On the future of Department of MFoAI

Vladimir Temlyakov

Moscow; May 13, 2024

People

Name: Mathematical foundations of Artificial Intelligence.

- V.N. Temlyakov (Chair),
- P.A. Yaskov (Assistant Chair),
- E.V. Burnaev,
- A.I. Bufetov,
- A.V. Gasnikov,
- V.A. Kalyagin,
- B.S. Kashin,
- A.A. Naumov,
- I.V. Oseledets,
- A.A. Razborov,
- E.V. Schepin,
- E.E. Tyrtyshnikov,
- D.A. Yarotskii.

Seminar

Programming committee: A.I. Avetisyan, A.I. Aptekarev, B.S. Kashin, V.N. Temlyakov, E.E. Tyrtyshnikov.

Seminar

Programming committee: A.I. Avetisyan, A.I. Aptekarev, B.S. Kashin, V.N. Temlyakov, E.E. Tyrtyshnikov.
Organizing committee: V.N. Temlyakov, P.A. Yaskov.

Seminar

Programming committee: A.I. Avetisyan, A.I. Aptekarev, B.S. Kashin, V.N. Temlyakov, E.E. Tyrtyshnikov.
Organizing committee: V.N. Temlyakov, P.A. Yaskov.
Talks

- A.V. Gasnikov, Federated learning. Fundamental aspects. Open problems.
  14 February 2024.
- I.V. Oseledets, How mathematics can help in development and study of algorithms of the artificial intelligence.
  28 February 2024.
- D.A. Yarotskii, Kolmogorov's theorem and neural nets.
  13 March 2024.

Talks continue

- D.A. Yarotskii, Universal formulas: structure and learning by graduate descent.
  27 March 2024.
- E.V. Burnaev, Optimal transport, barycenters of distributions, and generative models.
  10 April 2024.
- V.N. Temlyakov, Is there an alternative to neural networks?
  24 April 2024.

AI semester at MIAN

We plan to organize a semester on Artificial Intelligence in the Fall 2024. It will include

- Conference: Approximation, Optimization, and Sparse Recovery, Sirius, Sochi, September 16–20, 2024.

AI semester at MIAN

We plan to organize a semester on Artificial Intelligence in the Fall 2024. It will include

- Conference: Approximation, Optimization, and Sparse Recovery, Sirius, Sochi, September 16–20, 2024.
- Conference: International conference on computational optimization, Innopolis, Kazan, October 10–12.

General remark

- The goal of our research is to tackle fundamental mathematical problems in the area of Artificial Intellengence.

General remark

- The goal of our research is to tackle fundamental mathematical problems in the area of Artificial Intellengence.
- One of the main problems here is to obtain results, which provide rigorous mathematical justification for a successful practical applications of certain algorithms and approximation methods.

General remark

- The goal of our research is to tackle fundamental mathematical problems in the area of Artificial Intellengence.
- One of the main problems here is to obtain results, which provide rigorous mathematical justification for a successful practical applications of certain algorithms and approximation methods.
- The following big areas of research are of special importance here.

General remark

- The goal of our research is to tackle fundamental mathematical problems in the area of Artificial Intellengence.
- One of the main problems here is to obtain results, which provide rigorous mathematical justification for a successful practical applications of certain algorithms and approximation methods.
- The following big areas of research are of special importance here.
  a) Machine Learning (Learning Theory, Deep Learning, Nonparametric Statistics).

General remark

- The goal of our research is to tackle fundamental mathematical problems in the area of Artificial Intellengence.
- One of the main problems here is to obtain results, which provide rigorous mathematical justification for a successful practical applications of certain algorithms and approximation methods.
- The following big areas of research are of special importance here.
  a) Machine Learning (Learning Theory, Deep Learning, Nonparametric Statistics).
  b) Optimization Theory.

General Remark continue

- Two different approaches to these problems:

General Remark continue

- Two different approaches to these problems:
- (A) The classical approach is based on approximation and optimization in finite dimensional spaces.

General Remark continue

- Two different approaches to these problems:
- (A) The classical approach is based on approximation and optimization in finite dimensional spaces.
  The corresponding dimensions might be huge. Therefore, an important problem here is to obtain results, which do not depend on the dimension of the problem or, at least, the dependence on dimension is weak. This motivates the other approach:

General Remark continue

- Two different approaches to these problems:
- (A) The classical approach is based on approximation and optimization in finite dimensional spaces.

  The corresponding dimensions might be huge. Therefore, an important problem here is to obtain results, which do not depend on the dimension of the problem or, at least, the dependence on dimension is weak. This motivates the other approach:
- (B) A contemporary approach is based on the infinite dimensional model and on nonlinear sparse approximation with respect to a suitable system (dictionary).

Related area of research. Big Data

- A critical challenge for big data management and analysis is an economical representation of data (data compression).

Related area of research. Big Data

- A critical challenge for big data management and analysis is an economical representation of data (data compression).
- In addition to a thorough study of mathematical foundations of the AI, we propose to conduct a theoretical study of fundamental techniques used in sparse representation of data. We plan to develop a theory of sparse representation that broadly applicable to Big Data problems.

Related area of research. Big Data

- A critical challenge for big data management and analysis is an economical representation of data (data compression).
- In addition to a thorough study of mathematical foundations of the AI, we propose to conduct a theoretical study of fundamental techniques used in sparse representation of data. We plan to develop a theory of sparse representation that broadly applicable to Big Data problems.
- The main technique to be used in achieving this goal is based on nonlinear sparse representations.

Related area of research. Big Data

- A critical challenge for big data management and analysis is an economical representation of data (data compression).
- In addition to a thorough study of mathematical foundations of the AI, we propose to conduct a theoretical study of fundamental techniques used in sparse representation of data. We plan to develop a theory of sparse representation that broadly applicable to Big Data problems.
- The main technique to be used in achieving this goal is based on nonlinear sparse representations.
- The contemporary challenge is unstructured data, which come from different sources. Therefore, the theory broadly applicable to Big Data problems must address the problem of sparse representation with respect to an arbitrary (structured and unstructured) dictionary.

Some specific directions

- Federated Learning.
  Federated learning (FL) is a secure distributed machine learning paradigm that addresses the issue of data silos in building a joint model. Its unique distributed training mode and the advantages of security aggregation mechanism are very suitable for various practical applications with strict privacy requirements.
  MIAN has a joint project with ISP (Ivannikov Institute for System Programming), which, in particular, addresses the study of FL.

Some specific directions

- Neural Networks.
  Mathematically, neural networks are manifolds which are obtained by a combination of operations of linear combination, thresholding (cutoffs), and superposition. Neural networks play a fundamental role in Deep Learning and AI. However, mathematical study of these manifolds is not sufficient. The key point here is to study properties of superpositions. The fundamental result in this direction is Kolmogorov's superposition theorem (KST). We plan to thoroughly study neural networks from the point of view of their representation and approximation power.

Neural networks 1

We fix a univariate function $\sigma(t)$, $t \in \mathbb{R}$. Usually, this function takes values in $(0,1)$ and increases. Then as a system (dictionary) we consider

$$\{g(\mathbf{x}) \,:\, g(\mathbf{x}) = \sigma(\omega \cdot \mathbf{x} + b), \quad \mathbf{x}, \omega \in \mathbb{R}^d, b \in \mathbb{R}, \|g\|_2 = 1\}.$$

Constructions based on this dictionary are called shallow neural networks or neural networks with one lair.

Neural networks 2

We build approximating manifolds inductively. Let $\mathbf{x} \in \mathbb{R}^d$. Fix a univariate function $h(t)$. A popular one is the *ReLU* function: $ReLU(t) = 0$ for $t < 0$ and $ReLu(t) = t$ for $t \geq 0$. ReLU = Rectified Linear Unit.

Take numbers $s, n \in \mathbb{N}$ and build s-term approximants of depth n (neural network with n lairs). In the capacity of parameters take n matrices: A_1 of size $s \times d$, $A_2, \dots, A_n$ of size $s \times s$ and vectors $\mathbf{b}^1, \dots, \mathbf{b}^n, \mathbf{c}$ from $\mathbb{R}^s$.

At the first step define $\mathbf{y}^1 \in \mathbb{R}^s$

$$\mathbf{y}^1 := h(A_1\mathbf{x} + \mathbf{b}^1) := (h((A_1\mathbf{x})_1 + \mathbf{b}^1_1), \dots, h((A_1\mathbf{x})_s + \mathbf{b}^1_s))^T.$$

Note that $\mathbf{y}^1$ is a function on $\mathbf{x}$.

Neural networks 3

At the kth step ($k = 2, \dots, n$) define

$$\mathbf{y}^k := h(A_k \mathbf{y}^{k-1} + \mathbf{b}^k)$$

$$:= (h((A_k \mathbf{y}^{k-1})_1 + \mathbf{b}_1^k), \dots, h((A_k \mathbf{y}^{k-1})_s + \mathbf{b}_s^k))^T.$$

Finally, after the nth step we define

$$g_n(\mathbf{x}) := \langle \mathbf{c}, \mathbf{y}^n \rangle = \sum_{j=1}^{s} c_j y_j^n.$$

Thus we build a manifold, which is described by the following parameters: n matrices: A_1 of size $s \times d$, $A_2, \dots, A_n$ of size $s \times s$ and vectors: $\mathbf{b}^1, \dots, \mathbf{b}^n, \mathbf{c}$ from $\mathbb{R}^s$.

Convex optimization. Setting

A typical problem of convex optimization is to find an approximate solution to the problem

$$\inf_{x\in D} E(x) \tag{1}$$

under assumption that E is a convex function. We consider a convex function E defined on a Banach space X. Let X be a Banach space with norm $\|\cdot\|$.

We say that a set of elements (functions) $\mathcal{D}$ from X is a dictionary if each $g\in\mathcal{D}$ has norm bounded by one ($\|g\|\le 1$) and the closure of $\operatorname{span}\mathcal{D}$ is X.

We denote the closure (in X) of the convex hull of $\mathcal{D}^\pm := \{\pm g, g\in\mathcal{D}\}$ by $A_1(\mathcal{D})$.

Typical constraints

Sparsity Constraints: The set $\Sigma_n(\mathcal{D})$ of functions

$$g = \sum_{g \in \Lambda} c_g g, \quad |\Lambda| = n,$$

is called the set of *sparse* functions of order n with respect to the dictionary $\mathcal{D}$. One common assumption is to minimize E on $D = \Sigma_n(\mathcal{D})$, i.e. to look for an n sparse minimizer of (1).

Typical constraints

Sparsity Constraints: The set $\Sigma_n(\mathcal{D})$ of functions

$$g = \sum_{g \in \Lambda} c_g g, \quad |\Lambda| = n,$$

is called the set of *sparse* functions of order n with respect to the dictionary $\mathcal{D}$. One common assumption is to minimize E on $D = \Sigma_n(\mathcal{D})$, i.e. to look for an n sparse minimizer of (1).

ℓ_1 constraints: Minimize E over $A_1(\mathcal{D})$. A slightly more general setting is to minimize E over one of the sets

$$\mathcal{L}_M := \{g \in X : \; g/M \in A_1(\mathcal{D})\}.$$

Example

There has been considerable interest in solving the convex unconstrained optimization problem

$$\min_x \frac{1}{2}\|y - \Phi x\|_2^2 + \lambda \|x\|_1 \tag{2}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^k$, Φ is an $k \times n$ matrix, λ is a nonnegative parameter, $\|v\|_2$ denotes the Euclidian norm of v, and $\|v\|_1$ is the ℓ_1 norm of v.

Example

There has been considerable interest in solving the convex unconstrained optimization problem

$$\min_x \frac{1}{2}\|y - \Phi x\|_2^2 + \lambda\|x\|_1 \tag{2}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^k$, Φ is an $k \times n$ matrix, λ is a nonnegative parameter, $\|v\|_2$ denotes the Euclidian norm of v, and $\|v\|_1$ is the ℓ_1 norm of v.

Problems of the form (2) have become familiar over the past three decades, particularly in statistical and signal processing contexts. Problem (2) is closely related to the following convex constrained optimization problem

$$\min_x \frac{1}{2}\|y - \Phi x\|_2^2 \quad \text{subject to} \quad \|x\|_1 \le A. \tag{3}$$

Optimization in $\mathbb{R}^n$

The problem (3) is a convex optimization problem of the energy function $E(x,\Phi) := \frac{1}{2}\|y - \Phi x\|_2^2$ on the octahedron $\{x : \|x\|_1 \le A\}$ in $\mathbb{R}^n$.

Optimization in $\mathbb{R}^n$

The problem (3) is a convex optimization problem of the energy function $E(x,\Phi) := \frac{1}{2}\|y - \Phi x\|_2^2$ on the octahedron $\{x : \|x\|_1 \le A\}$ in $\mathbb{R}^n$.

- The domain of optimization is simple and all dependence on the matrix Φ is in the energy function $E(x,\Phi)$, which makes the problem difficult.

Optimization in $\mathbb{R}^n$

The problem (3) is a convex optimization problem of the energy function $E(x,\Phi) := \frac{1}{2}\|y - \Phi x\|_2^2$ on the octahedron $\{x : \|x\|_1 \leq A\}$ in $\mathbb{R}^n$.

- The domain of optimization is simple and all dependence on the matrix Φ is in the energy function $E(x,\Phi)$, which makes the problem difficult.
- Also, the domain is in the high dimensional space $\mathbb{R}^n$.

Recast of the above example

In typical applications, for instance in compressed sensing, k is much smaller than n. We recast the above problem as an optimization problem with $E(z) := \frac{1}{2}\|y - z\|_2^2$ over the domain $A_1(\mathcal{D})$.

Recast of the above example

In typical applications, for instance in compressed sensing, k is much smaller than n. We recast the above problem as an optimization problem with $E(z) := \frac{1}{2}\|y - z\|_2^2$ over the domain $A_1(\mathcal{D})$.

- We optimize with respect to a dictionary $\mathcal{D} := \{\varphi_i\}_{i=1}^n$, which is associated with a $k \times n$ matrix $\Phi = [\varphi_1 \dots \varphi_n]$ with $\varphi_j \in \mathbb{R}^k$ being the column vectors of Φ.

Recast of the above example

In typical applications, for instance in compressed sensing, k is much smaller than n. We recast the above problem as an optimization problem with $E(z) := \frac{1}{2}\|y - z\|_2^2$ over the domain $A_1(\mathcal{D})$.

- We optimize with respect to a dictionary $\mathcal{D} := \{\varphi_i\}_{i=1}^n$, which is associated with a $k \times n$ matrix $\Phi = [\varphi_1 \dots \varphi_n]$ with $\varphi_j \in \mathbb{R}^k$ being the column vectors of Φ.
- In this formulation the energy function $E(z)$ is very simple and all dependence on Φ is in the form of the domain $A_1(\mathcal{D})$.

Recast of the above example

In typical applications, for instance in compressed sensing, k is much smaller than n. We recast the above problem as an optimization problem with $E(z) := \frac{1}{2}\|y - z\|_2^2$ over the domain $A_1(\mathcal{D})$.

- We optimize with respect to a dictionary $\mathcal{D} := \{\varphi_i\}_{i=1}^n$, which is associated with a $k \times n$ matrix $\Phi = [\varphi_1 \dots \varphi_n]$ with $\varphi_j \in \mathbb{R}^k$ being the column vectors of Φ.
- In this formulation the energy function $E(z)$ is very simple and all dependence on Φ is in the form of the domain $A_1(\mathcal{D})$.
- Other important feature of the new formulation is that optimization takes place in the $\mathbb{R}^k$ with relatively small k.

Modulus of smoothness

We assume that the set $D := \{x : E(x) \le E(0)\}$ is bounded. For a bounded set D define the modulus of smoothness of E on D as follows

$$\rho(E,u) := \frac{1}{2} \sup_{x\in D, \|y\|=1} |E(x+uy)+E(x-uy)-2E(x)|. \qquad (4)$$

Modulus of smoothness

We assume that the set $D := \{x : E(x) \leq E(0)\}$ is bounded. For a bounded set D define the modulus of smoothness of E on D as follows

$$\rho(E,u) := \frac{1}{2} \sup_{x\in D, \|y\|=1} |E(x+uy)+E(x-uy)-2E(x)|. \qquad (4)$$

A typical assumption in convex optimization is of the form $(\|y\| = 1)$

$$|E(x+uy)-E(x)-\langle E'(x),uy\rangle| \leq Cu^2$$

which corresponds to the case $\rho(E,u)$ of order u^2. We assume that E is Fréchet differentiable.

Greedy algorithms. The Frank-Wolfe-type algorithm

Let $t \in (0,1]$ be a given weakness parameter.
Weak Relaxed Greedy Algorithm (WRGA(co)). We define $G_0 := 0$.
Then, for each $m \geq 1$ we define:

1. $\varphi_m \in \mathcal{D}$ is any element satisfying

$$\langle -E'(G_{m-1}), \varphi_m - G_{m-1} \rangle \geq t \sup_{g \in \mathcal{D}} \langle -E'(G_{m-1}), g - G_{m-1} \rangle.$$

Greedy algorithms. The Frank-Wolfe-type algorithm

Let $t \in (0, 1]$ be a given weakness parameter.
Weak Relaxed Greedy Algorithm (WRGA(co)). We define $G_0 := 0$. Then, for each $m \geq 1$ we define:

1. $\varphi_m \in \mathcal{D}$ is any element satisfying
$$\langle -E'(G_{m-1}), \varphi_m - G_{m-1} \rangle \geq t \sup_{g \in \mathcal{D}} \langle -E'(G_{m-1}), g - G_{m-1} \rangle.$$
2. Find $0 \leq \lambda_m \leq 1$ such that
$$E((1 - \lambda_m)G_{m-1} + \lambda_m \varphi_m) = \inf_{0 \leq \lambda \leq 1} E((1 - \lambda)G_{m-1} + \lambda \varphi_m)$$
and define $$G_m := (1 - \lambda_m)G_{m-1} + \lambda_m \varphi_m.$$

Rate of approximation

Theorem (T., 2012)

Let E *be a uniformly smooth convex function with modulus of smoothness* $\rho(E,u) \le \gamma u^q$, $1 < q \le 2$. *Then, for a parameter* $t \in (0,1]$ *we have*

$$E(G_m) - \inf_{f \in A_1(\mathcal{D})} E(f) \le C(t,q,\gamma) m^{1-q}.$$

Rate of approximation

Theorem (T., 2012)

Let E be a uniformly smooth convex function with modulus of smoothness $\rho(E,u) \le \gamma u^q$, $1 < q \le 2$. Then, for a parameter $t \in (0,1]$ we have

$$E(G_m) - \inf_{f \in A_1(\mathcal{D})} E(f) \le C(t,q,\gamma) m^{1-q}.$$

In the case $q = 2$ it goes back to Frank and Wolfe, 1956 (special case) and to Tewari, Ravikumar, and Dhillon, 2012.

Chebyshev algorithm

Weak Chebyshev Greedy Algorithm (WCGA(co)). Let $t \in (0,1]$ be a weakness parameter. We define $G_0 := 0$. Then for each $m \geq 1$ we have the following inductive definition.

- (1) $\varphi_m \in \mathcal{D}$ is any element satisfying

$$|\langle -E'(G_{m-1}), \varphi_m\rangle| \geq t \sup_{g\in\mathcal{D}} |\langle -E'(G_{m-1}), g\rangle|.$$

Chebyshev algorithm

Weak Chebyshev Greedy Algorithm (WCGA(co)). Let $t \in (0,1]$ be a weakness parameter. We define $G_0 := 0$. Then for each $m \geq 1$ we have the following inductive definition.

- (1) $\varphi_m \in \mathcal{D}$ is any element satisfying

$$|\langle -E'(G_{m-1}), \varphi_m \rangle| \geq t \sup_{g \in \mathcal{D}} |\langle -E'(G_{m-1}), g \rangle|.$$

- (2) Define $\Phi_m := \text{span}\{\varphi_j\}_{j=1}^m$, and define G_m to be the point from Φ_m at which E attains the minimum:
  $E(G_m) = \inf_{x \in \Phi_m} E(x)$.

Chebyshev algorithm with function evaluations

E-Greedy Chebyshev Algorithm (EGCA(co)). We define $G_0 := 0$. Then for each $m \geq 1$ we have the following inductive definition.

- (1) $\varphi_m \in \mathcal{D}$ is any element satisfying (assume existence)

$$\inf_c E(G_{m-1} + c\varphi_m) = \inf_{c, g \in \mathcal{D}} E(G_{m-1} + cg).$$

Chebyshev algorithm with function evaluations

E-Greedy Chebyshev Algorithm (EGCA(co)). We define $G_0 := 0$. Then for each $m \geq 1$ we have the following inductive definition.

- (1) $\varphi_m \in \mathcal{D}$ is any element satisfying (assume existence)

$$\inf_c E(G_{m-1} + c\varphi_m) = \inf_{c, g \in \mathcal{D}} E(G_{m-1} + cg).$$

- (2) Define $\Phi_m := \operatorname{span}\{\varphi_j\}_{j=1}^m$, and define G_m to be the point from Φ_m at which E attains the minimum: $E(G_m) = \inf_{x \in \Phi_m} E(x)$.

WGAFR(co)

Weak Greedy Algorithm with Free Relaxation (WGAFR(co)). Let $t \in (0, 1]$, be a weakness parameter. We define $G_0 := 0$. Then for each $m \geq 1$ we have:

1. $\varphi_m \in \mathcal{D}$ is any element satisfying

$$|\langle -E'(G_{m-1}), \varphi_m \rangle| \geq t \sup_{g \in \mathcal{D}} |\langle -E'(G_{m-1}), g \rangle|.$$

WGAFR(co)

Weak Greedy Algorithm with Free Relaxation (WGAFR(co)). Let $t \in (0,1]$, be a weakness parameter. We define $G_0 := 0$. Then for each $m \geq 1$ we have:

1. $\varphi_m \in \mathcal{D}$ is any element satisfying

$$|\langle -E'(G_{m-1}), \varphi_m \rangle| \geq t \sup_{g \in \mathcal{D}} |\langle -E'(G_{m-1}), g \rangle|.$$

2. Find w_m and λ_m such that

$$E((1-w_m)G_{m-1} + \lambda_m \varphi_m) = \inf_{\lambda, w} E((1-w)G_{m-1} + \lambda \varphi_m)$$

and define $\quad G_m := (1-w_m)G_{m-1} + \lambda_m \varphi_m.$

Rate of convergence

Theorem (T, 2012)

Let E be a uniformly smooth convex function with modulus of smoothness $\rho(E,u) \leq \gamma u^q$, $1 < q \leq 2$. Take a number $\varepsilon \geq 0$ and assume that there is an element f^ε from D such that

$$E(f^\varepsilon) \leq \inf_{x\in D} E(x) + \varepsilon, \quad f^\varepsilon/B \in A_1(\mathcal{D}),$$

with some number $B = C(E,\varepsilon,\mathcal{D}) \geq 1$. Then we have for the WCGA(co), ECGA(co), and WGAFR(co)

$$E(G_m) - \inf_{x\in D} E(x) \leq \max\left(2\varepsilon, C_1(t,E,q,\gamma)B^q m^{1-q}\right).$$

Extra conditions on E

E1. Smoothness. We assume that E is a convex function with $\rho(E,u) \leq \gamma u^2$.

Extra conditions on E

E1. Smoothness. We assume that E is a convex function with $\rho(E,u) \le \gamma u^2$.

E2. Restricted strong convexity. We assume that for any S-sparse element f we have

$$E(f) - E(f_0) \ge \beta \|f - f_0\|^2. \tag{5}$$

Gradient type algorithms

The most difficult part of an algorithm is to find an element $\varphi_m \in \mathcal{D}$ to be used in approximation process. We consider greedy methods for finding $\varphi_m \in \mathcal{D}$. We have two types of greedy steps to find $\varphi_m \in \mathcal{D}$.

I. Gradient greedy step. At this step we look for an element $\varphi_m \in \mathcal{D}$ such that

$$\langle -E'(G_{m-1}), \varphi_m \rangle \geq t_m \sup_{g \in \mathcal{D}} \langle -E'(G_{m-1}), g \rangle.$$

Algorithms that use the first derivative of the objective function E are called first order optimization algorithms.

Zero order algorithms

II. E-greedy step. At this step we look for an element $\varphi_m \in \mathcal{D}$ which satisfies (we assume existence):

$$\inf_{c\in\mathbb{R}} E(G_{m-1}+c\varphi_m) = \inf_{g\in\mathcal{D},c\in\mathbb{R}} E(G_{m-1}+cg).$$

Algorithms that only use the values of the objective function E are called zero order optimization algorithms.

Approximation step

After we found $\varphi_m \in \mathcal{D}$ we can proceed in different ways. We now list some typical steps that are motivated by the corresponding steps in greedy approximation theory. These steps or their variants are used in optimization algorithms like gradient method, reduced gradient method, conjugate gradients, gradient pursuits.

(A) Best step in the direction $\varphi_m \in \mathcal{D}$. We choose c_m such that

$$E(G_{m-1} + c_m\varphi_m) = \inf_{c\in\mathbb{R}} E(G_{m-1} + c\varphi_m)$$

and define

$$G_m := G_{m-1} + c_m\varphi_m.$$

Other approximation steps

(B) Shortened best step in the direction $\varphi_m \in \mathcal{D}$. We choose c_m as in (A) and for a given parameter $b > 0$ define

$$G_m^b := G_{m-1}^b + bc_m\varphi_m.$$

Usually, $b \in (0,1)$. This is why we call it shortened.

(C) Chebyshev-type (fully corrective) methods. We choose $G_m \in \operatorname{span}(\varphi_1, \dots, \varphi_m)$ which satisfies

$$E(G_m) = \inf_{c_j, j=1,\dots,m} E(c_1\varphi_1 + \dots + c_m\varphi_m).$$

(D) Fixed relaxation. For a given sequence $\{r_k\}_{k=1}^{\infty}$ of relaxation parameters $r_k \in [0,1)$ we choose $G_m := (1-r_m)G_{m-1} + c_m\varphi_m$ with c_m from

$$E((1-r_m)G_{m-1} + c_m\varphi_m) = \inf_{c\in\mathbb{R}} E((1-r_m)G_{m-1} + c\varphi_m).$$

More approximation steps

(F) Free relaxation. We choose $G_m \in \operatorname{span}(G_{m-1}, \varphi_m)$ which satisfies

$$E(G_m) = \inf_{c_1, c_2} E(c_1 G_{m-1} + c_2 \varphi_m).$$

(G) Prescribed coefficients. For a given sequence $\{c_k\}_{k=1}^{\infty}$ of positive coefficients in the case of greedy step **I** we define

$$G_m := G_{m-1} + c_m \varphi_m. \tag{6}$$

In the case of greedy step **II** we define G_m by formula (6) with the greedy step **II** modified as follows: $\varphi_m \in \mathcal{D}$ is an element satisfying

$$E(G_{m-1} + c_m \varphi_m) = \inf_{g \in \mathcal{D}} E(G_{m-1} + c_m g).$$

The End

Thank you!