambda(y, q)$, $0 \leq \lambda \leq 1$, of (f^k, d^k) and (y, q) that minimizes the objective function P_3 and

4. Dual method for NE problem

Here we used the following notations:

$$\phi_0(t) = \frac{1}{2}\left\|t - t^0\right\|_2^2,$$

$$\phi_{k+1}(t) = \phi_k(t) + \alpha_{k+1}\left[\tilde{\Phi}(y^{k+1}) + \left\langle\tilde{\nabla}\Phi(y^{k+1}), t - y^{k+1}\right\rangle + h(t)\right].$$

Note that we did not specify the stopping criterion as it can be different for different models

Algorithm Universal Method of Similar Triangles

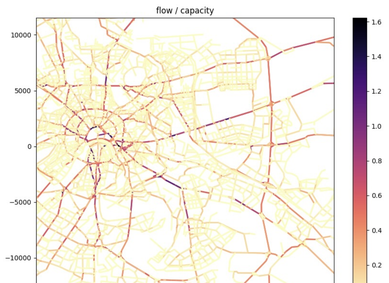
Require: $L_0 > 0$, starting point t^0, accuracy $\varepsilon > 0$
$u^0 := t^0$, $A_0 := 0$, $k := 0$
repeat
  $L_{k+1} := L_k/2$
  while true do
    $\alpha_{k+1} := \frac{1}{2L_{k+1}} + \sqrt{\frac{1}{4L_{k+1}^2} + \frac{A_k}{L_{k+1}}}$, $A_{k+1} := A_k + \alpha_{k+1}$
    $y^{k+1} := \frac{\alpha_{k+1}u^k + A_k t^k}{A_{k+1}}$
    $u^{k+1} := \arg\min_{t\in\operatorname{dom} h}\phi_{k+1}(t)$
    $t^{k+1} := \frac{\alpha_{k+1}u^{k+1} + A_k t^k}{A_{k+1}}$
    if $\tilde{\Phi}(t^{k+1}) \leq \tilde{\Phi}(y^{k+1}) + \left\langle\tilde{\nabla}\Phi(y^{k+1}), t^{k+1} - y^{k+1}\right\rangle + \frac{L_{k+1}}{2}\left\|t^{k+1} - y^{k+1}\right\|_2^2 + \frac{\alpha_{k+1}}{2A_{k+1}}\varepsilon$ **then**
    else
      $L_{k+1} := 2L_{k+1}$
    end if
  end while
  $k := k + 1$
until Stopping criterion is fulfilled

Main Contributions

- We propose a way to **solve** the dual problem of **the nested combined model** of [1] with a universal accelerated gradient method USTM [2];
- We **extend the nested combined model** to the case of capacitated networks: namely, we propose NE with the stable dynamics [3] traffic assignment model;
- We provide **theoretical upper bounds** on the complexity of searching network equilibrium by the USTM algorithm.

publication ref. :

5. Numerical Experiments



- road network: 63073 nodes 94546 arcs (roads, permitted turns at the intersections)
- 1420 transportation zones
- trip purposes: home-work, home-other
- users types: car-owners, non-car-owners
- travel modes: by foot, car, and public transport

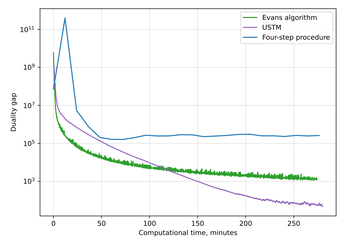


Figure: Duality gap convergence

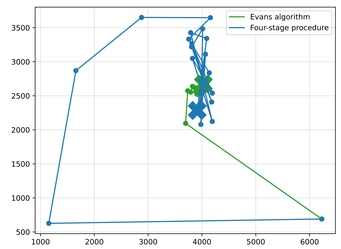


Figure: 2-Dimensional projections of d_{ij}^m trajectories for the Evans algorithm and the Four-stage procedure, obtained by multidimensional scaling. The trajectory of the Evans method is sparsified to 50 points. The last point is marked with a large cross

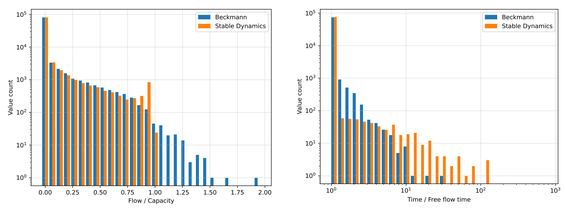


Figure: Histograms of the network load: a) histogram of the ratio of the amount of flow on the link to its capacity, b) histogram of the ratio of the travel time on the link to the travel time on the same link when it is free

8. References

[1] Torgil Abrahamsson and Lars Lundqvist. "Formulation and estimation of combined network equilibrium models with applications to Stockholm". In: *Transportation Science* 33.1 (1999), pp. 80–100.

[2] Alexander Vladimirovich Gasnikov and Yu E Nesterov. "Universal method for stochastic composite optimization problems". In: *Computational Mathematics and Mathematical Physics* 58.1 (2018), pp. 48–64.

[3] Yurii Nesterov and Andre De Palma. "Stationary dynamic solutions in congested transportation networks: summary and perspectives". In: *Networks and spatial economics* 3 (2003), pp. 371–395.

ASCOMP-2024
Fall 2024

NeRF with semantics for reconstruction of 3D scene and object segmentation by text prompts

Julia Kudryavtseva[1]

Problem Statement

Develop a system capable of matching the points in a 3D scene with their respective visual-semantic meaning. Given a text prompt the system is capable of segmenting objects in 3D space and extracting their coordinates in the real world.

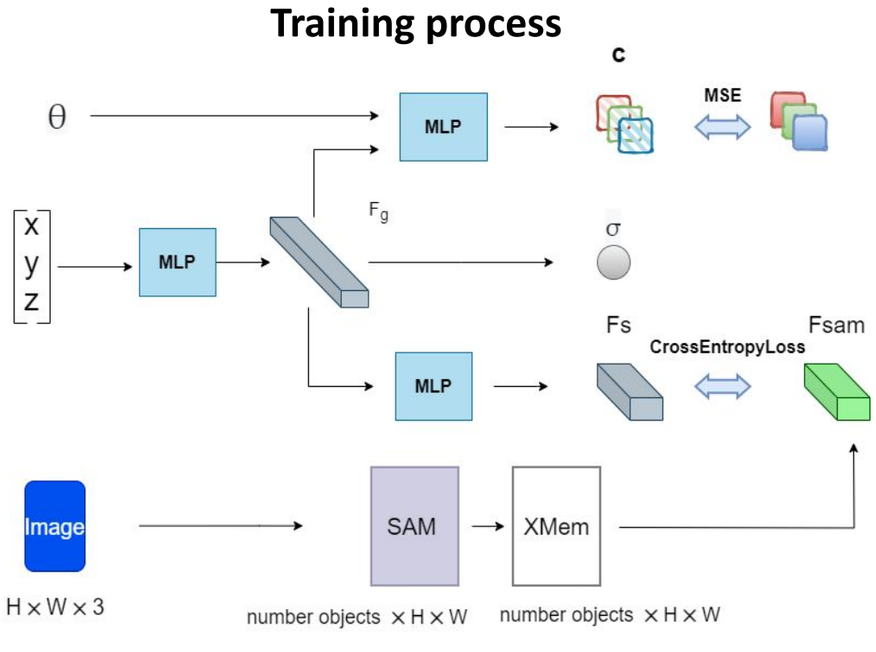
Motivation

To bring an object at user's request, a mobile robot has to find a queried item and determine its coordinates in a 3D space. The robot has to understand the geometry of an object to interact with it effectively.

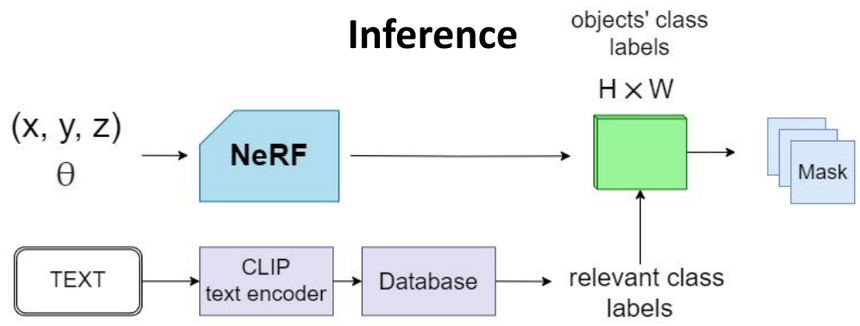
Research structure:

- **Preprocessing:** SAM model generates high-quality masks selected by an NMS-like algorithm. The VOS model (XMem) makes them consistent: receives object mask as input and tracks it on the video. To capture all occurrences on the video, XMem is sequentially restarted for each frame. The mask detected earlier is taken.
- **Image-language embeddings database:** For every class label it's consistent masks are multiplied by the corresponding image frames to eliminate the background. Then they are encoded using the CLIP model, which are averaged over the dataset to capture an object from multi-views.
- **Training:** Hash-NeRF predict labels of semantic class and RGB color for every pixel taking as input 3d coordinates and view direction.

Training process



Inference



Model	Use GPU for inference	Time inference	Memory
NeRF + SAM 2D	Yes	-	-
LERF	Yes	-	-
CLIP-Fields	No	*O*(number of points × CLIP dim)	*O*(number of points × CLIP dim)
OpenScene	No	*O*(number of points × CLIP dim)	*O*(number of points × CLIP dim)
Hash-NeRF (ours)	**No**	***O*(number of classes × CLIP dim)**	***O*(number of points + CLIP dim)***

Tab 1. **IoU metrics** of segmentation masks generated with text prompts on test set of LERF dataset.

text prompt	NeRF+SAM2D	LERF	Hash-NeRF
nerf gun	0,524	0,626	**0,862**
typewriter	0,439	0,640	**0,807**
white cabinet	**0,800**	0,265	0,389
yellow bulldozer	0,423	0,819	**0,878**
scene: `dozer_nerfgun_waldo`	0,546	0,588	**0,734**
apple	**0,930**	0,595	0,879
bear	0,778	0,480	**0.911**
mug	**0,871**	0,531	0,700
plate	**0,985**	0,688	0,934
scene: `teatime`	**0,891**	0,573	0.856
knives	0,599	0,238	**0,698**
refrigerator	0,683	0,170	**0,746**
sink	**0,910**	0,103	0,779
`mIoU scene: waldo_kitchen`	0,731	0,170	**0,741**

Tab 2. **Accuracy of object** localization by text prompts on test set of LERF dataset scenes. The object is localized successfully if its IoU metrics > 0,5 .

text prompt	NeRF+SAM2D	LERF	Hash-NeRF
nerf gun	0,541	0,514	**0,919**
typewriter	0,486	0,730	**0,865**
white cabinet	**0,838**	0,189	0,324
yellow bulldozer	0,459	0,892	**0,973**
scene: `dozer_nerfgun_waldo`	0,581	0,588	**0,770**
apple	**0,944**	0,611	**0,944**
bear	0,833	0,500	**0,944**
mug	**0,944**	0,889	**0,944**
plate	**0.999**	**0.999**	**0.999**
scene: `teatime`	0,930	0,750	**0,958**
knives	0,526	0,158	**0,789**
refrigerator	0,684	0,1	**0,789**
sink	**0,947**	0,105	0,737
`mAcc scene: waldo_kitchen`	0,719	0.121	**0,772**

Results:

- outperforms baselines on both segmentation quality and consistency on all scenes in average.
- addresses both general and specific language concepts with significant quality improvements.
- could overcome ambiguous queries with user guidance.

Fig. 1: Segmented point cloud produced by Hash-NeRF for "teatime" scene LeRF dataset.

Conclusion

Hash-NeRF could localize semantic information into robot memory effectively and interact with a user by text queries. It characterizes with: high-quality of segmentation masks, open-vocabulary, object, localization on occluded areas, superiority in time inference and memory-saving.

1 Skolkovo Institute of Science and Technology, Skolkovo, Moscow Region, Russia
Group contacts: julia.kudr.alex@gmail.com

Interior-point methods for mathematical optimization

@Kulakov_Nikita
akulakov11@gmail.com

RTU MIREA student / B1 Group intern

Moscow, Russia

INTRODUCTION

- The area of IPM has been one of the liveliest in mathematical programming in the last two decades. These techniques were primarily in the form of barrier methods, widely used during the 1960s for problems with nonlinear constraints, their use for the fundamental problem of linear programming was unthinkable because of the total dominance of the simplex method. During the 1970s, barrier methods were superseded, nearly to the point of oblivion, by newly emerging and seemingly more efficient alternatives such as augmented Lagrangian and sequential quadratic programming methods. By the early 1980s, barrier methods were almost universally regarded as a closed chapter in the history of optimization
- In 1984 Narendra Karmarkar announced a fast polynomial-time interior method for linear programming; in 1985, a formal connection was established between his method and classical barrier methods. Since then, interior methods have continued to transform both the theory and practice of constrained optimization.

Semi-definite and conic programming

- Semidefinite programming may be viewed as a generalization of linear programming, where the variables are $n \times n$ symmetric matricies, denoted by X, rather than n-vectors. In SDP we wish to minimize an affine function of symmetric matrix X subject to linear constraints and semidefinite constraints, the latter requiring that "X must be positive semidefinite". This typically written as $X \succeq 0$ that resembles inequality constraints in continuous optimization.
- Many extra complications arise in SDP. For example, the feasible region defined by constraints is not polyhedral, so there is no analogue of the simplex method.
- Nesterov and Nemirovski showed that function $\log\det x$ is self-concordant for SDP, which means that SDP can be solved in polynomial time via sequence of barrier subproblems parametrized by μ
- Conic programming is a subclass of convex optimization that deals with optimization problems where the feasible region is defined by a convex cone. It generalizes linear programming and encompasses several important types of problems, including quadratic programming and semidefinite programming.

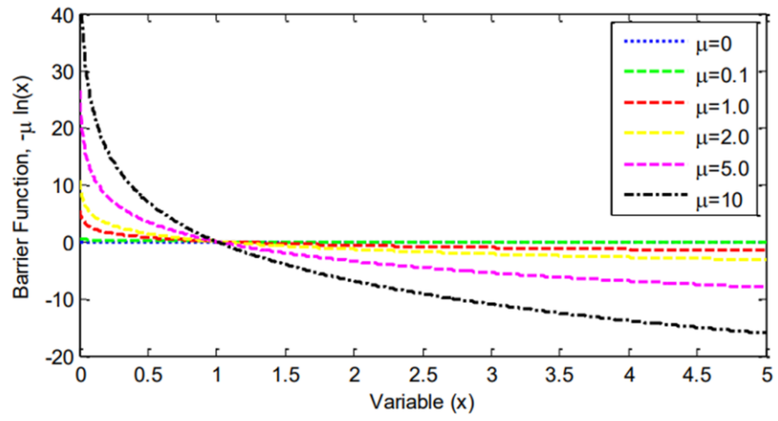
Early development and key Inspirations

- Karmarkar's method main features:
- Faster than any other for large scale programs
- Polynomial time convergence
- Ideas can be utilized in development of polynomial time algorithms for other optimization problems

Main ideas behind:

1. Iterative process starts from centre of feasible region to steepest decent direction
2. Used transformations in order to place current point near the center

- Include logarithmic barrier term in objective function f:
- $B(x, \mu) = f(x) - \mu \sum_{j=1}^{m} \ln c_j(x)$,
- Here μ is a parameter, as converges to zero to the minimum of $B(x, \mu)$ should converge to a solution of COP.



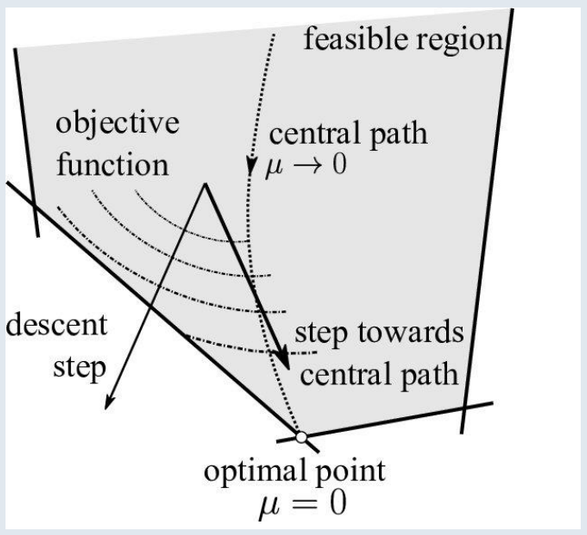
Concluding remarks and feature research

- Semidefinite and conic programming is an extremely lively research area today, producing new theory, algorithms, and implementations.
- IPM`s is a powerful tool in optimization, allowing for the efficient handling of a wide variety of real-world problems with structured constraints and objectives. Their adaptability and robustness make them particularly appealing for tackling complex real-world problems
- Numerous individual papers exist in the fields of semidefinite programming and conic programming, require thorough examination. Currently, there is no unified framework for applying interior point algorithms to various optimization problems with constraints. As a result, our team intends to integrate different types of interior point methods to evaluate their effectiveness on low-dimensional problems and extend these approaches primarily to large-scale issues.

Primal-Dual

- Iterative estimation of objective function proposed by Todd and Burrel solving dual for LP problem:
  $\max\{b^T y : A^T y + s = c, s \geq 0\}$,
  And also for standard LP from
- $\min\{\vec{c}^T * \vec{x}\}$ *such that* $Ax = 0$, $e^T x = 1, x \geq 0$,
- The main goal is to replace complementarity condition with parametrized condition $xs = \mu e, \mu >= 0$
- Advantages:
- More efficient than barrier in cases of high accuracy is needed
- often exhibit superliner asymptotic convergence
- search directions can be interpreted as Newton directions for modified KKT conditions
- could start at infeasible point
- cost per iteration same as barrier method
- Short and long step methods

Central path a new class of IPMs. These methods don't use Newtos direction, instead they use steepest decent direction for a so-called self-regular barrier function



REFERENCES

- THE INTERIOR_POINT REVOLUTION IN OPTIMIZATION__HISTORY_RECENT DEVELOPMENTS_AND LASTING CONSEQUENCES-Wright-2004
- Interior point optimization methods__theory_implementations and engineering applications-Vannelli-et_al-1992
- Renegar-A_mathematical_view_of_interior_point_methods_in_convex_optimization-2001
- Implementation of interior point methods for mixed semidefinite and second order cone optimization problems-Sturm-2002
- Krumke-Interior Point Methods__with Applications to Discrete Optimization-2005
- Jansen-Interior_Point_Techniques_in_Optimization__Complementarity_Sensitivity_and_Algorithms-1997

Acceleration Exists! Optimization Problems When Oracle Can Only Compare Objective Function Values

Aleksandr Lobanov[1,2,3], Alexander Gasnikov[1,2,3], Andrei Krasnov[1]

[1] MIPT [2] Skoltech Skolkovo Institute of Science and Technology [3] ISP RAS

ASCOMP 2024

ASCOMP 2024, Innopolis

Setup

Optimization problem, possibly non-convex, possibly stochastic:

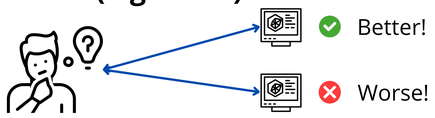
$$\min_{x \in \mathbb{R}^d} \{f(x) := \mathbb{E}_{\xi \sim \mathcal{D}} f_\xi(x)\}$$

Feedback, Deterministic Order Oracle:

$$\phi(x, y) = \text{sign}\left[f(x) - f(y) + \delta(x, y)\right]$$

Motivation

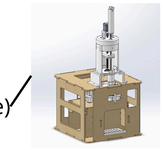
Human Feedback (e.g. RLHF):



Our motivation: AI chocolate, AI wine, etc.

(Valio's chocolate)

(Smart coffee machine)



Friendly method to our oracle

Algorithm Golden Ratio Method (GRM)

Input: Interval $[a, b]$
Initialization: Choose constants $\epsilon > 0$
$y \leftarrow a + (1 - \rho)(b - a)$
$z \leftarrow a + \rho(b - a)$ → $\rho = \frac{1}{\Phi} = \frac{\sqrt{5}-1}{2}$
while $b - a > \epsilon$ **do**
  if $\phi(y, z) = -1$ **then**
    $b \leftarrow z$
    $z \leftarrow y$
    $y \leftarrow a + (1 - \rho)(b - a)$
  else
    $a \leftarrow y$
    $y \leftarrow z$
    $z \leftarrow a + \rho(b - a)$
  end if
end while
Return: $\frac{a+b}{2}$

$$\frac{\text{z} - \text{a}}{\text{b} - \text{z}} = \frac{\text{y} - \text{a}}{\text{z} - \text{y}}$$

Trouble: GRM solve a one-dimensional optimization problem

Assumptions

***L*-coordinate-Lipschitz smoothness:**

$$|\nabla_i f(x + h\mathbf{e}_i) - \nabla_i f(x)| \leq L_i |h|; \quad i \in [d], x \in \mathbb{R}^d$$

(Strong) convexity w.r.t. the norm $\|\cdot\|_{[1-\alpha]}$:

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu_{1-\alpha}}{2} \|y - x\|^2_{[1-\alpha]}$$

Define norms (*Y. Nesterov, 2012*)**:**

$$\|\cdot\|_{[\alpha]} := \sqrt{\sum_{i=1}^d L_i^\alpha x_i^2} \text{ and } \|\cdot\|^*_{[\alpha]} := \sqrt{\sum_{i=1}^d \frac{1}{L_i^\alpha} x_i^2}$$

Non-Accelerated Methods

Algorithm Random Coordinate Descent with Order Oracle (OrderRCD)

Input: $x_0 \in \mathbb{R}^d$, random generator $\mathcal{R}_\alpha(L)$
for $k = 0$ **to** $N - 1$ **do**
1. choose active coordinate $i_k = \mathcal{R}_\alpha(L)$ → $p_\alpha(i) = L_i^\alpha / S_\alpha$ → $S_\alpha := \sum_i^d L_i^\alpha$
2. compute $\eta_k = \text{argmin}_\eta\{f(x_k + \eta \mathbf{e}_{i_k})\}$ via (GRM)
3. $x_{k+1} \leftarrow x_k + \eta_k \mathbf{e}_{i_k}$

end for
Return: x_N

Non-convex setting:

$$\frac{1}{N} \sum_{k=0}^{N-1} \left(\|\nabla f(x_k)\|^*_{[1-\alpha]}\right)^2 \leq \mathcal{O}\left(\frac{S_\alpha F_0}{N} + S_\alpha \epsilon + S_\alpha \Phi \Delta\right)$$

Convex setting:

$$\mathbb{E}\left[f(x_N)\right] - f(x^*) \leq \mathcal{O}\left(\frac{S_\alpha R^2_{[1-\alpha]}}{N} + \frac{2 S_\alpha R^2_{[1-\alpha]} (\epsilon + \Phi\Delta)}{F_{N-1}}\right)$$

Strongly convex setting:

$$\mathbb{E}\left[f(x_N)\right] - f(x^*) \leq \left(1 - \frac{\mu_{1-\alpha}}{S_\alpha}\right)^N F_0 + \frac{2 S_\alpha \epsilon}{\mu_{1-\alpha}} + \frac{2c S_\alpha \Phi \Delta}{\mu_{1-\alpha}}$$

Prior works

Reference	Nesterov (2012)	Gorbunov et al. (2019)	Saha et al. (2021)	Tang et al. (2023)	This paper	
Non-convex	✗	$\mathcal{O}\left(\frac{dS_\alpha F_0}{\varepsilon^2}\right)$	✗	$\mathcal{O}\left(\frac{dL_\alpha F_0}{\varepsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{S_\alpha F_0}{\varepsilon^2}\right)$	
Convex	$\mathcal{O}\left(\frac{S_\alpha R^2_{[1-\alpha]}}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{dS_\alpha R^2_{[1-\alpha]}}{\varepsilon} \log \frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{dLR^2}{\varepsilon}\right)$	✗	$\tilde{\mathcal{O}}\left(\frac{S_\alpha R^2_{[1-\alpha]}}{\varepsilon}\right)$	
Strongly convex	$\mathcal{O}\left(\frac{S_\alpha}{\mu_{1-\alpha}} \log \frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{S_\alpha}{\mu_{1-\alpha}} \log \frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(d\frac{L}{\mu} \log \frac{1}{\varepsilon}\right)$	✗	$\tilde{\mathcal{O}}\left(\frac{S_\alpha}{\mu_{1-\alpha}} \log \frac{1}{\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(\frac{S_{\alpha/2}}{\sqrt{\mu_{1-\alpha}}} \log \frac{1}{\varepsilon}\right)$
Order Oracle?	✗	✓	✓	✓	✓	✓
Acceleration?	✗	✗	✗	✗	✗	✓

Accelerated Method

Algorithm Accelerated Coordinate Descent Method with Order Oracle (OrderACDM)

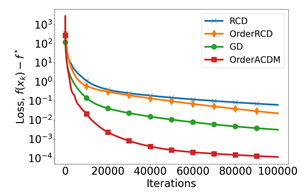
Input: $x_0 = z_0 \in \mathbb{R}^d, \mathcal{R}_\alpha(L), A_0 = 0, B_0 = 1, \beta = \frac{\alpha}{2}$
for $k = 0$ **to** $N - 1$ **do**
1. choose active coordinate $i_k = \mathcal{R}_\beta(L)$ → $p_\beta(i) = \frac{L_i^{\alpha/2}}{S_{\alpha/2}}$
2. find parameter a_{k+1} from $a_{k+1}^2 S_\beta^2 = A_{k+1} B_{k+1}$, where $A_{k+1} = A_k + a_{k+1}$ and $B_{k+1} = B_k + \mu_{1-\alpha} a_{k+1}$
3. $\alpha_k \leftarrow \frac{a_{k+1}}{A_{k+1}}$
4. $\beta_k \leftarrow \frac{\mu_{1-\alpha} a_{k+1}}{B_{k+1}}$
5. $y_k \leftarrow \frac{(1-\alpha_k) x_k + \alpha_k (1-\beta_k) z_k}{1 - \alpha_k \beta_k}$
6. compute $\eta_k = \text{argmin}_\eta\{f(y_k + \eta \mathbf{e}_{i_k})\}$ via (GRM)
7. $x_{k+1} \leftarrow y_k + \eta_k \mathbf{e}_{i_k}$
8. $w_k \leftarrow (1 - \beta_k) z_k + \beta_k y_k + \frac{a_{k+1} L_{i_k}^\alpha}{B_{k+1} p_\beta(i)} \eta_k \mathbf{e}_{i_k}$
9. compute $\zeta_k = \text{argmin}_\zeta\{f(w_k + \zeta \mathbf{e}_{i_k})\}$ via (GRM)
10. $z_{k+1} \leftarrow w_k + \zeta_k \mathbf{e}_{i_k}$

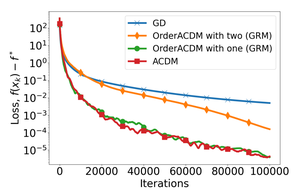
end for
Return: x_N

Convergence in strongly convex setting:

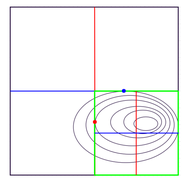
$$\mathbb{E}\left[f(x_N)\right] - f(x^*) \leq \left(1 - \frac{\sqrt{\mu_{1-\alpha}}}{S_{\alpha/2}}\right)^N F_0$$

Experiments





Low-dimensional case



Golden ration method + Nesterov's 2D generalization

Total number of Order Oracle calls:

$$\sim \log^2\left(\frac{LR}{\varepsilon}\right)$$

Stochastic Order Oracle

$$\phi(x, y, \xi) = \text{sign}\left[f(x, \xi) - f(y, \xi)\right]$$ → New feedback

Algorithm:

$$x_{k+1} = x_k - \eta_k \phi(x_k + \gamma \mathbf{e}_k, x_k - \gamma \mathbf{e}_k, \xi_k) \mathbf{e}_k$$

smoothing parameter; uniformly distributed on the Euclidean sphere

Asymptotic convergence: $\sqrt{N}(x_k - x^*) \sim \mathcal{N}(0, V)$, where the matrix V is:

$$V = \frac{\eta^2}{d}\left(2\eta(1 - 1/d)\frac{c}{\sqrt{d}}\alpha \nabla^2 f(x^*) - I\right)^{-1}$$

$2\eta(1 - 1/d)\frac{c}{\sqrt{d}}\alpha \nabla^2 f(x^*) > I$; $\alpha = \int \|z\|^{-1} dP(z) < \infty$

LLM for RecSys

Ainura Zakirova [1], Irina Maltseva [1], Andrei Semenov [2], Robert Zaraev [2], Dmitri Kiselev [3]

[1]Innopolis university,[2] ITMO university, [3] AIRI

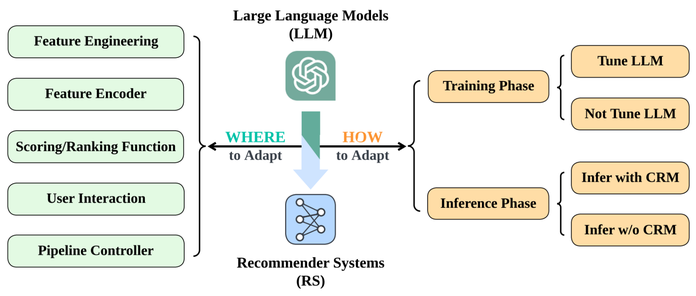
Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across various tasks, including recommendation systems. However, comparing the performance of LLM-based models with traditional benchmarks has been challenging due to the absence of a unified evaluation framework. To address this, we developed LLM4Rec, a comprehensive framework that integrates LLMs into multiple stages of the recommendation process. This framework facilitates fair performance comparisons between LLM-based recommendation systems and traditional models.

Literature Review

Recent research on applying LLMs to recommendation systems identifies several key applications [2]:

- LLMs generate detailed user profiles by summarizing experiences and adding personalized traits to item descriptions.
- They serve as feature encoders, transforming open-world and semantic knowledge into dense vectors to enhance user and item representations.
- LLMs are utilized for scoring and ranking tasks, leveraging their text understanding and reasoning capabilities.
- Their conversational abilities can be used to make recommendations interactive and personalized, enhancing transparency in user interactions.
- LLMs can function as autonomous agents, controlling the recommendation system pipeline, adapting strategies based on feedback, and simulating specific roles to further personalize the user experience.



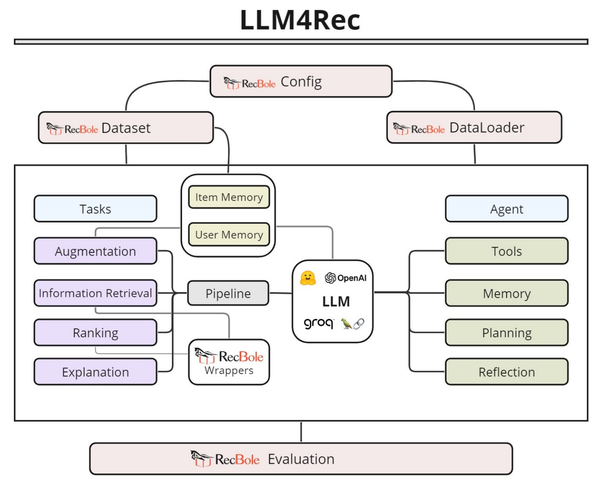
How to use LLM in RecSys

Methodology

LLM4Rec is designed to help researchers develop and evaluate recommendation models that leverage LLMs. Based on our review of current literature on LLM applications in recommendation systems, we have identified and implemented key use cases of LLM within the recommendation pipeline. To facilitate a standardized environment for performance evaluation and comparison with traditional models, we have integrated LLM4Rec with the RecBole [3] framework, which supports a wide range of datasets and models.

Framework structure

The framework is structured into two key components. The first is dedicated to constructing the experimental setup with methods adopted from RecBole. The second component focuses on the integration of LLMs into the recommendation system pipeline. There are two applications supported: running LLM-based recommendation tasks in a sequential pipeline and implementing LLM agents within the recommendation system.

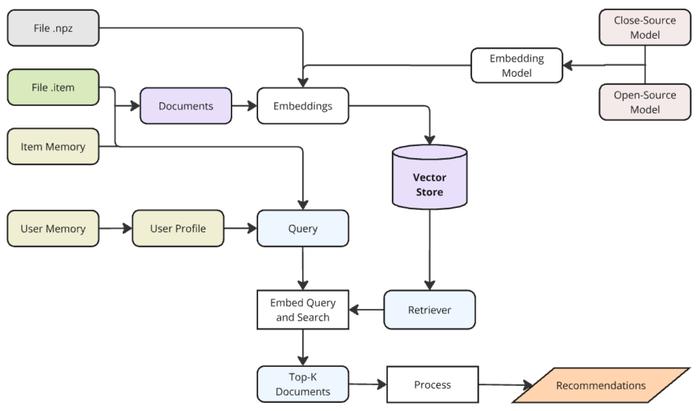


Framework structure

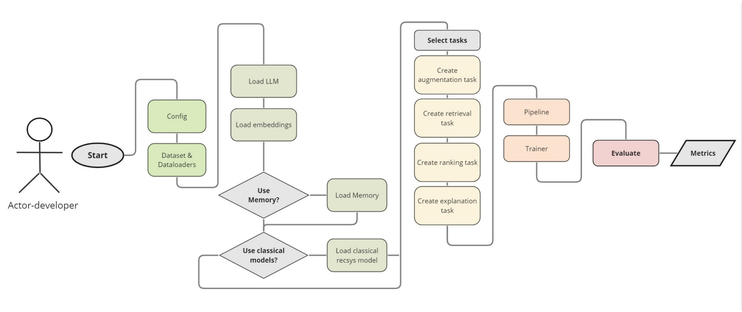
Sequential pipeline of recommendation tasks

The proposed framework incorporates a sequential use of Large Language Models across various stages of the recommendation system, forming an integrated pipeline. The components implemented within this pipeline include:

- **Information retrieval**: uses a FAISS retrieval model to efficiently find relevant items.
- **Ranking**: employs an LLM-based ranking system (LLMRank [1]) to reorder the retrieved items.
- **Augmentation**: builds user and item memories, populated from historical interactions and external data sources, to enhance the recommendation context.
- **Explanation**: uses LLMs to generate explanations for each recommendation provided to the user.



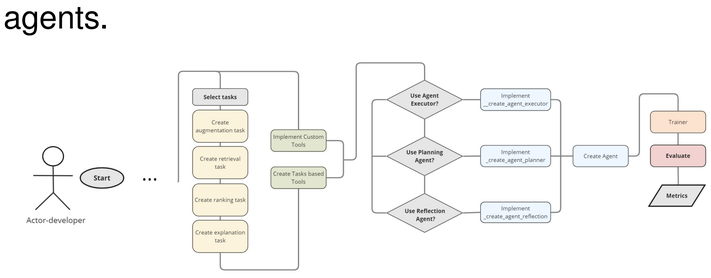
Information retrieval task example



Example of sequential pipeline of tasks

Agents

The framework is designed to support LLM based agents, that can be integrated into conversational systems and manage the recommendation system pipeline. It includes key related components such as various tools, memory, and mechanisms for reflection and planning to facilitate the use of these agents.



Example of agent pipeline

Experiments

We evaluated our framework through a series of experiments using the MovieLens-100K dataset. Our analysis included comparisons between both open-source and closed-source models, as well as with traditional recommendation system models. For Information Retrieval component we compared various configurations of item features, including overviews extracted from Wikipedia.

Table 1: Comparison of Information Retrieval

Model	Item Features	Recall@20	Recall@50	Recall@100
all-MiniLM-L6-v2	title genres year	0.0456	0.0912	0.1495
all-MiniLM-L6-v2	title genres year + augmentation	0.0180	0.0467	0.0880
text-embedding-ada-002	title genres year	0.0467	**0.0986**	**0.1569**
text-embedding-ada-002	title genres year + augmentation	0.0350	0.0679	0.1050
ALS	-	**0.0615**	0.0891	0.1135

Table 2: Comparison of sequental pipeline

Retrieval Model	Ranker model	Recall@10	Recall@20	NDCG@10	NDCG@20
text-embedding-ada-002	LLAMA3-70B	0.0286	0.0414	0.0157	0.0189
text-embedding-ada-002	gpt-3.5-turbo	0.0286	**0.0456**	0.0162	0.0204
all-MiniLM-L6-v2	gpt-3.5-turbo	**0.0329**	**0.0456**	**0.0183**	**0.0216**

Table 3: Performance of traditional recommender system model

Model	Recall@5	Recall@10	Recall@20	NDCG@5	NDCG@10	NDCG@20
SASRec	0.0647	0.1166	0.2195	0.0382	0.0548	0.0808

In our experiments, the best results for information retrieval were achieved using embeddings that excluded movie overviews from Wikipedia.These overviews, although comprehensive, may include irrelevant details that negatively impact similarity searches. While closed-source OpenAI models outperformed open-source alternatives, they were less effective than traditional models like ALS embeddings for information retrieval and the SASRec recommendation model.

Conclusion

We have developed LLM4Rec, a comprehensive framework for integrating and reproducibly evaluating LLM-based components within recommendation systems. While LLM-based recommendation systems currently underperform traditional algorithms in some scenarios, they show promise in addressing challenges such as the cold-start problem and improving conversational recommender systems.

References

[1] Y. Hou, J. Zhang, Z. Lin, H. Lu, R. Xie, J. McAuley, and W. X. Zhao. Large language models are zero-shot rankers for recommender systems, 2024.

[2] J. Lin, X. Dai, Y. Xi, W. Liu, B. Chen, H. Zhang, Y. Liu, C. Wu, X. Li, C. Zhu, H. Guo, Y. Yu, R. Tang, and W. Zhang. How can recommender systems benefit from large language models: A survey, 2024.

[3] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Lu, H. Wang, C. Tian, Y. Min, Z. Feng, X. Fan, X. Chen, P. Wang, W. Ji, Y. Li, X. Wang, and J.-R. Wen. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms, 2021.

https://github.com/ainura-z/llm-for-rec

Comparison of Traditional and AI-based Models for IMD2 Cancellation

A. A. Degtyarev, N.V. ,Bakholdin, A. Y. Maslovskiy, S. A. Bakhurin

Motivation

The use of **direct conversion receivers (DCRs)** in modern smartphones has become widespread due to their simplicity and ability to support multiple frequency bands. Self-interference cancellation is **essential** for these devices, and traditional approaches involve adapting behavioral models[1]. However, digital interference cancellation through neural network training has emerged as a promising alternative. Neural network-based models are capable of learning temporal information and accounting for the memory effects of nonlinearity[2]. This work focuses on researching second intermodulation distortion (IMD2) generated by the nonlinear distortion of a single RF mixer using complex data for transmission (Tx) and real-valued samples for reception (Rx).

Traditional and NN methods

Behavioral modelling is introduced by the **generalized memory polynomial (GMP)** model[3]. Polynomial model is known as a structure, which describes the PA physical properties for different PA kinds and modes. For the task of IMD2 cancellation special case of GMP is decided to be exploited. Moreover, currently we use orthogonal Chebyshev polynomials basis, which could be expressed mathematically as:

$$y_n = \sum_{k=0}^{K-1} \sum_{p=0}^{P-1} \theta_{k,p} T_p(|x_{n-d_k}|),$$

$$T_p(|x_{n-d_k}|) = \cos(n \cdot \arccos(|x_{n-d_k}|)),$$

where $\theta_{k,p} \in \mathbb{R}$ -parameters of Chebyshev polynomial model, x - samples of baseband (BB) signal d- signal samples delays, $T_p(\cdot)$- order Chebyshev polynomial of the first kind.

Neural network architecture for IMD2 cancellation is shown lower. Since non-linearity is inertial, NN structure is implied to take into account memory effects. In current work we realized really small non-linear model based on behavioral modelling approach. As an example architecture to follow we have chosen Wiener-Hammerstein model.

$$y_n = \boldsymbol{W}_{\text{out}} \sigma_{L-1}(\boldsymbol{W}_{L-1} \cdots \sigma_1(\boldsymbol{W}_1 \sigma_0(\boldsymbol{W}_0 \boldsymbol{f}_n))),$$

$$\boldsymbol{f}_n = (|x_{n-d_0}| \quad |x_{n-d_1}| \quad \cdots \quad |x_{n-d_{M-1}}|).$$

Explanation of IMD2 generation

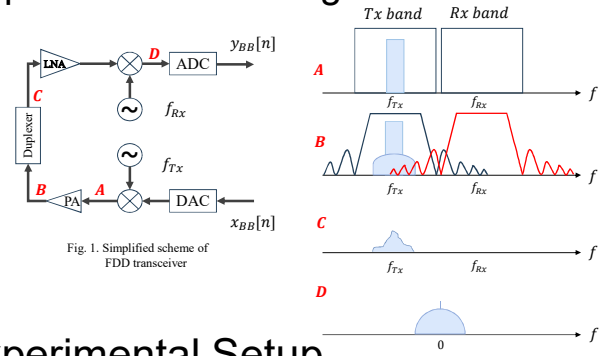


Fig. 1. Simplified scheme of FDD transceiver

Experimental Setup

The setup is shown in fig. 1. It has a computer where data are loaded. Then they go to the SMW200A generator. The signal is amplified by a PA ZRL-3500+. It has 26 dB gain and 24 dBm power at 1 dB. The PA has an OIP3 of 42 dBm. After the PA, there is a bandpass filter. It is like a duplexer. It has 30 dB stopband attenuation. The output of the BPF goes to an LNA ZRL-3500+. Its NF is 2 dB. There is also a ZX05-63LH-S+ mixer. It has 37 dB LO-to-RF isolation and 30 dB LO-to-IF isolation. It uses a complex valued OFDM signal. The bandwidth is 5 MHz. The transmit frequency is 814 MHz and the receive frequency is 859 MHz. The duplex spacing is 45 MHz.. The signal from the transmitter goes through the PA and leaks into the receiver. It has a frequency of 45 MHz. The receiver has a LO signal with power 10 dBm. The transmitter power is 8 dBm and the duplexer has an attenuation of 30 dB at 814 MHz. This makes the power on the mixer input 4 dBm.

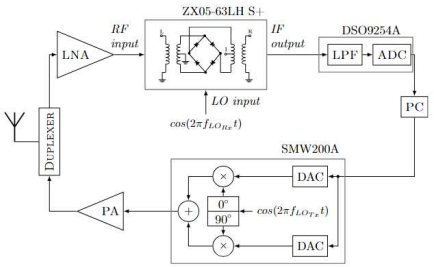
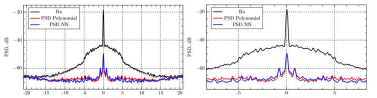


Figure 2: The scheme of testbench

Results

In the current article we researched NN and polynomial based models for IMD2 cancellation induced by Tx leakage signal in presence of limited stopband attenuation of duplexer.% Based on the achieved experiments there were shown that NN with SGD-like method can achieve better performance than polynomial with less resources: up to NMSE -23.6 dB with 17 coefficients. L-BFGS method showed us closest to the best performance which we can achieve for both traditional and NN based cancellation strategies.% The convergence speed of L-BFGS method is higher than for Adam method for both cancellation strategies, it can provide good performance starting from 1000 iterations whereas for Adam we need 5 times more counts to converge.% We found that after delay's search we can achieve performance better, but NN allowed us to find the best performance without any fine tuning.Current paper presents that both neural network and Chebyshev polynomial based models can achieve good performance but NN model can suppress IMD2 signal without any parameter tuning, whereas for polynomial model requires searching the set of optimal delays.The findings show that the L-BFGS approach delivers performance for both architectures near to the LS solution for polynomial NMSE=-23.59 dB. Furthermore, the L-BFGS simulation method for both structures requires fewer than 2000 epochs. Current findings demonstrate its use in the evaluation of models' performance in the interference cancellation domain.Due to neural networks' capacity for generalization, the first-order technique for NN-based models also demonstrates a greater convergence rate when compared to polynomial-based cancellers. For example, in 20000 epochs, the NN architecture achieves 0.44 dB performance gain over the polynomial. However, polynomial can reach full convergence performance by fine-tuning the first-order optimizer parameters. This demonstrates one of the amazing benefits of NN architectures.[1]



(a) IMD2 before and after cancellation (b) IMD2 before and after cancellation, magnified view

Model	Algorithm	Number of iterations		
		1000	5000	20000
Polynomial	LS	23.59	23.59	23.59
	Adam	21.96	22.76	22.91
	LBFGS	23.41	23.41	23.41
NN	LS	N/A	N/A	N/A
	Adam	21.00	22.03	23.35
	LBFGS	23.20	23.63	23.63

Table 1 . Comparison table for different models and optimization algorithms

[1] Shinil Chang and Hyunchol Shin. "2.4-GHz CMOS Bluetooth RF Receiver With Improved IM2 Distortion Tolerance". In: IEEE Transactions on Microwave Theory and Techniques 68.11 (2020),

[2] Yann Kurzo, Andreas Toftegaard Kristensen, Andreas Burg, and Alexios Balatsoukas-Stimming. "Hardware Implementation of Neural Self-Interference Cancellation". In: IEEE Journal on Emerging and Selected Topics in Circuits and Systems 10.2 (2020)

[3] Ville Syrjala, Mikko Valkama, Lauri Anttila, Taneli Riihonen, and Dani Korpi. "Analysis of oscillator phase-noise effects on self-interference cancellation in full-duplex OFDM radio transceivers". In: IEEE Transactions on Wireless Communications 1

Rejection-based EEG Signal Classification

Artyom Matveev[1]
matveev.as@phystech.edu
[1]Moscow Institute of Physics and Technology

ASCOMP 2024

Problem Definition and Contribution

Goal: Find an optimal sample such that it includes a minimum amount of information needed to solve an EEG signal classification problem robustly.
Key Contributions: A new approach to the classification of EEG signals.

- First, we reconstruct the probability density function of each class, taking the Riemannian Gaussian distribution of data into account [1].
- Second, we define a specific confidence interval for each class so that we can use it in our rejection strategy.
- Third, we solve the classification problem by evaluating the statistical significance of data concerning the classes' distributions.

Formulation

Assumption: There is an optimal sample size that is enough to make a robust decision during the classification of signals.

1. Let $\mathfrak{D} = \{(\mathbf{X}_i, y_i)_{i=1}^{L}\}$ be the given dataset, where a segment of EEG signals

$$\mathbf{X}_i = [\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_j, \ldots, \boldsymbol{\chi}_{n_C}]^{\mathrm{T}}, \mathbf{X}_i \in \mathbb{R}^{n_C \times n_T} \text{ and } j \in \mathcal{J} = \{1, \ldots, n_C\}.$$

2. Let $\mathbf{P}_i \in \mathbb{S}_{++}$ be a Symmetric Positive Definite (SPD) matrix, where

$$\mathbf{P}_i = \frac{1}{n_T - 1}\mathbf{X}_i\mathbf{X}_i^{\mathrm{T}}$$

3. Let $y_i \in \mathbb{Y} = \{1, \ldots, K\}$ be a class label.

4. Let $p(y|\mathbf{P}, \mathbf{w})$ be a parametric family, where $\mathbf{w} \in \mathbb{R}^n$.

5. The likelihood function is then as follows:

$$L(\mathfrak{D}, \mathbf{w}) = \prod_{i=1}^{m} p(y_i|\mathbf{P}_i, \mathbf{w}), \quad l(\mathfrak{D}, \mathbf{w}) = \sum_{i=1}^{m} \log p(y_i|\mathbf{P}_i, \mathbf{w})$$

Find:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w} \in \mathbb{R}^n} L(\mathfrak{D}, \mathbf{w})$$

Criterion:

ROC AUC

Method

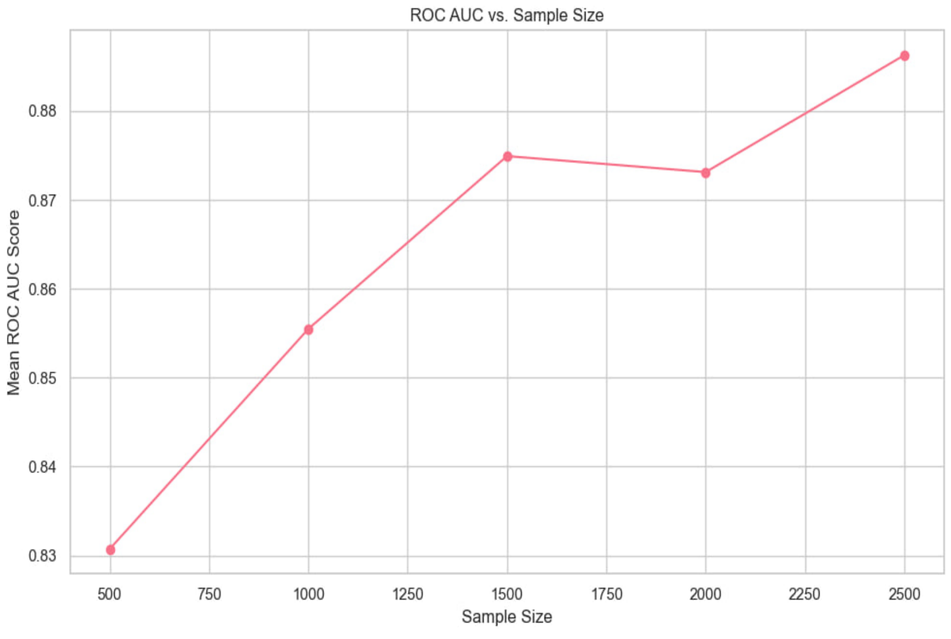
Sufficient Sample Size Estimation. Bootstrap:
Given some sample size m calculate the quantile confident intervals

$$(a_1^m, b_1^m), (a_2^m, b_2^m), \ldots, (a_n^m, b_n^m)$$

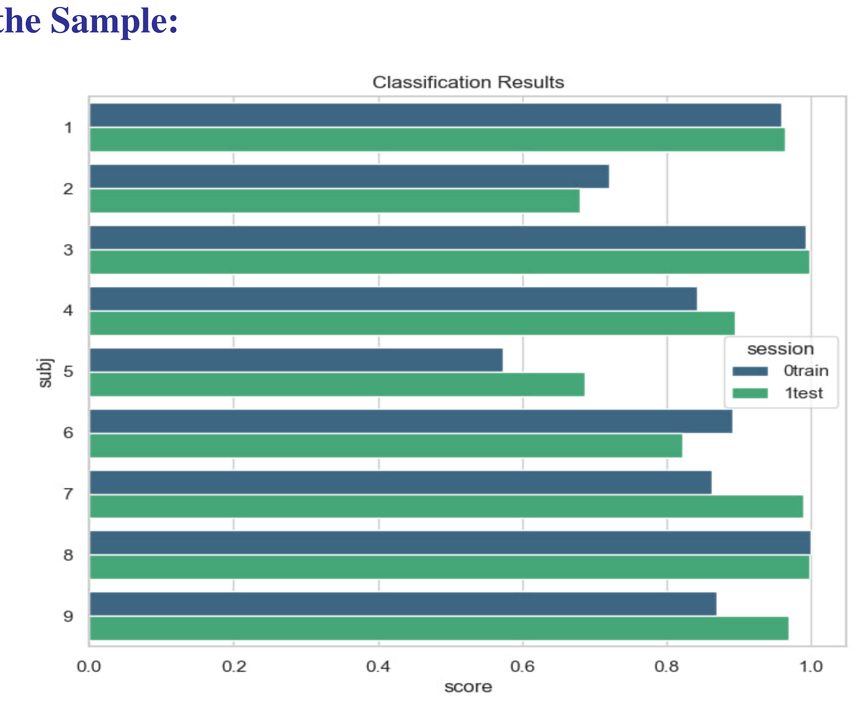
with the significance level of α using bootstrap for every parameter of the model.
Sufficient sample size m^* : $\forall m \geq m^* \max_i (b_i^m - a_i^m) < l$, where (a_i^m, b_i^m) is a quantile bootstrap confident interval calculated on the i-th bootstrap subset of the m size [2].

Experiments & Results

Quantitative Results. ROC AUC vs Sample Size:



Quantitative Results. ROC AUC Score Given 2500 Examples in the Sample:



References

[1] S. Said, S. Heuveline, and C. Mostajeran, "Riemannian statistics meets random matrix theory: Toward learning from high-dimensional covariance matrices," *IEEE Transactions on Information Theory*, vol. 69(1), 2023.

[2] A. Grabovoy, T. Gadaev, A. Motrenko, and V. Strijov, "Numerical methods of sufficient sample size estimation for generalised linear models," *Lobachevskii Journal of Mathematics*, vol. 43, 2022.

Shuffling Heuristic in Variational Inequalities: Establishing New Convergence Guarantees

Daniil Medyakov [1, 2] Gleb Molodtsov [1, 2] Grigoriy Evseev [1] Egor Petrov [1] Aleksandr Beznosikov [1, 2, 3]

[1] Moscow Institute of Physics and Technology [2] Institute for System Programming RAS [3] Innopolis University

Distributed problem

• **Variational Inequality (VI)**

Find $z^* \in Z : \forall z \in Z \hookrightarrow \langle F(z^*), z - z^* \rangle + g(z) - g(z^*) \geqslant 0, \quad (1)$

where F is a monotone operator and g is a proper convex lower semicontinuous function, which plays the role of regularizer.

• Training data describing F is **distributed** across n devices: $F(z) = \frac{1}{n}\sum_{i=1}^{n} F_i(z)$, where each F_i corresponds to an individual data point.

Example (Convex optimization)

$$\min_{z \in \mathbb{R}^d} [f(z) + g(z)]. \quad (2)$$

In this example, f is a smooth data representative term, and g is probably a non-smooth regularizer. In this setting, we define $F(z) = \nabla f(z)$. Then $z^* \in \operatorname{dom} g$ is the solution of (1) if and only if $z^* \in \operatorname{dom} g$ is the solution of (2). In this way, the problem (2) can be considered as a variational inequality.

Example (Convex-concave saddles)

We consider the following convex-concave saddle point problem:

$$\min_{x \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}^{d_y}} [f(x, y) + g_1(x) - g_2(y)]. \quad (3)$$

There, f has the same interpretation as in Example 1, and g_1, g_2 can also be perceived as regularizers. In this setting, we define $F(z) = F(x, y) = [\nabla_x f(x, y), -\nabla_y f(x, y)]$. Then $z^* \in \operatorname{dom} g_1 \times \operatorname{dom} g_2$ is the solution of (1) if and only if $z^* \in \operatorname{dom} g_1 \times \operatorname{dom} g_2$ is the solution of (3). In this way, the problem (3) can be considered as a variational inequality.

Variance Reduction methods

We explore stochastic algorithms which are particularly suitable for practical extensive applications. The stochastic version of the EXTRAGRADIENT method select random independent indexes i_t, j_t at iteration t and performs the following updates:

$$z^{t+\frac{1}{2}} = z^t - \gamma F_{i_t}(z^t),$$
$$z^{t+1} = z^t - \gamma F_{j_t}(z^{t+\frac{1}{2}}).$$

The **variance reduction (VR)** technique was developed for a classical finite-sum minimization task. Considering convex optimization problem (see Example 1), we can formally write the stochastic reduced gradient at the point $z^{t+\frac{1}{2}}$ as

$$\nabla \hat{f}_{i_t}(z^{t+\frac{1}{2}}) = \nabla f_{i_t}(z^{t+\frac{1}{2}}) - \nabla f_{i_t}(\omega^t) + \nabla f(\omega^t).$$

Setup

Assumption 1: Each operator F_i is L-Lipschitz, i.e., it satisfies

$$\|F_i(z_1) - F_i(z_2)\| \leq L\|z_1 - z_2\|$$

for any $z_1, z_2 \in Z$.

Assumption 2: Each operator F_i is μ-strongly monotone, i.e., it satisfies

$$\langle F_i(z_1) - F_i(z_2), z_1 - z_2 \rangle \geqslant \mu\|z_1 - z_2\|^2$$

for any $z_1, z_2 \in Z$.

Assumption 3: Each stochastic operator F_i and full operator F is bounded at the point of the solution $z^* \in \operatorname{dom} g$, i.e.,

$$\mathbb{E}\|F_i(z^*)\|^2 \leqslant \sigma_*^2, \|F(z^*)\|^2 \leqslant \sigma_*^2.$$

Proximal Algorithm

We often encounter the need to minimize the function decomposed into two parts: a smooth differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ and a possibly non-smooth function $g : \mathbb{R}^n \to \mathbb{R}$ which is proximal friendly. To solve it, we can utilize the proximal gradient method:

$$\operatorname{prox}_g(z) = \arg\min_{y \in \mathbb{R}^n} \left\{ g(y) + \frac{1}{2}\|y - z\|^2 \right\}.$$

The update step for solving the problem can be written as

$$z^{t+1} = \operatorname{prox}_{\alpha_t g}\left(z^t - \alpha_t \nabla f(z^t)\right).$$

Table: Comparison of the convergence results for the methods for solving VI.

Algorithm	Sampling	VR?	Strongly Monotone Complexity	Monotone Complexity
Extragradient (Korpelevich, 1976; Mokhtari, 2020)	D	✗	$\tilde{\mathcal{O}}\left(\frac{nL}{\mu}\right)$	$\mathcal{O}\left(\frac{nL}{\varepsilon}\right)$
Mirror-prox (Nemirovski, 2004)	D	✗	\	$\mathcal{O}\left(\frac{nL}{\varepsilon}\right)$
FBF (Tseng, 2000)	D	✗	\	$\mathcal{O}\left(\frac{nL}{\varepsilon}\right)$
FoRB (Malitsky, 2020)	D	✗	\	$\mathcal{O}\left(\frac{nL}{\varepsilon}\right)$
Mirror-prox (Juditsky, 2011)	I	✗	\	$\mathcal{O}\left(\frac{L}{\varepsilon} + \frac{1}{\varepsilon^2}\right)$
Extragradient (Beznosikov, 2020)	I	✗	$\tilde{\mathcal{O}}\left(\frac{L}{\mu} + \frac{1}{\mu^2\varepsilon}\right)$	$\mathcal{O}\left(\frac{L}{\varepsilon} + \frac{1}{\varepsilon^2}\right)$
REG (Mishchenko, 2020)	I	✗	$\tilde{\mathcal{O}}\left(\frac{L}{\mu} + \frac{1}{\mu^2\varepsilon}\right)$	$\mathcal{O}\left(\frac{L}{\varepsilon} + \frac{1}{\varepsilon^2}\right)$
Extragradient (Carmon, 2019)	I	✓	\	$\tilde{\mathcal{O}}\left(n + \frac{\sqrt{n}L}{\varepsilon}\right)$
Mirror-prox (Carmon, 2019)	I	✓	\	$\tilde{\mathcal{O}}\left(n + \frac{\sqrt{n}L}{\varepsilon}\right)$
FBF (Palaniappan, 2016)	I	✓	$\tilde{\mathcal{O}}\left(n + \frac{\sqrt{n}L}{\mu}\right)$	$\tilde{\mathcal{O}}\left(n + \frac{\sqrt{n}L}{\varepsilon}\right)^{(1)}$
Extragradient (Chavdarova, 2019)	I	✓	$\tilde{\mathcal{O}}\left(n + \frac{L^2}{\mu^2}\right)$	$\tilde{\mathcal{O}}\left(n + \frac{L^2}{\varepsilon^2}\right)^{(1)}$
FoRB (Alacaoglu, 2021)	I	✓	\	$\mathcal{O}\left(n + \frac{nL}{\varepsilon}\right)$
Extragradient (Alacaoglu, 2022)	I	✓	$\tilde{\mathcal{O}}\left(n + \frac{\sqrt{n}L}{\mu}\right)$	$\mathcal{O}\left(n + \frac{\sqrt{n}L}{\varepsilon}\right)$
Mirror-prox (Alacaoglu, 2022)	I	✓	\	$\mathcal{O}\left(n + \frac{\sqrt{n}L}{\varepsilon}\right)$
Extragradient (this paper)	RR / SO	✗	$\tilde{\mathcal{O}}\left(n + \frac{L}{\mu} + \frac{n^2}{\mu^2\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(n + \frac{L}{\varepsilon} + \frac{n^2}{\varepsilon^3}\right)^{(1)}$
Extragradient (this paper)	RR / SO	✓	$\tilde{\mathcal{O}}\left(n\frac{L^2}{\mu^2}\right)$	$\tilde{\mathcal{O}}\left(n\frac{L^2}{\varepsilon^2}\right)^{(1)}$

Columns: Sampling = D if considered deterministic method, I if method uses independent choice of operator's indexes, RR / SO if method uses shuffling heuristic, Assumption = assumption on operator F, VR? = whether the method uses variance reduction technique.

Notation: μ = constant of strong monotonicity, L = Lipschitz constant of F, $\overline{L}$ = Lipschitz in mean constant, i.e. $1/n\sum_{i=1}^{n}\|F_i(z_1) - F_i(z_2)\| \leqslant L\|z_1 - z_2\|\ \forall z_1, z_2 \in Z$, n = size of the dataset, ε = accuracy of the solution.
(1): This result is obtained with regularization trick: $\mu \sim \varepsilon/D^2$.

Main Contributions

• **Novel approach to proof.** We present a technique that allows us to "return" to the starting point of an epoch in which there is a property of unbiasedness.

• **Convergence estimates.** We provide the first theoretical convergence rates for shuffling methods applied to the finite-sum variational inequality problem considering both EXTRAGRADIENT (our linear term coincides with that for the method without shuffling) and EXTRAGRADIENT with VR (the first to obtain a linear convergence estimate for methods with shuffling in VIs).

• **Experiments.** Our experiments emphasize the superiority of shuffling over the random index selection heuristic. We also consider two classical practical applications: adversarial training and image denoising.

Algorithms and convergence analysis

The **unbiasedness** of stochastic operators complicates the analysis. However, the equality holds at two points: z_s^0, the first point of the epoch, and z^*. Thus, we can leverage the unbiased operators by **"going back"** to the start of the epoch. This approach is also useful for methods involving Markov chains, where the unbiased property only holds at the chain's correlation point.

Extragradient

Algorithm 1. RR EXTRAGRADIENT

Input: Starting point $z_0^0 \in \mathbb{R}^d$
Parameter: Stepsize γ

for $s = 0, 1, 2, \ldots, S-1$ **do**
  Generate a permutation $\pi_0, \pi_1, \ldots, \pi_{n-1}$ of sequence $\{1, 2, \ldots, n\}$
  for $t = 0, 1, 2, \ldots, n-1$ **do**
    $z_s^{t+\frac{1}{2}} = \operatorname{prox}_{\gamma g}\left(z_s^t - \gamma F_{\pi_s^t}(z_s^t)\right)$
    $z_s^{t+1} = \operatorname{prox}_{\gamma g}\left(z_s^t - \gamma F_{\pi_s^t}(z_s^{t+\frac{1}{2}})\right)$
  end for
  $z_s^n = z_{s+1}^0$
end for
Output: z_S^n

Algorithm 2. SO EXTRAGRADIENT

Input: Starting point $z_0^0 \in \mathbb{R}^d$
Parameter: Stepsize γ

for $s = 0, 1, 2, \ldots, S-1$ **do**
  Generate a permutation $\pi_0, \pi_1, \ldots, \pi_{n-1}$ of sequence $\{1, 2, \ldots, n\}$
  for $t = 0, 1, 2, \ldots, n-1$ **do**
    $z_s^{t+\frac{1}{2}} = \operatorname{prox}_{\gamma g}\left(z_s^t - \gamma F_{\pi_s^t}(z_s^t)\right)$
    $z_s^{t+1} = \operatorname{prox}_{\gamma g}\left(z_s^t - \gamma F_{\pi_s^t}(z_s^{t+\frac{1}{2}})\right)$
  end for
  $z_s^n = z_{s+1}^0$
end for
Output: z_S^n

Theorem 1

Suppose Assumptions 1, 2, 3 hold. Then for Algorithms 1, 2 with $\gamma \leqslant \min\left\{\frac{1}{2\mu n}, \frac{1}{6L}\right\}$ after S epochs,

$$\|z_S^n - z^*\|^2 \leqslant (1 - \frac{\gamma\mu}{2})^{Sn}\|z_0^0 - z^*\|^2 + \frac{256\gamma n^2\sigma_*^2}{\mu}.$$

Remark 1

To transform the obtained estimation for the case of monotone stochastic operators, we use a regularization trick with $\mu \sim \frac{\varepsilon}{D}$. Thus, we obtain $\tilde{\mathcal{O}}\left(n + \frac{L}{\varepsilon} + \frac{n^2}{\varepsilon^3}\right)$ iteration and oracle complexity. This is convergence in argument, it differs from the classical form.

Our result is a great achievement in the shuffling theory, since despite the deterioration on n in the sublinear term, the estimation on the **linear term coincides** with that in the classical setting with independent choice of stochastic operators.

Extragradient with Variance Reduction

Previously, authors used a more classical version and compute $F(\omega_s^t)$ at the beginning of each epoch. We consider another option and compute this full operator randomly with probability p. We put $p = \frac{1}{n}$ not to increase the oracle complexity and obtain that on average we also update the full operator once per epoch.

Theorem 2

Suppose that Assumptions 1, 2 hold. Then for Algorithm 3 with $\gamma \leqslant \frac{(1-\alpha)\mu}{6L^2}, p = \frac{1}{n}$ and $V_s^t = \mathbb{E}\|z_s^t - z^*\|^2 + \mathbb{E}\|\omega_s^t - z^*\|^2$ after T iterations,

$$V_S^n \leqslant \left(1 - \frac{\gamma\mu}{4}\right)^T V_0^0.$$

Remark 2

Similarly to Remark 1, we can use our result in the monotone case by the regularization trick and obtain $\tilde{\mathcal{O}}\left(n\frac{L^2}{\varepsilon^2}\right)$.

We remove the variance that arose in Theorem 1 and **obtain linear convergence**. However, according to current theory, methods with the shuffling heuristic are inferior to those with independent sampling for variance reduction methods.

Experiments (Adversarial Training)

We address an adversarial training problem. Let us formulate it in the following way:

$$\min_{w \in \mathbb{R}^d} \max_{\|r_i\| \leqslant D} \left[\frac{1}{2N}\sum_{i=1}^{N}\left(w^T(x_i + r_i) - y_i\right)^2 + \frac{\lambda}{2}\|w\|^2 - \frac{\beta}{2}\|r\|^2\right], \quad (4)$$

where the samples corresponds to features x_i and targets y_i. The results are presented in Figure 1.

Algorithm 3. RR/SO EXTRAGRADIENT with variance reduction

Input: Parameters: z_0^0, ω_0^0
Parameter: Stepsize γ, $\alpha \in (0, 1)$
Generate a permutation $\pi_0, \pi_1, \ldots, \pi_{n-1}$ of sequence $\{1, 2, \ldots, n\}$ // SO heuristic
for $s = 0, 1, \ldots$ **do**
  Generate a permutation $\pi_0, \pi_1, \ldots, \pi_{n-1}$ of sequence $\{1, 2, \ldots, n\}$ // RR heuristic
  for $t = 0, 1, \ldots, n-1$ **do**
    $\overline{z}_s^t = \alpha z_s^t + (1 - \alpha)\omega_s^t$
    $z_s^{t+1/2} = \operatorname{prox}_{\gamma g}\left(\overline{z}_s^t - \gamma F(\omega_s^t)\right)$
    $\hat{F}(z_s^{t+1/2}) = F_{\pi_s^t}(z_s^{t+1/2}) - F_{\pi_s^t}(\omega_s^t) + F(\omega_s^t)$
    $z_s^{t+1} = \operatorname{prox}_{\gamma g}\left(\overline{z}_s^t - \gamma\hat{F}\left(z_s^{t+1/2}\right)\right)$
    $\omega_s^{t+1} = \begin{cases} z_s^t, & \text{with probability } p \\ \omega_s^t, & \text{with probability } 1 - p \end{cases}$
  end for
  $z_{s+1}^0 = z_s^n$
  $\omega_{s+1}^0 = \omega_s^n$
end for
Output: z_S^n

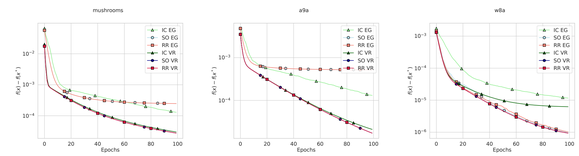


Figure: EXTRAGRADIENT with and without VR compared using various shuffling heuristics on `mushrooms`, `a9a` and `w8a` datasets on the problem (4).

Experiments (Image Denoising)

We consider the classic saddle point problem as in Example 2:

$$\min_{u \in \mathcal{X}} \max_{p \in \mathcal{Y}} [\langle Kx, y\rangle + G_1(x) - G_2(y)],$$

where G_1 and G_2 are proper convex lower semicontinuous regularizers, and K is a continuous linear operator. Let g be a given noisy image and u – a solution we seek. Thus, for the image denoising,

$$\min_{u \in \mathcal{X}} \max_{p \in \mathcal{Y}} \left[\langle \nabla u, p\rangle_{\mathcal{Y}} + \lambda/2\|u - g\|_2^2 - \delta_P(p)\right]$$

is the considered problem with p being a dual variable and $\delta_P(p)$ – the indicator function of the set $P = \{p \in \mathcal{Y} : \|p(x)\| \leq 1\}$. Using duality, we can write the final formulation of considering problem as

$$\min_{u \in \mathcal{X}} \max_{p \in \mathcal{Y}} \left[-\langle u, \operatorname{div} p\rangle_{\mathcal{X}} + \lambda/2\|u - g\|_2^2 - \delta_P(p)\right]. \quad (5)$$

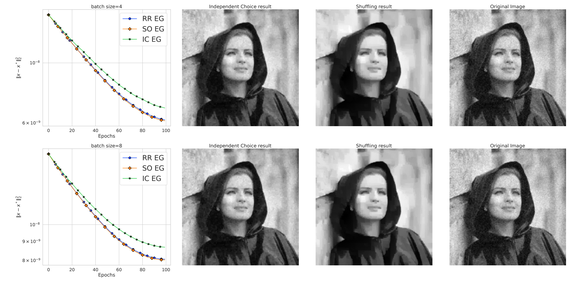


Figure: EXTRAGRADIENT on image with $\sigma = 0.05$ on the problem (5).

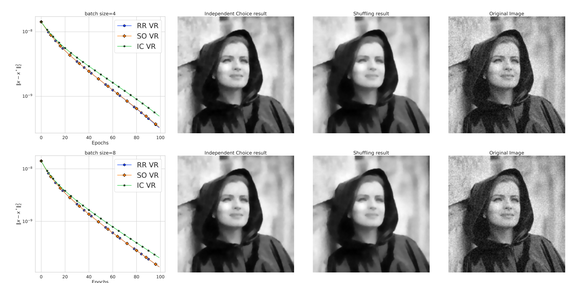


Figure: EXTRAGRADIENT with VR on image with $\sigma = 0.05$ on the problem (5).

Average-case optimization analysis for distributed consensus algorithms on regular graphs

Nhat Trung Nguyen **Alexander Rogozin** **Alexander Gasnikov**

Moscow Institute of Physics and Technology

Consensus Problem

Consider a network of agents represented by an undirected finite graph $G = (V, E)$, where $V = \{1, \dots, n\}$ represents the set of vertices (agents) and E represents the set of edges (communication links). Each agent i holds an initial vector $x_0^{(i)} \in \mathbb{R}^d$. We denote by $x_0 = \left(\left(x_0^{(1)}\right)^\top, \dots, \left(x_0^{(n)}\right)^\top\right)^\top$. The goal is to design efficient algorithms that allow each agent to quickly compute the average value $\overline{x}_0 = \frac{1}{n}\sum_{i=1}^n x_0^{(i)}$, with the constraint that at each iteration of the algorithm, agents can only exchange their vectors with their neighbors.

To achieve consensus on the graph G, we solve the following problem starting with the initial vector x_0:

$$\min_{x \in \mathbb{R}^{nd}} f(x) = \frac{1}{2} x^T \mathbf{L} x, \tag{1}$$

where $\mathbf{L} = L \otimes I_d$, the symbol $\otimes$ denotes the Kronecker product, and L is a *gossip matrix*, which is defined as follows

Definition 1. *A gossip matrix $L \in \mathbb{R}^{n \times n}$ on the graph $G = (V, E)$ is a matrix satisfying following properties:*

1. *L is an $n \times n$ symmetric matrix,*
2. *L is positive semi-definite,*
3. $\ker(L) = \mathrm{span}(\mathbf{1})$, *where* $\mathbf{1} = (1, \dots, 1)^\top$,
4. *L is defined on the edges of the network: $L_{ij} \neq 0$ only if $i = j$ or $(i, j) \in E$.*

Polynomial-Based Iterative Methods

We consider *first-order methods* or *gradient-based methods* to solve the problem (1). These are methods in which the sequence of iterates x_t is in the span of previous gradients, i.e.,

$$x_{t+1} \in x_0 + \mathrm{span}\{\nabla f(x_0), \dots, \nabla f(x_t)\}. \tag{2}$$

Lemma 1. *Let x_t be generated by a first-order method of kind (2). Then there exists a polynomial P_t of degree t such that $P_t(0) = 1$ and it verifies*

$$x_t - x_* = P_t(\mathbf{L})(x_0 - x_*) \tag{3}$$

The polynomial P_t is called the residual polynomial.

Average-case analysis

Definition 2. *Let L be a random matrix with eigenvalues $\{\lambda_1, \dots, \lambda_n\}$. The empirical spectral distribution of L is the probability measure*

$$\mu_L(\lambda) = \frac{1}{n}\sum_{i=1}^n \delta_{\lambda_i}(\lambda), \tag{4}$$

where δ_{λ_i} is the Dirac delta. Since L is random, the empirical spectral distribution μ_L is a random measure. Its expectation over L,

$$\mu = \mathbb{E}_L\left[\mu_L\right] \tag{5}$$

is called the expected spectral distribution

Theorem 1. *Let x_t be generated by a first-order method, associated to the polynomial P_t. Then we can decompose the expected error at iteration t as*

$$\mathbb{E}\|x_t - x_*\|^2 = R^2 \int P_t^2 d\mu. \tag{6}$$

Spectrum of regular graph

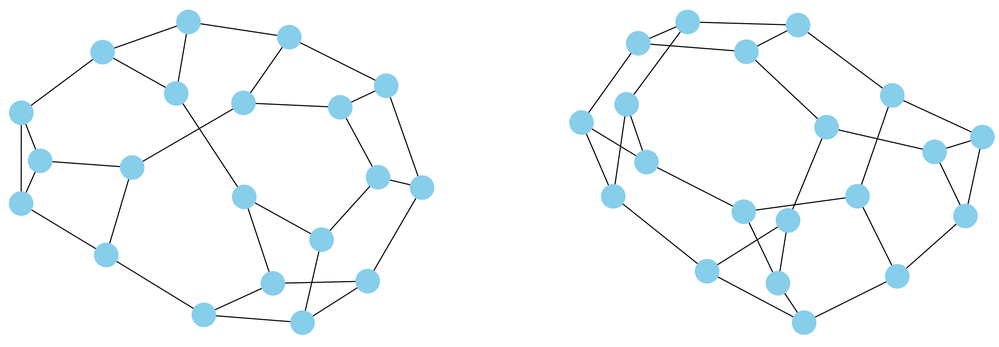


Figure 1: Regular graphs with $n = 20$, $k = 3$.

$$d\mu(\lambda) = \frac{k}{2\pi}\frac{\sqrt{\frac{4(k-1)}{k^2} - (1-\lambda)^2}}{1-(1-\lambda)^2}\, d\lambda \tag{7}$$

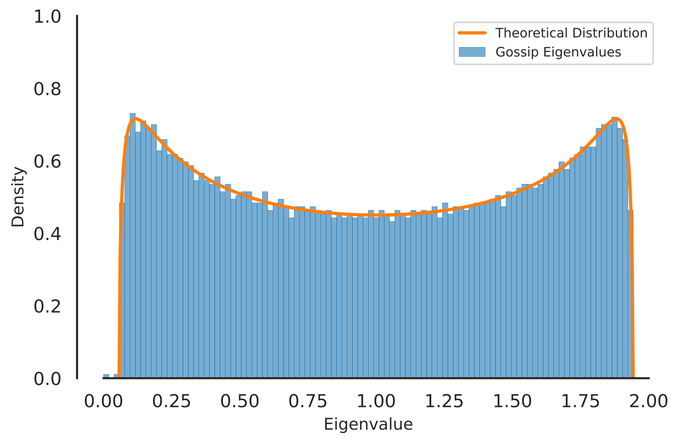


Figure 2: Spectrum of regular graph with $n = 5000$, $k = 3$.

Optimal method

Algorithm 1 Optimal average-case method for regular graphs

Input: starting guess x_0, regular parameter k, $\delta_0 = \frac{k}{k+1}$.
Initialize: $x_1 = x_0 - \delta_0 \cdot L x_0$
for $t = 1, 2, \dots$ **do**
  $\delta_t = \left(1 - \frac{k-1}{k^2} \cdot \delta_{t-1}\right)^{-1}$
  $x_{t+1} = x_t + (\delta_t - 1)(x_t - x_{t-1}) - \delta_t \cdot L x_t$
end for

Theorem 2. *If we apply Algorithm 1 to problem (1), where L is the gossip matrix of random k-regular graphs, then*

$$\mathbb{E}\|x_t - x_*\|^2 = \Theta\left(\left(\frac{1}{k-1}\right)^t \cdot \left(\frac{1}{1 + \frac{2}{k-2}\left(1 - \frac{1}{(k-1)^t}\right)}\right)^2\right). \tag{8}$$

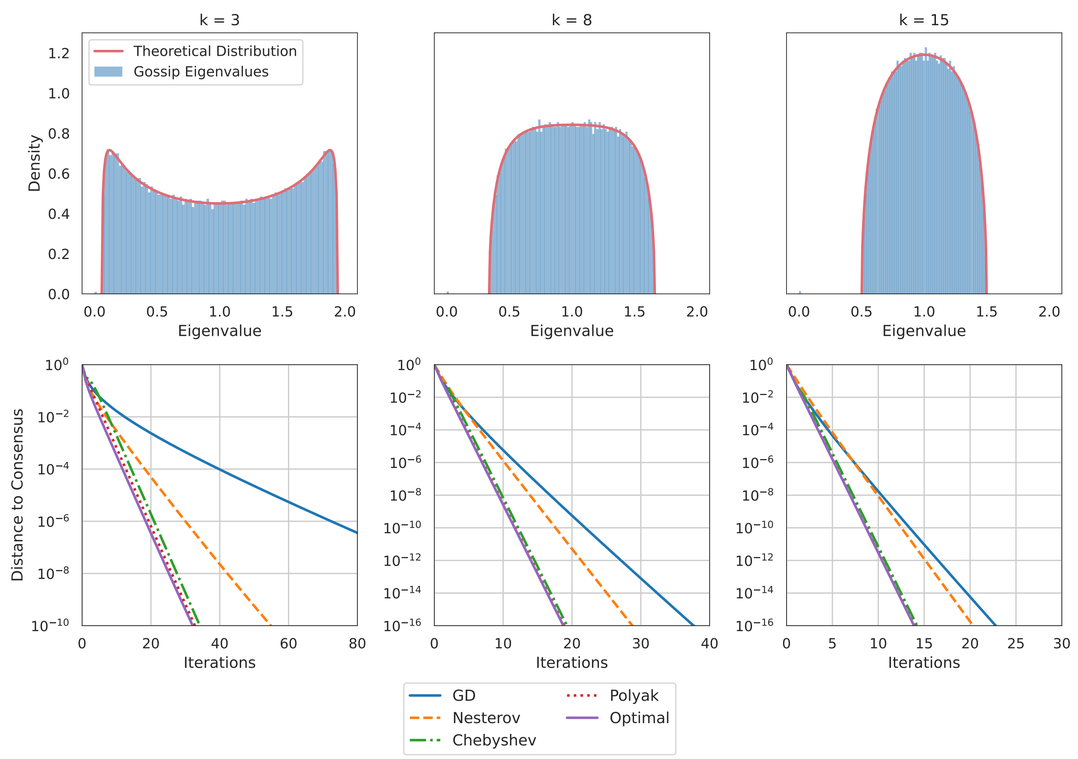


Figure 3: Comparison of convergence speeds of algorithms on regular graphs.

Application of gradient methods to solving ill-posed problems of mathematical physics

Nikita V. Pletnev
Moscow Institute of Physics and Technology

ASCOMP 2024

Ill-posed and inverse problems

- Various physical meaning and applications. Different type: differential or integral equations.
- A small change in the initial data leads to a significant change in the solution.
- Often ill-posed problems are inverse to well-posed and relatively easy-to-solve problems.
- The main idea is to replace condition f by condition q on another boundary.
- So we seek q making the well-posed problem equivalent to ill-posed.

Operator of the problem. Reduction to optimization

- Let $q : [0, 1] \to \mathbb{R}$ be an element of functional Hilbert space H with usual scalar product.
- We define the operator $A : H \to H$, which associates the corresponding f to a known q.
- Calculation of Aq is a well-posed (direct) problem.
- Each ill-posed Cauchy problem under consideration is reduced to the operator equation $Aq = f$.
- $Aq^* = f \Rightarrow q^* = \arg\min\limits_{q \in H} J(q)$. If $J(q^*) = 0$, q^* is the solution. Otherwise, solution does not exist.
- $J(q) = \frac{1}{2}||Aq - f||^2$ is convex and smooth functional. Its gradient can be calculated by general formula: $\nabla J(q) = A^*(Aq - f)$, A^* is a conjugate operator, similar (sometimes equal) to A, can be found by the method of Lagrange's multipliers.

Optimization methods: classic

- Gradient descent: $q_{n+1} = q_n - \alpha_n \nabla J(q_n)$. $\quad J(q_n) = O(1/n)$.
  Step: constant; $\alpha_n = \frac{||\nabla J(q_n)||^2}{||A_0 \nabla J(q_n)||^2}$ (fastest descent); $\alpha_n = \frac{2J(q_n)}{||\nabla J(q_n)||^2}$ (nearest descent).
- Accelerated methods: $q_{n+1} = q_n - \alpha_n \nabla J(q_n) + \beta_n (q_n - q_{n-1})$. $\quad J(q_n) = O(1/n^2)$.
  Conjugate Gradient Descent ($J(q_{n+1})$ is minimized on every step); Heavy Ball; STM.

New method: Conjugate Gradient Descent with minimization of distance to the exact solution

$$q_{n+1} = q_n - \alpha_n \nabla J(q_n) + \beta_n (q_n - q_{n-1}); \quad (\alpha_n, \beta_n) = \arg\min_{\alpha_n, \beta_n} ||q_{n+1} - q^*||^2.$$

It can be rewritten:

$$s_n = -\nabla J(q_n) + \frac{\langle \nabla J(q_n), s_{n-1} \rangle}{||s_{n-1}||^2} s_{n-1}; \quad q_{n+1} = q_n + \frac{2J(q_n)}{||s_n||^2} s_n.$$

- The nearby steps produced by this method are orthogonal: $\langle s_n, s_{n-1} \rangle = 0$.
- q_n converges to q^*, and the distance decreases monotonically.
- The functional converges to 0, but not monotonically.
- The method is appropriate only for quadratic functions, but it's OK: $J(q)$ is quadratic.
- $||q_{n+1} - q^*||^2 \le \left(1 - \frac{\mu}{L \sin^2 \varphi_n}\right) \cdot ||q_n - q^*||^2$, where φ_n is angle between $-\nabla J(q_n)$ and $q_n - q_{n-1}$.
- The method can be improved by using more steps in definition of s_n. The modification using *all* previous steps is the best first order method, but its computational complexity is $O(n^2)$, not $O(n)$.

Results are submitted at arXiv.org: «On the modification of the conjugate gradient method with minimization of the distance to the exact solution when choosing the step length».

Retrospective Cauchy problem for heat equation: parabolic type

$$\begin{cases} u_t - \kappa^2(x,t)\Delta_x u = 0, & (x,t) \in \Omega = \Pi \times (0,1) \\ u|_{x \in \partial\Pi} = 0 \text{ or } \frac{\partial u}{\partial \overline{n}}|_{x \in \partial\Pi} = 0, & t \in [0,1] \\ u|_{t=1} = f(x), & x \in \Pi \end{cases}$$

$\kappa(x,t)$, $f(x) = (Aq^*)(x)$ are known continuous functions.

$(Aq)(x) = u(x,1)$, where $q(x) \in C^2(\Pi)$, $u(x,t)$ is solution of

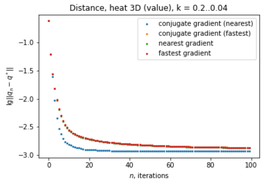
$$\begin{cases} u_t - \kappa^2(x,t)\Delta_x u = 0, & (x,t) \in \Omega = \Pi \times (0,1) \\ u|_{x \in \partial\Pi} = 0 \text{ or } \frac{\partial u}{\partial \overline{n}}|_{x \in \partial\Pi} = 0, & t \in [0,1] \\ u|_{t=0} = q(x), & x \in \Pi \end{cases}$$

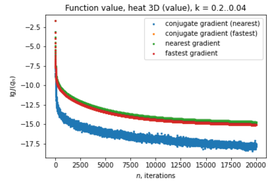
$$\kappa(x,t) \equiv const \Leftrightarrow A^* = A.$$

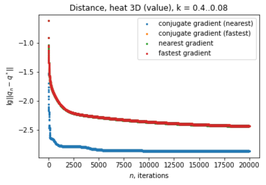
Experiments: boundary conditions on the value

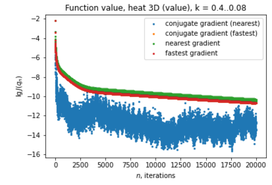
$f(x) = (Aq^*)(x)$, where $q^*(x) = \sin 2\pi x_1 \sin^2 2\pi x_2 \sin^3 2\pi x_3$,

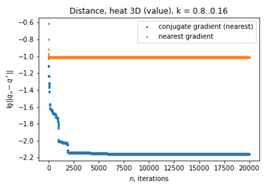
$$\kappa(x,t) = \begin{cases} \kappa_{max}, & 0.4 < x_i < 0.6,\ i = 1,2,3 \\ \frac{\kappa_{max}}{5}, & \text{otherwise} \end{cases}$$

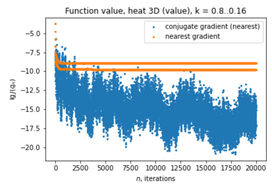












Achieved distance to minimum point (boundary conditions on value), $\times 10^{-3}$. Initial distance: 0.242.

κ_{max}	grad J	grad ρ	conj J	conj ρ
0.2	1.1757	1.1757	1.1757	1.1756
0.4	3.67	3.66	3.67	1.36
0.6	46.8	40.0	46.8	3.17
0.8	-	96.7	-	6.95
1.0	-	-	-	21.1
1.2	-	-	-	47.6

Fredholm integral equation of the first kind

$$\int_0^1 K(x,s)q(s)ds = f(x), \quad x \in [0,1]$$

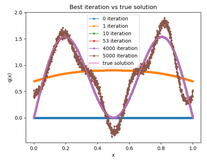
$K(x,s)$, $f(x) = (Aq^*)(x)$ are known continuous functions.

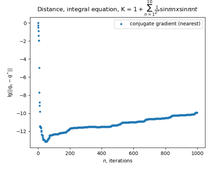
$$(Aq)(x) = \int_0^1 K(x,s)q(s)ds, \quad x \in [0,1];$$
$$K(x,s) = K(s,x) \Leftrightarrow A^* = A.$$

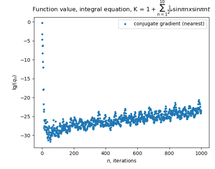
Experiments

Symmetric kernel: conjugate gradient method with minimization of distance to the exact solution

$K(x,s) = 1 + \sum\limits_{n=1}^{10} \frac{1}{2^n} \sin \pi n x \sin \pi n s$; $q^*(x) = \sin \pi x + \sin 3\pi x$.

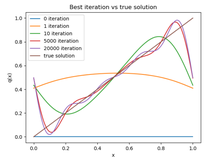


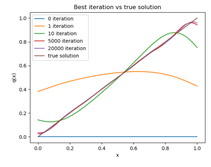


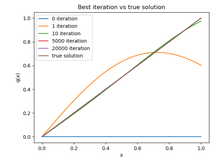


After quickly reaching the optimum (distance to the exact solution is $6.789 \cdot 10^{-14}$, functional value is $2.126 \cdot 10^{-32}$) on 53-th iteration the quality deteriorates.

Reconstruction of $q^*(x) = x$ for symmetric kernels: conjugate gradient method with minimization of functional







$K(x,s)$	$\|\|q - q^*\|\|_2$	$J(q)$
$1 + \sum\limits_{n=1}^{10} \frac{1}{2^n} \sin \pi n x \sin \pi n s$	0.104	$2.39 \cdot 10^{-10}$
$1 + \sum\limits_{n=1}^{10} \frac{1}{2^n} \cos \pi n (x - s)$	$6.79 \cdot 10^{-3}$	$8.08 \cdot 10^{-11}$
$\sin \pi x s$	$1.52 \cdot 10^{-4}$	$3.54 \cdot 10^{-17}$

Conclusions

- Conjugate gradient descent with minimization of the distance to the exact solution is quite simple and effective to solve inverse and ill-posed problems.
- Retrospective Cauchy problem for heat equation is as worse as κ is larger. New method can be used for $\kappa \lesssim 0.8$, unlike old methods (only for $\kappa \lesssim 0.5$).
- Integral equation is solved very good, but using of new method requires the stopping rule.

Path-based approach for Important Node Exploration

E. Sabitov In collaboration with

SBER AI Lab

Introduction

Our main goal is to provide people with good insights on technological areas, they have little knowledge about. We do that by obtaining graph of patents citations with respect to provided request and finding core nodes in it.

Approach: get clear data and utilize GNN architectures to highlight core nodes.

Challenge: there are no labels, validation can be done only with the help of specialists from the specific field.

Objectives

- Create pipeline with
  - Scrapping relevant patents from google patents
  - Enrichment of citation graph with more relevant nodes and edges
  - Pruning of non-relevant nodes with preservation of graph topology
- Research potential ways to highlight core nodes
- Validate the results

Process & Methods

1. Designed data load pipeline
2. Tested and incorporated relevance score estimation with modification of BERT model
3. Designed and implemented algorithm for graph enrichment.
4. Experimented with various versions of algorithm for finding core node
   - GNN for link prediction
   - Attention on paths
   - Attention on random walks
5. Validation on 5 technological domains
   - photolithography simulation
   - etch simulation
   - plasma discharge simulation
   - physical photoresist model
   - microscope image stitching

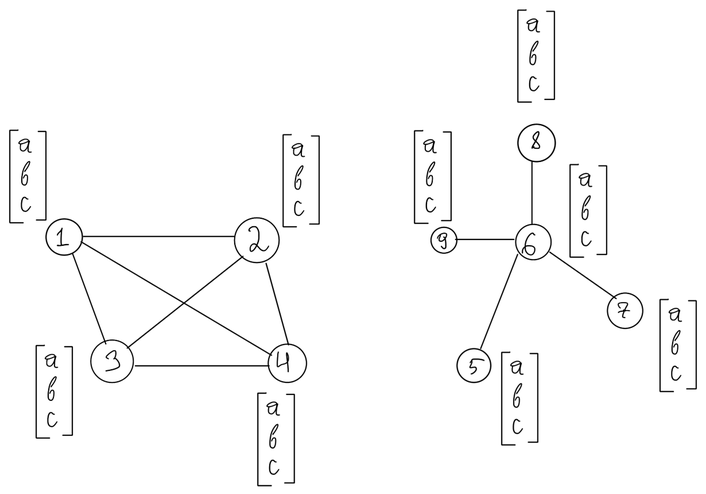


Fig.1 Graph with vector features

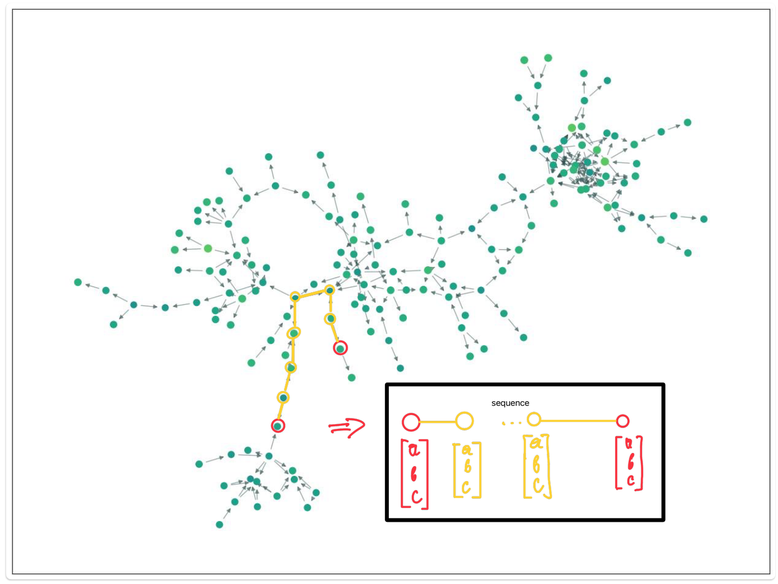


Fig.2 Attention on paths

Results

Repository for forming request to google patents and:

- Data preparation
- Enrichment of graph with
  - patent citations, obtained from merging duplicates
  - new patents, highly cited by patents from existing set
- PINE Algorithm for finding core nodes
- Validation from experts

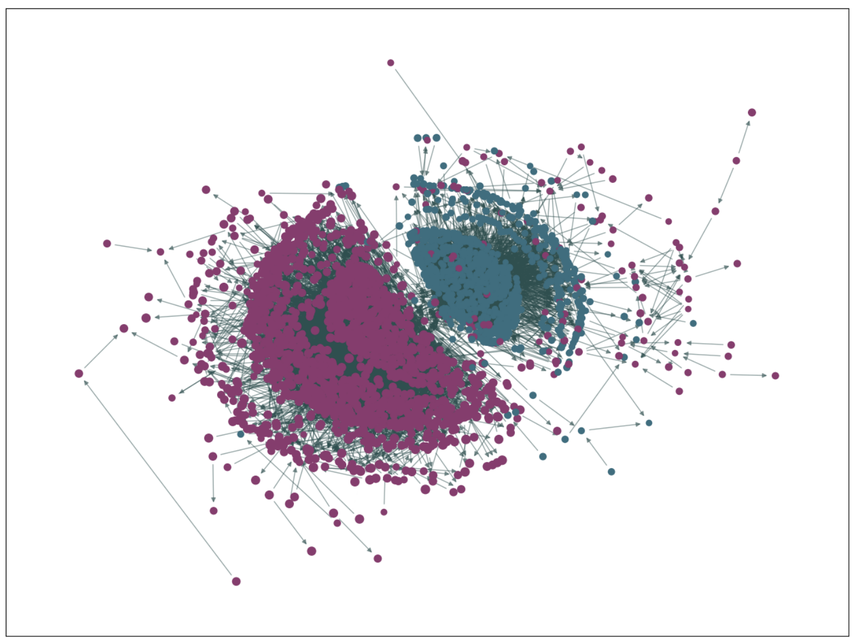


Fig.3 KMeans clusterization visualized on graph for lithography patents

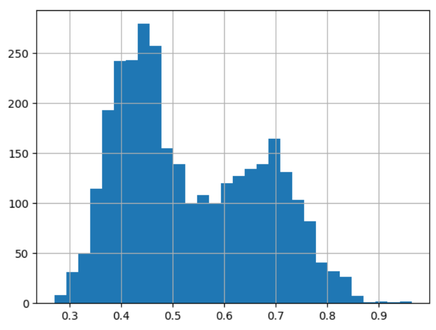


Fig.4 Relevance score distribution

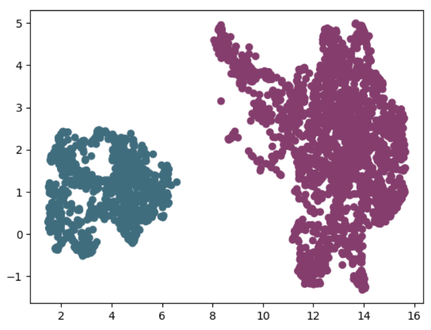


Fig.5 UMAP projection of title embeddings

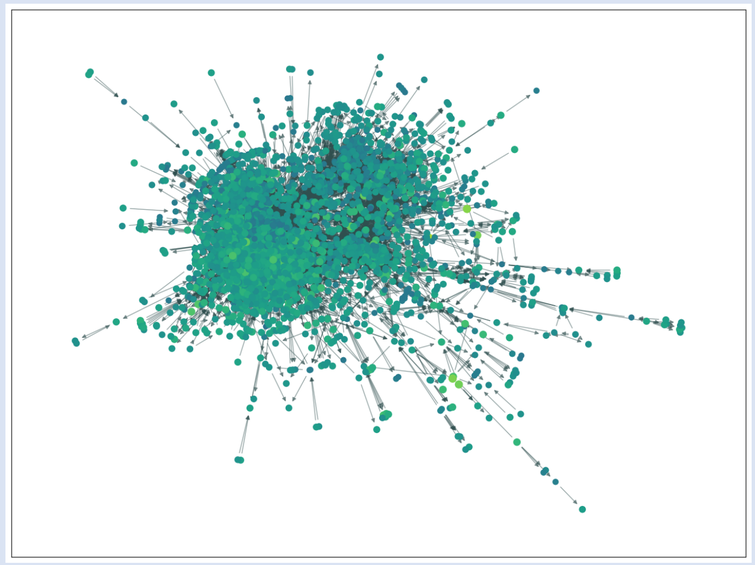


Fig.8 Citation graph for Etch simulation

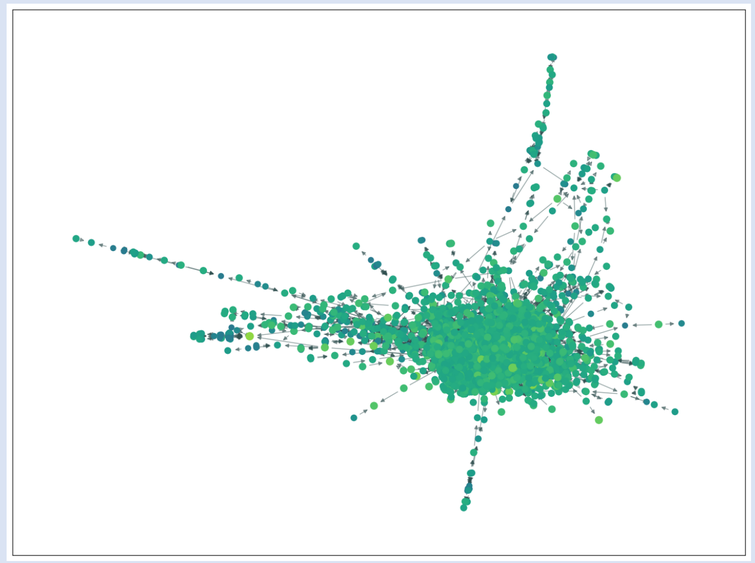


Fig.8 Citation graph for plasma discharge simulation

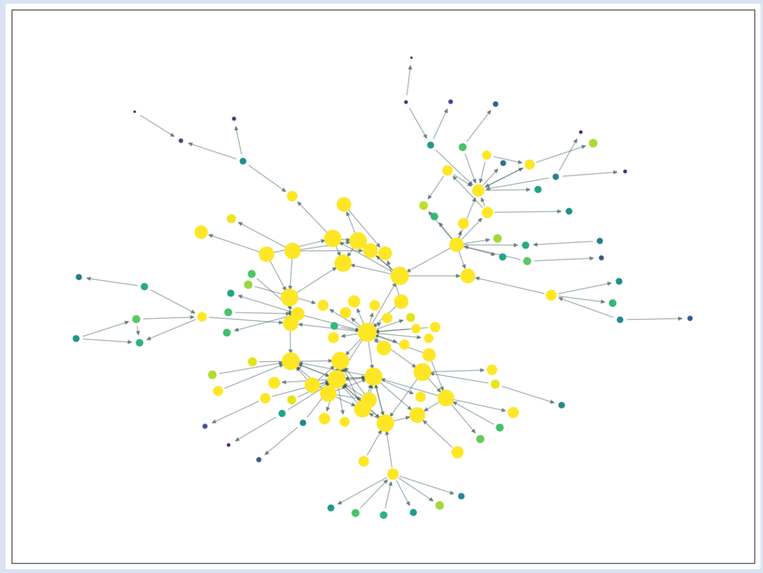


Fig.8 Plasma discharge model, resistance disctance on relevance scores

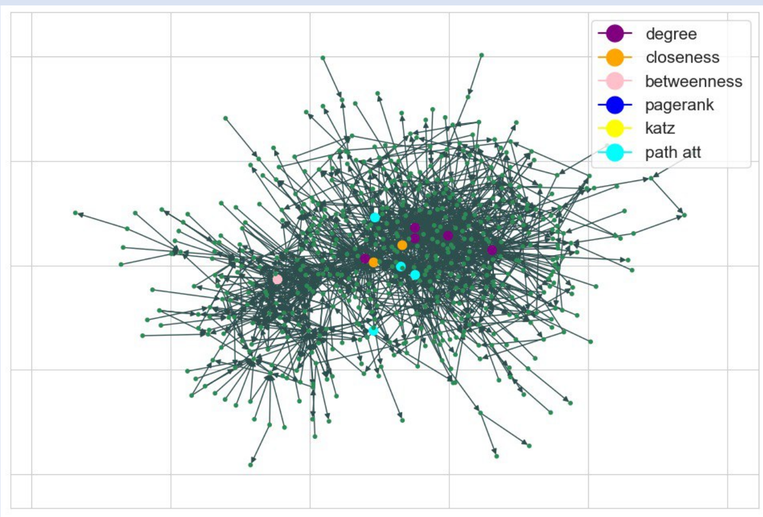


Fig.7 Etch modelling patents pruning

ASCOMP 2024

2024

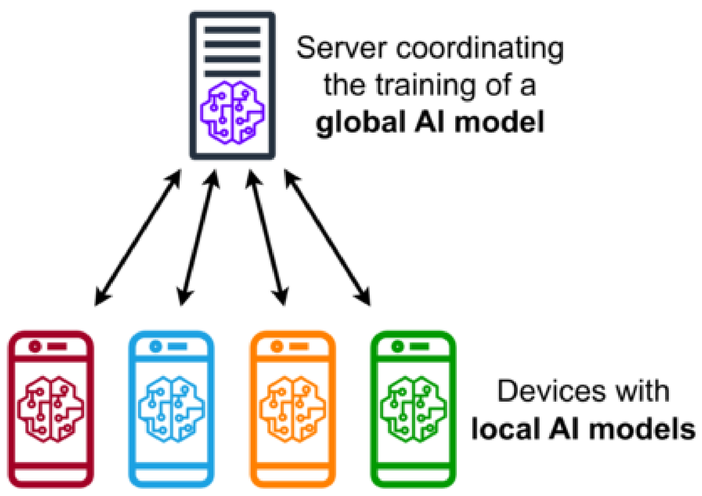
Skoltech

Local SGD converges **faster** for **quadratic-like** objectives and requires **less** communication.

Motivation and Challenges

- Larger models need data and tasks to be shared across many (M) devices.
- Devices calculate local stochastic gradients and transmit them to a central server.
- Transmitting large amounts of data is costly.
- Our goal: **Reduce** the number of **communication** rounds.

We denote the concept above as "Federated Learning"



Federated Learning process

Local SGD

- The most popular Federated Learning method is called **Local SGD**.
- It performs multiple local SGD steps between communications.
- Problem: if we reduce the number of communications and increase the number of local steps (H), the performance **degrades**.

Woodworth et al., 2020 noted the following: For **quadratic** objectives, Local SGD convergence rate **is not affected** by the number of local steps, making it **highly efficient** for such problems.

But what happens when we diverge from pure quadratic setting?

Thus, our aim was to establish better communication complexity rates for objectives somehow **close to the quadratic form.**

In order to measure the proximity of an objective F to the quadratic form, we decompose F into the sum: $F = Q + R$, where Q is a convex quadratic function, and R is some convex residue.

Then we introduce was *quadraticity* parameter $\varepsilon := \frac{L_R}{L} \leq 1$.

- For quadratic objectives, where F is equal to Q, **the value of ε is zero**
- For quadratic-like objectives, i.e. cases where Q is somewhat larger than R, ε is **small**.

Quadraticity concept allows us to improve over the previous lower bounds for Local SGD

Breaking existing bounds

Under the assumption of *uniformly bounded variance*, when $E\|\nabla F(x) - \nabla F(x,z)\|^2 \leq \sigma^2$:

Case $\mu = 0$, $E[F(x_T) - F(x_*)] =$

$$O\left(\frac{LD^2}{T} + \frac{\sigma D}{\sqrt{MT}} + \left(\frac{HL\sigma^2 D^4}{T^2}\right)^{1/3}\right) \quad [\text{Woodworth et al., 2020}]$$

$$\downarrow$$

$$O\left(\frac{LD^2}{T} + \frac{\sigma D}{\sqrt{MT}} + \left(\frac{\varepsilon HL\sigma^2 D^4}{T^2}\right)^{1/3}\right) \quad [\text{This work}]$$

If we denote $\lambda = \mu_Q + \mu_R$ we can also get an estimate for the case $\lambda > 0$:

$$\tilde{O}\left(\text{exp.decay} + \frac{\sigma^2}{\mu MT} + \frac{HL\sigma^2}{\mu^2 T^2}\right) \quad [\text{Woodworth et al., 2020}]$$

$$\downarrow$$

$$\tilde{O}\left(\text{exp.decay} + \frac{\sigma^2}{\lambda MT} + \frac{\varepsilon HL\sigma^2}{\lambda^2 T^2}\right) \quad [\text{This work}]$$

Abandoning restrictive assumption

If we replace uniformly bounded variance assumption with more **general** one, i.e.

$$E\|\nabla F(x) - \nabla F(x,z)\|^2 \leq \sigma^2 + \rho\|\nabla F(x)\|^2$$

the acceleration given by quadraticity **persists**.

Case $\lambda > 0$:

$$E[F(x_T) - F(x_*)] = O\left(\text{exp.decay} + \frac{\sigma^2}{\lambda MT} + \frac{\rho HL^2\sigma^2}{\lambda^3 MT^3} + \frac{\varepsilon HL\sigma^2}{\lambda^2 T^2}\right)$$

Variance reduction term

Represents the impact of $\rho\|\nabla F(x)\|^2$

Represents the ***drift*** caused by rare communication

In all the estimates above, the last term represents ***drift*** that appears due to many local steps (or rare communication, which is equivalent)

So, when it is multiplicated by the ε factor it shows that **the influence of rare communications weakens for quadratic functions.**

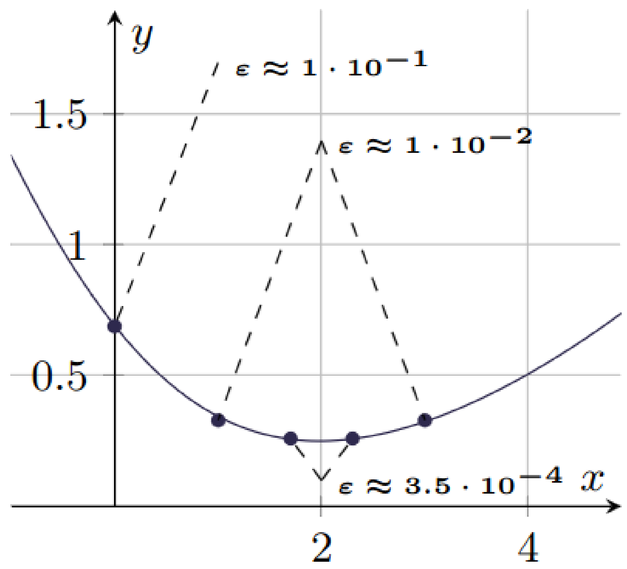
Notation

The following symbols and definitions are used throughout this work:

Symbol	Definition
M	Number of devices
H	Number of local SGD steps
T	Total number of iterations on a given device
D	Initial distance to the optimum, $\|x_0 - x_*\|$
μ	Strong convexity constant
L	Lipschitz gradient constant

Discussion

An important observation about quadraticity is that for functions with a **Lipschitz Hessian**, ε decreases rapidly, as illustrated in the graph below.



Decrease of ε for LogLoss with l_2 regularization

References

Woodworth, B., Patel, K. K., Stich, S. U., Dai, Z., Bullins, B., McMahan, H. B., Shamir, O., & Srebro, N. (2020). Is local sgd better than minibatch sgd?

Local SGD for Near-Quadratic Problems: Improving Convergence under Unconstrained Noise Conditions

Andrey Sadchikov, Savelii Chezhegov, Aleksandr Beznosikov and Alexander Gasnikov
Moscow Institute of Physics and Technology, Moscow, Russia

Automatic segmentation of epicardial fat and quantification of radiomic parameters in cardiac computed tomography

Samatov Denis[1] · Merzlikin Borisl[1] · Zavadovsky Konstantin[2]

[1]Department of Mathematics and Mathematical Physics , Tomsk Polytechnic University, Russia

[2]Department of X-ray and tomographic diagnostic methods, Research Institute of Cardiology of Tomsk National Research Medical Center, Russia

Motivation

The objective of this work is to develop a computational tool that automatically segments epicardial adipose tissue (EAT) and quantifies radiomic parameters, while also offering functionality for manual correction of pericardium tracking.

Cardiovascular Diseases

Cardiovascular diseases (CVDs) remain a global challenge. New methods based on deep learning, segmentation, and radiomics offer new perspectives for more accurate diagnosis and treatment.

Proposed Solution

Develop an architecture for automatic segmentation of epicardial adipose tissue (EAT) on cardiac CT images. Create a computational tool with automatic segmentation and manual pericardium correction.

1: Radiomics

Radiomics is one of the fastest growing areas of research in nuclear medicine, related to the extraction of quantitative metrics from medical images.

The radiomics workflow involves acquiring and enhancing medical images (CT, MRI) to ensure uniformity and quality, followed by the systematic extraction and analysis of quantitative features using statistical and machine learning methods for disease diagnosis.

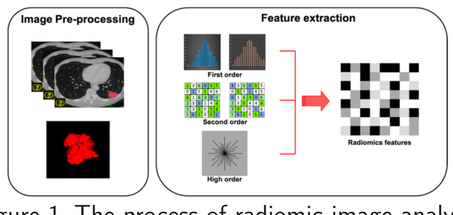


Figure 1. The process of radiomic image analysis

2: Architecture Segmentation Algorithm

The reported approach performs fully automatically without prior specialist input. After automatic segmentation, specialists can make necessary adjustments and re-run the algorithm.

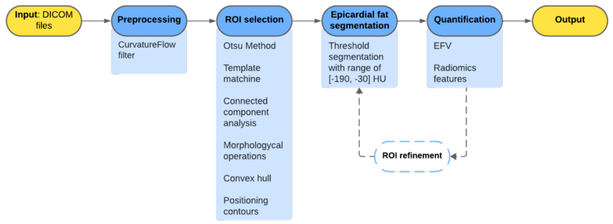


Figure 2. Architecture of the proposed algorithm

3: App

The graphical interface of this tool is developed as a desktop application for the Windows operating system. The implementation is done in the Python programming language using the PyQt5 library.

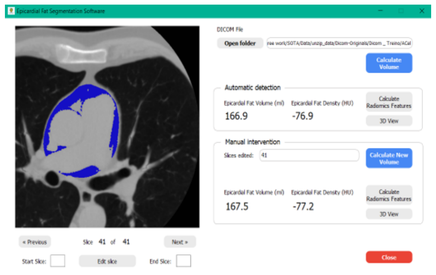


Figure 3. User interface of the application

The tool allows you to perform the following operations

- Manual editing of the pericardium.
- Calculation of radiomic indicators of the studied ROI.

4: Results

To perform statistical analysis, images automatically segmented using the described method were compared with manually segmented images (considered reference), slice by slice.

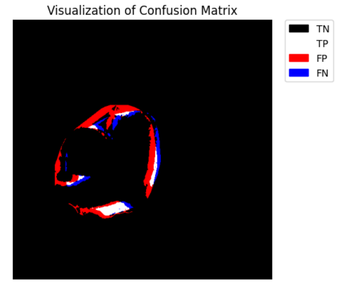


Figure 4. Evaluation image obtained by comparing manual and automatic segmentation

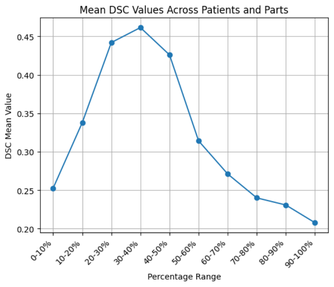


Figure 5. Local DSC from lower to upper heart

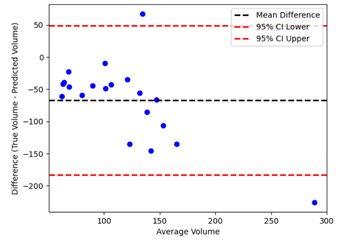


Figure 6. Bland-Altman plot of EFV measurements, with 95% confidence interval for optimized $\mathrm{EFV_a}$ and manually derived $\mathrm{EFV_m}$

By comparing manual and automatic segmentation on a pixel-by-pixel basis on 878 images, an average accuracy of **0.95** was obtained.

Conclusion

(a) An architecture for automatic segmentation of epicardial fat in cardiac CT images has been developed.

(b) The processing speed of cardiac CT images is $\mathbf{22.3 \pm 1.7}$ seconds.

(c) A tool has been created for automatic segmentation of cardiac EAT from CT images, calculation of radiomic parameters of ROI, and manual pericardium correction.

References

[1] Rebelo, A. F., Ferreira, A. M., & Fonseca, J. M. (2022). Automatic segmentation of epicardial fat and volume quantification in non-contrast cardiac CT. *Update of Computer Methods and Programs in Biomedicine, 2*, 100079.

[2] Visual Lab. (2014). *A computed tomography cardiac dataset*. Retrieved February 15, 2024, from `http://visual.ic.uff.br/en/cardio/ctfat/`

Metropolis-Hastings with Approximate Acceptance Ratio Calculation

Yury Svirschevsky[1]*, Sergey Samsonov[1]
[1]HDI Lab, HSE
*ysvirshchevskii@hse.ru

Approximate Metropolis-Hastings

Algorithm 1 Adaptive Approximate Metropolis-Hastings Step

Input: Target density $\pi(x)$, Generative model $\mathcal{M}$
Output: Samples $Y_{1:n}$ approximating $\pi(x)$, Improved model $\mathcal{M}'$

$\hat{p}_{\mathcal{M}} \leftarrow$ Marginal likelihood estimator for $\mathcal{M}$
$X_{1:n} \leftarrow$ Draw n i.i.d. samples from $\mathcal{M}$
$Y_0 \leftarrow X_0$
for i=1 **to** n **do**
Compute acceptance probability

$$\alpha(Y_{i-1}, X_i) \leftarrow \frac{\pi(X_i)\hat{p}_{\mathcal{M}}(Y_{i-1})}{\pi(Y_{i-1})\hat{p}_{\mathcal{M}}(X_i)} \wedge 1$$

Get next sample

$$Y_i \leftarrow \begin{cases} X_i & \text{with probability } \alpha(Y_{i-1}, X_i), \\ Y_{i-1} & \text{with probability } 1 - \alpha(Y_{i-1}, X_i) \end{cases}$$

end for
$\mathcal{M}' \leftarrow$ Use $Y_{1:n}$ to train new model / fine-tune $\mathcal{M}$

- Generalization of the Metropolis-Hastings Algorithm, a popular MCMC method
- Generative model with **intractable marginal likelihood** used to model the proposal distribution
- **Estimate** of the model's marginal likelihood used in acceptance probability calculations instead of the exact value

Sample Quality is Improved

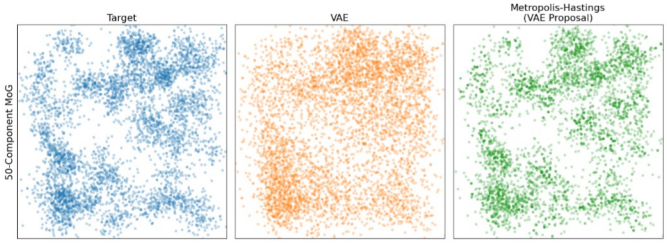


Figure 1: Demonstration on a 2D Mixture-of-Gaussians Target

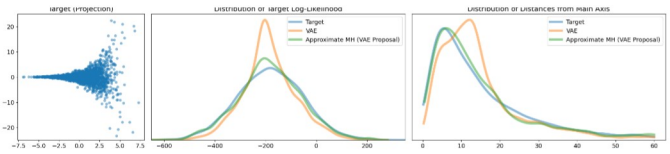


Figure 2: Approximate Metropolis-Hastings Improves Feature Distributions for a 128D Funnel

Marginal Likelihood Estimation

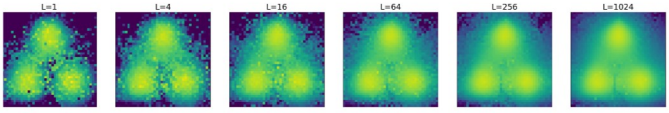


Figure 3: Importance Weighted Likelihood Estimates

The marginal likelihood of a **Variational Autoencoder** is intractable, but can be approximated. Let x and z denote the observed and latent variables respectively, $p_\theta(x, z)$ denote the joint distribution, and $q_\phi(z|x)$ denote the posterior approximation.

Importance Weighted estimator:

$$\hat{p}^{\text{IW}}(x) = \frac{1}{L}\sum_{i=1}^{L} \frac{p_\theta(x, Z^i)}{q_\phi(Z^i|x)}, \quad Z^1, \ldots, Z^L \overset{\text{iid}}{\sim} q_\phi(\cdot|x)$$

Sequential Importance Sampling estimator:

$$\hat{p}^{\text{SIS}}(x) = \frac{1}{L}\sum_{i=1}^{L} \frac{p_\theta^K(x, Z_{0:K}^i)}{q_\phi^K(Z_{0:K}^i|x)}, \quad Z^1, \ldots, Z^L \overset{\text{iid}}{\sim} q_\phi^K(\cdot|x)$$

$$p_\theta^K(x, Z_{0:K}) = p_\theta(x, Z_k) \prod_{k=0}^{K-1} l_k(z_{k+1}, z_k)$$

$$q_\phi^K(Z_{0:K}^i|x) = q_\phi(Z_k|x) \prod_{k=1}^{K} m_k(z_{k-1}, z_k)$$

where m_k and l_k are densities of forward and reverse Markov kernels. Different choices of kernels lead to different estimators.

Comparison with Classic Metropolis-Hastings

While Normalizing Flows have tractable marginal likelihoods, which allows exact acceptance probability calculation when using them as proposals, they are less flexible than VAEs.

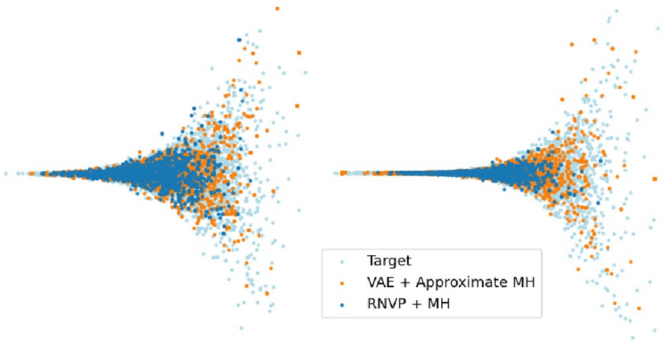


Figure 4: VAE-Proposal Approximate Metropolis-Hastings vs. Flow-Proposal Classic Metropolis-Hastings

MULTIMODAL APPROACHES IN STOCK PRICE PREDICTION

SIRIUS UNIVERSITY

KASYMKHAN KHUBIYEV
kasymkhankhubievnis@gmail.com
SCIENTIFIC CENTER FOR INFORMATION TECHNOLOGIES AND ARTIFICIAL INTELLEGENCE

LEVERAGING DIVERSE DATA FOR MORE ACCURATE STOCK PRICE PREDICTION

INTRODUCTION

Traditional stock price prediction relies heavily on numerical data, such as historical prices and trading volumes, limiting accuracy. Multimodal approaches, which combine data from diverse sources like text (news, social media) and numerical data, offer a more comprehensive analysis and improved predictive power.

Recent studies have shown the benefits of these methods: some used sentiment scores alongside price data [1], others examined the effect of news categorization on forecasts [2], and the latest research applied ChatGPT for news-based prediction [3]. Our research integrates raw text and numerical data to enhance stock price forecasting quality.

METHODOLOGY

MOEX API offered historical prices and trading volumes of companies included in IMOEX indices. We parsed news articles published in federal and social media within two years starting from July 7, 2022,

First, we trained machine learning models solely on numerical historical data. Then, we added sentiment analysis. Finally, we combined text and numerical data using embeddings and built a stock price prediction.

We also developed trading strategies based on the machine learning model predictions. As a baseline for comparison, we used the naive "Buy&Hold" strategy, where stocks were purchased at the beginning of the trading period and held in the same quantity until the end. The price prediction quality was evaluated using the MAPE metric, while the trading strategy performance was measured by the portfolio Sharpe ratio and turnover.

RESULTS

- Trading strategy based on Random Forest model with sentiment score data performed better than classic mean-reversion (mean_rev_alpha) and reverse (rev_alpha) algorithmic strategies.
- Integrating raw text data embeddings alongside historical returns data enhenced forecasting accuracy dramatically outperforming “buy&hold” strategy

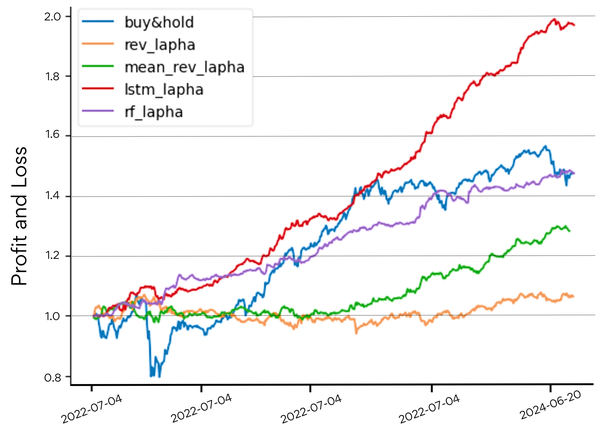


Figure 1. Profit and Loss graphs of trading strategies based on historical returns and news flow and normalized to initial money account

Strategy	Sharpe ratio	Turnover
Buy&Hold	1.71	0.00
Mean Reversion	2.53	1.36
Random Forest	4.49	1.33
LSTM	7.69	1.35

Table 1. Trading strategies cumulative profit and loss Sharpe ratio and turnover

CONCLUSION

The trading strategy based on multimodal forecasting demonstrated advantages over previous approaches. Although LLMs application remains unrevealed the current paper clames multimodal approaches open new doors for financial analysis and market forecasting, providing deeper insights and more accurate predictions.

This work was supported by the grant of the state program of the «Sirius» Federal Territory «Scientific and technological development of the «Sirius» Federal Territory».

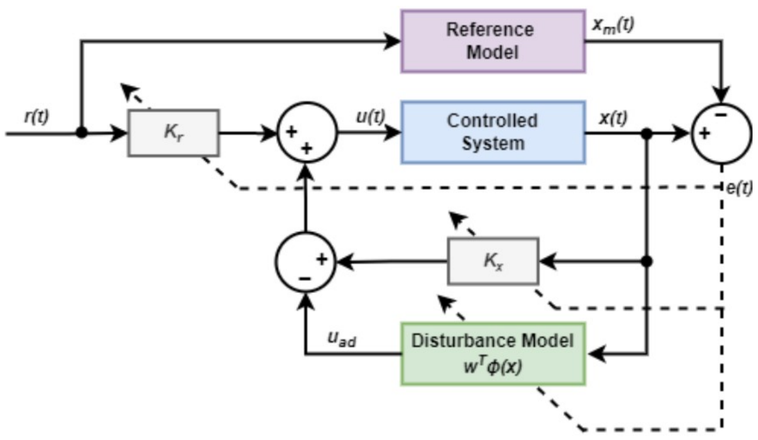
[1] FAZLIJA, BLEDAR AND HARDER, PEDRO. USING FINANCIAL NEWS SENTIMENT FOR STOCK PRICE DIRECTION PREDICTION. MATHEMATICS. 2022. V. 10. N. 13.
[2] T.D.KULIKOVA, E.Y.KOVTUN, AND S.A.BUDENNY. DO WE BENEFIT FROM THE CATEGORIZATION OF THE NEWS FLOW IN STOCK PRICE PREDICTION PROBLEM? RUSSIAN ACADEMY OF SCIENCE REPORTS. MATHEMATICS, COMPUTER SCIENCE, CONTROL PROCESSES. 2023. V. 514. N. 2. P. 385-394.
[3] ALEJANDRO LOPEZ-LIRA AND YUEHUA TANG. CAN CHATGPT FORECAST STOCK PRICE MOVEMENT? RETURN PREDICTABILITY AND LARGE LANGUAGE MODELS. SSRN. 2023.

Longitudinal Movement Adaptive Control

Ilia Chichkanov (Skotech)

Annotaion

Model Reference Adaptive Control (MRAC) computes control actions to enable an uncertain controlled system to follow the behavior of a reference model. This work explores the application of this method for longitudinal speed tracking in automonous vehicles. The core idea of the method involves adding an adaptive component to the control strategy, with parameters that are adjusted through feedback. Various methods for constructing the adaptive part of the control are proposed, including the use of specialized tables. This approach aims to enhance the performance and robustness of vehicle speed control in uncertain conditions



MRAC Theory

uncertainty — assumption

$$\dot{x}(t) = Ax(t) + B\Lambda(u + \delta(x(t)))$$

$$\begin{cases} \delta(x) = w^T\theta_0(x) \\ w \in R^{s\times m} \text{ - unknown constant weights matrix} \\ \theta_0(x) \text{ - known basis function} \end{cases}$$

$x(t) \in R^n$ - state vector

$u(t) \in R^m$ - control command

$A \in R^{n\times n}$ - known systems matrix

$B \in R^{n\times m}$ - known input matrix

$0<\Lambda \in R^1$ - control effort uncertainty gain

$\delta(x(t)) : R^n \to R^m$

Goal is to find such control law $u = u(.)$
that controllable output $y = Cx \quad C \in R^{1\times n}$
follows along some reference signal $r = r(t) \in R^1$

$e = x - r, e \to 0$

Nominal System

$$\dot{x}(t) = Ax(t) + Bu$$

Nominal Control

$$u_n = -Kx + K_{ff}r$$

where $K \in R^{N\times 1}$ is the feedback control matrix and
$K_{ff} \in R^1$ is the feedforward gain.

Control law

$$\begin{cases} u = u_n + u_a \\ u_a = \hat{w}^\top\theta(x) = \hat{w}^\top\begin{pmatrix}\theta_0(x)\\ u_n\end{pmatrix} \\ \dot{\hat{w}} = -\gamma\theta(x)B^\top Pe \end{cases}$$

$V(x)$ - Lyapunov function

$$V(x) = e^\top Pe + tr\left[\left(\hat{w}\Lambda^{\frac{1}{2}}\right)^\top\left(\hat{w}\Lambda^{\frac{1}{2}}\right)\right]$$

$$\dot{V} = -e^\top Re \le 0$$

Forum on Robotics & Control Engineering (FoRCE, http://force.eng.usf.edu/) Seminar Series: "Model Reference Adaptive Control Fundamentals" (Dr. Tansel Yucelen)

Car longitudinal motion model

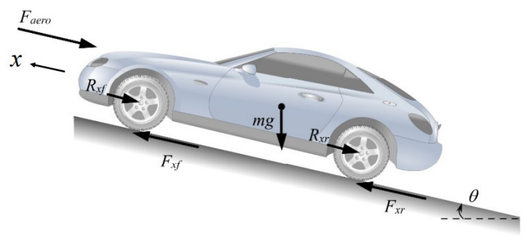
$$\begin{cases} m\dot{v} = F_x - F_{aero} - R_x - mg\sin\phi \\ J\dot{\omega} = T_{acc}(u>0) + T_{dec}(u<0) - F_x r_{eff} \end{cases}$$

$F_x = C_\sigma\sigma$ - longitudinal tire force

$\sigma = \dfrac{r_{eff}w}{v} - 1$ - slip ratio

$\dot{T}_{acc} = f_{acc}(u),\ u>0$ - torque transmitted from acceleration system

$\dot{T}_{dec} = f_{dec}(u),\ u<0$ - torque transmitted from braking system



$\dot{v} = u - \hat{F}_{aero} - \hat{R}_x - g\sin\phi$ - Nominal system

$$u = \underbrace{g\sin\phi + \dot{v}_{ref}}_{\text{FeedForward}} + \underbrace{\hat{F}_{aero} + \hat{R}_x + k_p(v - v_{ref})}_{\text{FeedBack}} \quad \to \quad \dot{e} = -k_p e \quad \to \quad e \to 0$$

Control law

$$\begin{cases} u = \underbrace{g\sin\phi + acc_{ref} + k_p(v - v_{ref})}_{u_n} + \underbrace{\hat{w}^T\theta(v,\ u_n)}_{u_a} \\ \dot{\hat{w}} = -\gamma\theta(v,\ u_n)e \end{cases}$$

Uncertanty compensation

$$\theta(v,u_n) = \begin{pmatrix} 1 \\ u_n[u_n>0] \\ u_n[u_n<0] \\ v/v_{max} \\ 1 - 2(v/v_{max})^2 \dots \end{pmatrix}^T$$

Acceleration error compensation

Deceleration error compensation

Chebyshev polinoms

$$u_a = \begin{pmatrix} w^1 \\ w^2 \\ w^3 \\ \dots \end{pmatrix}^T \begin{pmatrix} 1 \\ u_n[u_n>0] \\ u_n[u_n<0] \\ \dots \end{pmatrix}$$

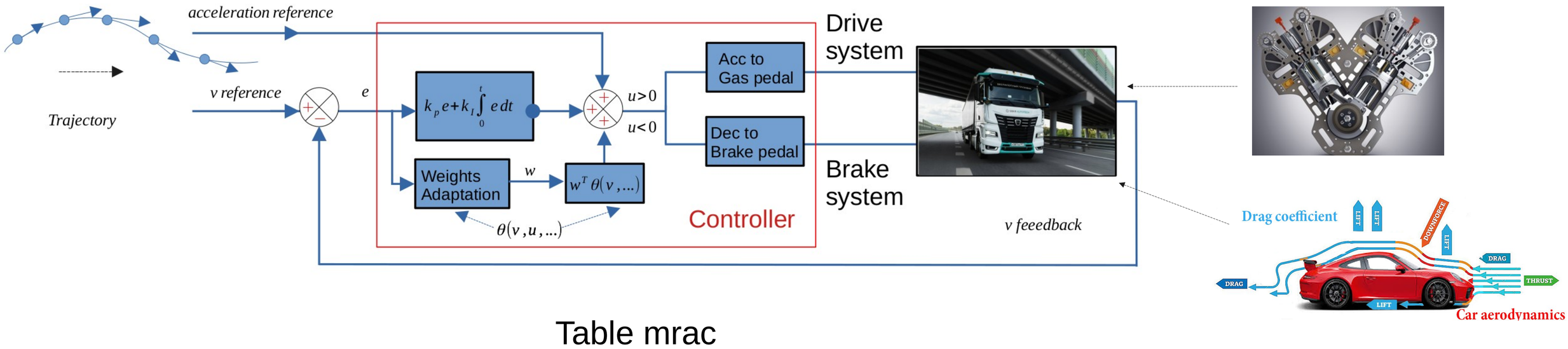


Table mrac

One dimentional table

$$f(V,v) = \begin{bmatrix} W \in R^m, V \in R^m \\ 0 \le i < m \\ \forall i : v^i \in V, v^i < v < v^{i+1} \end{bmatrix} = \begin{pmatrix} w^{i+1} \\ w^i \end{pmatrix}^T \underbrace{\begin{pmatrix} v^{i+1} - v \\ v - v^i \end{pmatrix} \frac{1}{(v^{i+1} - v^i)}}_{\theta(v,u_n)}$$

$$u_a = f(W^1,V,v) + f(W^2,V,v)u_n$$

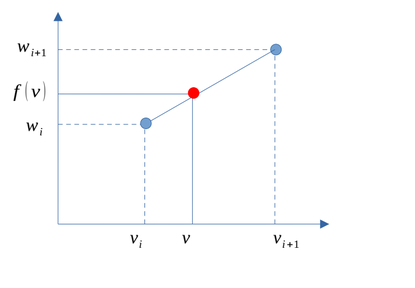
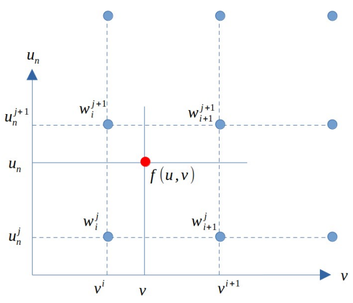


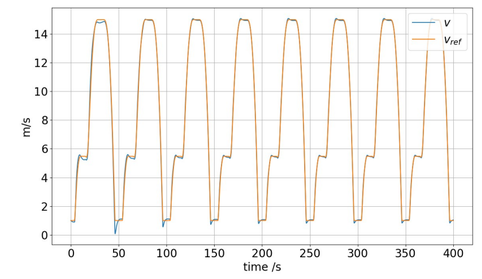
Table adaptive control relies on linear or bilinear interpolation from tables, with coefficients that are adjusted adaptively according to MRAC control law. The following graphs illustrate the simulation of vehicle speed tracking, taking into account the effects of tire characteristics and the nonlinear behavior of the engine. The method is based on the assumption that the system's characteristics vary at different speeds and with different levels of throttle and brake pedal engagement.

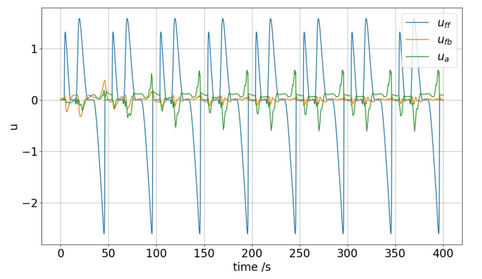
One dimentional table

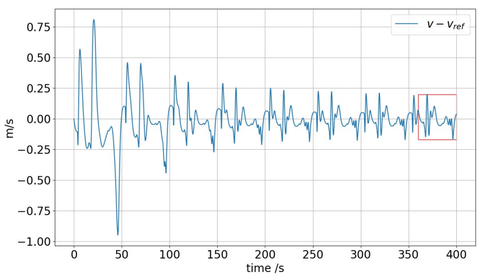
$$f(W,V,v) = \begin{bmatrix} W \in R^{m\times k}, \\ V \in R^k, U \in R^m \\ 0 \le i < m, 0 \le j < k \\ \forall i,j : v^i \in V, u_n^j \in V, \\ v^i < v < v^{i+1}, \\ u^i < u_n < u^{i+1} \end{bmatrix} = \begin{pmatrix} w_i^j \\ w_{i+1}^j \\ w_i^{j+1} \\ w_{i+1}^{j+1} \end{pmatrix}^T \underbrace{\begin{pmatrix} (v^{i+1} - v)(u^{j+1} - u_n) \\ (v - v^i)(u^{j+1} - u_n) \\ (v^{i+1} - v)(u_n - u^j) \\ (v - v^i)(u_n - u^j) \end{pmatrix} \frac{1}{((v^{i+1} - v^i)(u^{j+1} - u^j))}}_{\theta(v,u_n)}$$

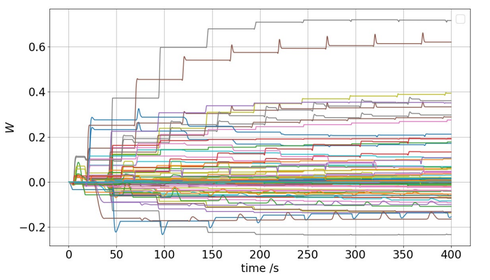
$$u_a = f([W^1,V,U],v,u_n) + f([W^1,V,U],v,u_n)u_n$$











Decentralized Optimization with Coupled Constraints

Demyan Yarmoshik [1, 2] Dmitry Kovalev [3] Alexander Rogozin [1, 4] Nikita Kiselev[1] Daniil Dorin[1] Alexander Gasnikov[5,1,2]

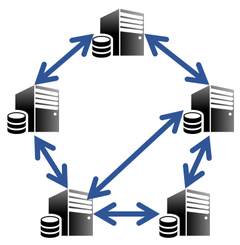
[1] Moscow Institute of Physics and Technology [2] Institute for Information Transmission Problems [3] Yandex [4] Skoltech [5] Innopolis University

The problem

We consider the decentralized optimization problem with coupled constraints

$$\min_{x_1\in\mathbb{R}^{d_1},\dots,x_n\in\mathbb{R}^{d_n}} \sum_{i=1}^n f_i(x_i)$$

$$\text{s.t. } \sum_{i=1}^n (\mathbf{A}_i x_i - b_i) = 0$$



Decentralized Topology

Function f_i, matrix $\mathbf{A}_i$ and vector b_i is a private information stored on i-th agent. Agents communicate only with their immediate neighbours in the communication network.

Our goal: obtain a linearly convergent first-order algorithm

Applications

• **Optimal exchange / Resource allocation**

$$\min_{x_1,\dots,x_n\in X} \sum_{i=1}^n f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^n x_i = b,$$

where $x_i \in X$ represents the quantities of commodities exchanged among the agents of the system, and $b \in X$ represents the shared budget or demand for each commodity.

• **Problems on graphs.** In electrical microgrids, telecommunication networks, drone swarms, etc, distributed systems are based on physical networks. Electric power network example: let $x_i \in \mathbb{R}^2$ denote the voltage phase angle and the magnitude at i-th electric node, let s be the vector of (active and reactive) power flows for each pair of adjacent electric nodes. Power flows can be derived (with high accuracy) from bus voltages using a linearization of Kirchhoff's law $\sum_{i=1}^n \mathbf{A}_i x_i = s$.

• **Consensus optimization.** Widely used in decentralized machine learning

$$\min_{x_1,\dots,x_n\in X} \sum_{i=1}^n f_i(x_i) \quad \text{s.t.} \quad x_1 = x_2 = \dots = x_n.$$

The consensus constraint can be reformulated in a decentralized-friendly manner as $\sum_{i=1}^n \mathbf{W}_i x_i = 0$, where $\mathbf{W}_i$ is the i-th vertical block of a gossip matrix (e.g., communication graph's Laplacian).

• **Vertical federated learning (VFL).** Let $\mathbf{F}$ be the matrix of features, split vertically (by features) between agents into submatrices $\mathbf{F}_i$.

$$\min_{\substack{z\in Y \\ x_1\in\mathbb{R}^{d_1},\dots,x_n\in\mathbb{R}^{d_n}}} \ell(z,l) + \sum_{i=1}^n r_i(x_i) \text{ s.t. } \sum_{i=1}^n \mathbf{F}_i x_i = z,$$

l is a vector of labels, x_i is a subvector of model parameters owned by the i-th node, ℓ is a loss function, r_i are regularizers.

Assumptions

• All f_i are μ_f-strongly convex and L_f-smooth; $\kappa_f := \frac{L_f}{\mu_f}$.

• The constraints are compatible. There exist constants $L_{\mathbf{A}} \geq \mu_{\mathbf{A}} > 0$, such that the constraint matrices $\mathbf{A}_1, \dots, \mathbf{A}_n$ satisfy $\sigma^2_{\max}(\mathbf{A}) = \max_{i\in 1\dots n} \sigma^2_{\max}(\mathbf{A}_i) \leq L_{\mathbf{A}}$, and $\mu_{\mathbf{A}} \leq \lambda_{\min^+}(\mathbf{S})$, where $\mathbf{S} = \frac{1}{n}\sum_{i=1}^n \mathbf{A}_i \mathbf{A}_i^\top$; $\kappa_{\mathbf{A}} := L_{\mathbf{A}}/\mu_{\mathbf{A}}$.

• We are given a gossip matrix W, such that:

1. $W_{ij} \neq 0$ if and only if agents i and j are neighbours or $i = j$.
2. $Wy = 0$ if and only if $y \in \mathcal{L}_1$, i.e. $y_1 = \dots = y_n$.
3. There exist constants $L_{\mathbf{W}} \geq \mu_{\mathbf{W}} > 0$ such that $\mu_{\mathbf{W}} \leq \lambda^2_{\min^+}(W)$ and $\lambda^2_{\max}(W) \leq L_{\mathbf{W}}$; $\kappa_{\mathbf{W}} := \frac{\lambda_{\max}(\mathbf{W})}{\lambda_{\min^+}(\mathbf{W})} = \sqrt{\frac{L_{\mathbf{W}}}{\mu_{\mathbf{W}}}}$.

Approach

Decentralized reformulation. Let $\mathbf{A} = \operatorname{diag}(A_1, \dots, A_n)$, $\mathbf{b} = (b_1^\top, \dots, b_n^\top)^\top$, $x = (x_1^\top, \dots, x_n^\top)^\top$, $\mathbf{W} = W \otimes I_m$. The original constraint can be equivalently reformulated as $\mathbf{A}x + \gamma \mathbf{W} y = \mathbf{b}$, $\gamma \neq 0$. Matrix multiplications in the reformulation can be performed using single communication with neighbours.

Base algorithm. We use algorithm from [1] (see also [2]), which was proposed for minimization of a smooth strongly convex function $G(u)$ under affine constraint $\mathbf{K}u = \mathbf{b}'$.

Algorithm 1: APAPC

$u_g^k := \tau u^k + (1-\tau) u_f^k$

$u^{k+\frac{1}{2}} := (1+\eta\alpha)^{-1}(u^k - \eta(\nabla G(u_g^k) - \alpha u_g^k + z^k))$

$z^{k+1} := z^k + \theta \mathbf{K}^\top (\mathbf{K} u^{k+\frac{1}{2}} - \mathbf{b}')$

$u^{k+1} := (1+\eta\alpha)^{-1}(u^k - \eta(\nabla G(u_g^k) - \alpha u_g^k + z^{k+1}))$

$u_f^{k+1} := u_g^k + \frac{2\tau}{2-\tau}(u^{k+1} - u^k)$

This first-order algorithm is based on the Forward-Backward algorithm and Nesterov's acceleration.

Augmentation. In the decentralized reformulation we introduced the variable y, making the objective a *non*-strongly convex function of (x, y). To still obtain linear convergence we add the augmentation term $G(x,y) = \sum_i f_i(x_i) + \frac{r}{2}\|\mathbf{A}x + \gamma\mathbf{W}y - \mathbf{b}\|^2$. With appropriate coefficients, G is smooth and strongly convex enough.

Chebyshev's acceleration. Our constraint matrix $(\mathbf{A}\ \gamma\mathbf{W})$ consists of two matrices, multiplications by which correspond to different oracles. Therefore, we modify application of Chebyshev's acceleration from [1], by replacing $\mathbf{W}$ with $P_W(\mathbf{W})$ first and then applying Chebyshev's acceleration to matrix $(\mathbf{A}\ \gamma P_W(\mathbf{W}))$.

Results

Theorem (Algorithm)

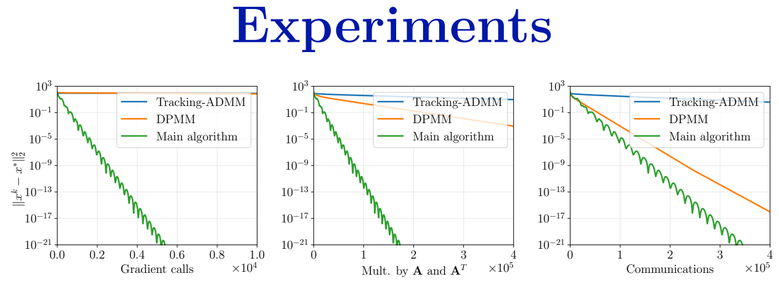
For every $\varepsilon > 0$, the proposed algorithm finds x^k for which $\|x^k - x^*\|^2 \leq \varepsilon$ using $O(\sqrt{\kappa_f}\log(1/\varepsilon))$ objective's gradient computations, $O(\sqrt{\kappa_f}\sqrt{\kappa_{\mathbf{A}}}\log(1/\varepsilon))$ multiplications by $\mathbf{A}$ and $\mathbf{A}^\top$, and $O(\sqrt{\kappa_f}\sqrt{\kappa_{\mathbf{A}}}\sqrt{\kappa_{\mathbf{W}}}\log(1/\varepsilon))$ communication rounds (multiplications by $\mathbf{W}$).

Theorem (Lower bound)

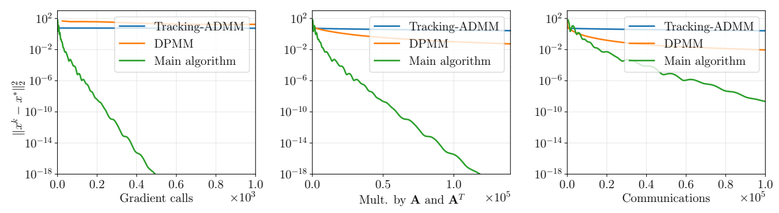
For any $L_f > \mu_f > 0$, $\kappa_{\mathbf{A}}, \kappa_{\mathbf{W}} > 0$ there exist L_f-smooth μ_f-strongly convex functions $\{f_i\}_{i=1}^n$, matrices $\mathbf{A}_i$ such that $\kappa_{\mathbf{A}} = L_{\mathbf{A}}/\mu_{\mathbf{A}}$, and a communication graph $\mathcal{G}$ with a corresponding gossip matrix $\mathbf{W}$ such that $\kappa_{\mathbf{W}} = \lambda_{\max}(\mathbf{W})/\lambda^+_{\min}(\mathbf{W})$, for which any first-order decentralized algorithm to reach accuracy ε requires at least $N_{\mathbf{A}} = \Omega\left(\sqrt{\kappa_f}\sqrt{\kappa_{\mathbf{A}}}\log\left(\frac{1}{\varepsilon}\right)\right)$ multiplications by $\mathbf{A}$ and $\mathbf{A}^\top$ and $N_{\mathbf{W}} = \Omega\left(\sqrt{\kappa_f}\sqrt{\kappa_{\mathbf{A}}}\sqrt{\kappa_{\mathbf{W}}}\log\left(\frac{1}{\varepsilon}\right)\right)$ communication rounds (multiplications by $\mathbf{W}$).

The corresponding lower bound on gradient computations is a classical result by Nesterov.

Experiments



Synthetic VFL, Erdős–Rényi graph, $n = 20$, $d_i = 3$, $m = 10$



LibSVM VFL, Erdős–Rényi graph, $n = 7$, $m = 100$

Summary

The simple augmentation trick and utilization of accelerated Forward-Backward algorithm [2] allowed to overpass the strong convexity issue and obtain an optimal first-order algorithm. Transition to the dual problem was not fruitful in this case.

The analysis is mostly linear algebra to derive spectral properties of block-matrices. All nasty inequalities stuff is hidden in the base algorithm's analysis.

References

[1] Salim et al., An optimal algorithm for strongly convex minimization under affine constraints

[2] Kovalev et al, Optimal and practical algorithms for smooth and strongly convex decentralized optimization