Randomized Algorithm of Multi-dimensional Optimization and its Implementations on Quantum Computers

ITMO University,
St. Petersburg, Russia

WS Quantum Informatics and Applications in Economic and Finance
06 November 2014

Outline

1. Introduction

2. Mean-risk multivariable optimization
   - Randomized stochastic approximation (SPSA)
   - Quantum computing
   - Estimation under arbitrary noise
   - Linear regression, filtering, and machine learning

3. Applications

4. Conclusion

Control and Data Processing “on the Fly”

Usually, to “extract” information from the data in the real time environment we have:

- substantial restrictions of resources;
- an insufficient number of data with the necessary diversity.

Mean-Risk Multivariable Optimization

Let us assume that we can chose
points of measurements $x_1, x_2, \ldots \in \mathbb{R}^d$
and on each iteration we can measure:

$$y_t = f(x_t, w_t) + v_t, \tag{1}$$

where $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$, w_t is a random vector which is defined on the basic probability space $\{\Omega, \mathscr{F}, P\}$. It represents non-controlled random uncertainty, and v_n is an external arbitrary observation noise.
Consider the minimization problem:

$$F(x) = \int f(x, w) \mathrm{P}(dw) \to \min_x . \tag{2}$$

Finite Difference Approach

The number of observations per one iteration is $N = 2d$
$\widehat{\theta}_0 \in \mathbb{R}^d$

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} - \frac{\alpha_n}{2\beta_n}(Y_n^+ - Y_n^-),$$

$$x_n^{(i,\pm)} = \widehat{\theta}_{n-1} \pm \beta_n e_i$$

$$Y_n^\pm = \begin{pmatrix} f(x_n^{(1,\pm)}, w_n^{(1,\pm)}) + v_n^{(1,\pm)} \\ f(x_n^{(2,\pm)}, w_n^{(2,\pm)}) + v_n^{(2,\pm)} \\ \vdots \\ f(x_n^{(d,\pm)}, w_n^{(d,\pm)}) + v_n^{(d,\pm)} \end{pmatrix}$$

Randomized Stochastic Approximation

We reduce the number of observations up to 1 or 2 instead of $2d$ (!)
One measurement form

$$x_n = \widehat{\theta}_{n-1} + \beta_n \Delta_n, \ \ \Delta_n = \begin{pmatrix} \pm 1 \\ \pm 1 \\ \vdots \\ \pm 1 \end{pmatrix}$$

$$y_n = f(x_n, w_n) + v_n$$

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n} \mathscr{K}_n(\Delta_n) y_n$$

Two measurements form

$$x_n^{\pm} = \widehat{\theta}_{n-1} \pm \beta_n^{\pm} \Delta_n$$

$$y_n^{\pm} = f(x_n^{\pm}, w_n^{\pm}) + v_n^{\pm}$$

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n^+ + \beta_n^-} \mathscr{K}_n(\Delta_n)(y_n^+ - y_n^-)$$

$$x_n^+ = \widehat{\theta}_{n-1} + \beta_n \Delta_n, \ x_n^- = \widehat{\theta}_{n-1}$$

$$y_n^{\pm} = f(x_n^{\pm}, w_n^{\pm}) + v_n^{\pm}$$

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n} \mathscr{K}_n(\Delta_n)(y_n^+ - y_n^-)$$

Tracking

Granichin et. al. (2009, 2014)

$$\{f_\xi(\mathbf{x},\mathbf{w})\}_{\xi\in\Xi}$$

$$y_t = f_{\xi_t}(\mathbf{u}_t,\mathbf{w}_t) + v_t \tag{3}$$

$$F_t(\mathbf{x}) = \int f_{\xi_t}(\mathbf{x},\mathbf{w}_t)\mathrm{P}(d\mathbf{w}_t) \to \min_{\mathbf{x}} \tag{4}$$

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \zeta,\ \mathbf{x}_n \in \mathbb{R}^d$$

Advantages of Algorithms

Granichin (1989, 1992), Polyak and Tsybakov (1990), Spall (1992, 1997), Chen, Duncan and Pasik-Duncan (1999)

- Asymptotic-optimal rate of convergence
- Min number of observations per iteration
- Allows tracking
- Consistency under almost arbitrary external noise
- Easy to implement on quantum computer

Quantum Computing

The representation of SPSA algorithm is associated with something well known to those familiar with the fundamentals of quantum computing. Virtually all known effective quantum algorithms implement a similar scheme:

- preparing input “superposition”,
- processing,
- measuring a result.

Quantum Computing

Let us assume that classical data, a row i of length l, $l \leq r$, is fed as input into a quantum computer. In quantum computation r q-bits initialize in state $|i00\dots0\rangle$. An executable circuit is constructed from a finite number of quantum circuits acting on these q-bits. At the end of computation, the quantum computer passes into some state that is a unit vector in space $\mathbb{C}^{2^r}$. This state can be represented as

$$\mathbf{x} = \sum_e x_e |e\rangle,$$

where summation is taken over all binary rows of length r, $x_e \in \mathbb{C}$, $\sum_e |x_e|^2 = 1$ (x_e are called probabilistic amplitudes and $\mathbf{x}$ is called a superposition of basis vectors $|e\rangle$).

Quantum Computing

The Heisenberg uncertainty principle asserts that the state of a quantum system cannot be predicted exactly. The observation concept is defined as an operator in Hilbertian state space **U** which is equivalent to the scalar product with some given vector **u**:

$$\mathbf{U}|\mathbf{x}\rangle = \langle \mathbf{u}, \mathbf{x}\rangle.$$

The projection of each q-bit on the basis $\{|0\rangle, |1\rangle\}$ is usually used in measurement. The result of this measurement is the computation result.

Deutsch's problem

Let function $f : \{0,1\} \longrightarrow \{0,1\}$ be defined as a black box and a process of its computation continues for 24 hours.

The following question *must be answered*: Is function $f(u)$ constant or balanced?

In the classical case, obviously, not less than 48 hours should be spent to answer the question.

Let the given quantum black box compute $f(u)$. More precisely, let us define two q-bit unitary transformations:

$$\mathbf{U}_f : |u\rangle|z\rangle \longrightarrow |u\rangle|z \oplus f(u)\rangle,$$

which flip the second q-bit if the value of f from the first q-bit is 1. We have to determine whether or not $f(0) = f(1)$. If we are limited to classical inputs $|0\rangle$ and $|1\rangle$, we must call the box twice ($u = 0$ and $u = 1$) in order to get the answer. However, if we are allowed to introduce a coherent superposition of these classical states, calling the box one time is sufficient to answer the question!

Hadamard Transformation

In quantum computing the Hadamard transformation plays a special role. It is defined by formula

$$\mathbf{H} : |u\rangle \longrightarrow \frac{1}{\sqrt{2}} \sum_{z} (-1)^{uz} |uz\rangle, \tag{5}$$

or $\mathbf{H} : \begin{pmatrix} |0\rangle \\ |1\rangle \end{pmatrix} \longrightarrow \begin{pmatrix} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{pmatrix}$, that is $\mathbf{H} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$.

The Hadamard transform is used to prepare the superposition of the input data. In a certain sense, this transform can be interpreted as randomized inputs.

Deutsch's Problem Solving

If the input of the circuit is a couple of quantum bits $|0\rangle|1\rangle$ then we obtain successively

$$|0\rangle|1\rangle \longrightarrow \frac{1}{2}(|0\rangle+|1\rangle)(|0\rangle-|1\rangle)\frac{1}{2}[(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle](|0\rangle-|1\rangle) \longrightarrow$$

$$\longrightarrow \frac{1}{2}\left[\left((-1)^{f(0)}+(-1)^{f(1)}\right)|0\rangle+\left((-1)^{f(0)}-(-1)^{f(1)}\right)|1\rangle\right]\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle).$$

Hence, when we measure the first q-bit we obtain the result $|0\rangle$ with probability one if $f(0)=f(1)$ (that is, f is a constant function) and the result $|1\rangle$ with probability one if $f(0)\neq f(1)$ (balanced function).

Deutsch's Problem Solving

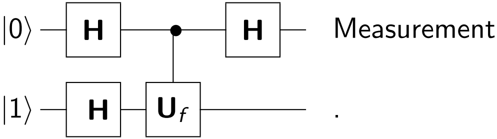
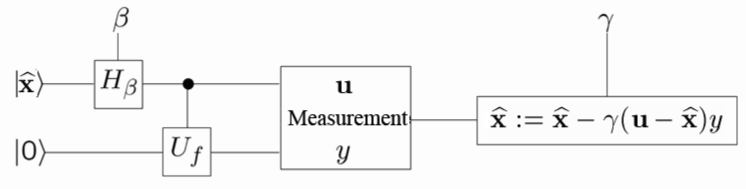


Figure: Quantum algorithm.

SPSA Algorithm Implementation and Quantum Computing

$$\mathbf{u} = \frac{1}{2^{\frac{d}{2}}} \sum_{\mathbf{\Delta}_i \in \{-1,+1\}^d} |\widehat{\mathbf{x}} + \beta \mathbf{\Delta}_i\rangle = \mathbf{H}_\beta |\widehat{\mathbf{x}}\rangle$$

$$\mathbf{U}_f |\mathbf{u}\rangle |0\rangle = \frac{1}{2^d} \sum_{\mathbf{\Delta}_i \in \{-1,+1\}^d} |\widehat{\mathbf{x}} + \beta \mathbf{\Delta}_i\rangle |f(\widehat{\mathbf{x}} + \beta \mathbf{\Delta}_i)\rangle$$



β
γ
$|\widehat{\mathbf{x}}\rangle$
H_β
$|0\rangle$
U_f
$\mathbf{u}$
Measurement
y
$\widehat{\mathbf{x}} := \widehat{\mathbf{x}} - \gamma(\mathbf{u} - \widehat{\mathbf{x}})y$

Figure: The quantum circuit for “on the fly” computing of the gradient.

The Algorithm's Ground

$$E\{\widehat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n}\Delta_n y_n | \mathscr{F}_{n-1}\} =$$

$$= \widehat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n}(E\{\Delta_n f(x_n)|\mathscr{F}_{n-1}\} + E\{\Delta_n\}E\{v_n|\mathscr{F}_{n-1}\}) =$$

$$= \widehat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n}(E\{\Delta_n f(\widehat{\theta}_{n-1} + \beta_n\Delta_n)|\mathscr{F}_{n-1}\} \approx$$

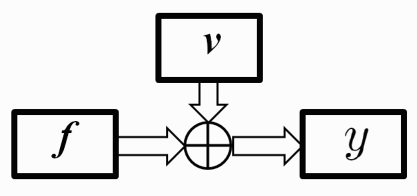
$$\approx \widehat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n}(E\{\Delta_n f(\widehat{\theta}_{n-1}) + \frac{\beta_n\Delta_n\Delta_n\nabla f(\widehat{\theta}_{n-1})}{2}|\mathscr{F}_{n-1}\} =$$

$$= \widehat{\theta}_{n-1} - \frac{\alpha_n}{2}\nabla f(\widehat{\theta}_{n-1})$$

Arbitrary External Noise

If some signal f goes into the recorder with a noise v then the “instantaneous” observation y_t can be written as

$$y_t = \mathrm{A} f_t + v_t \tag{6}$$



- $v = 0$.
- $v \approx 0$.
- $v_t \to 0$ as $t \to \infty$.
- v_t, $t = 1, \ldots, T$, is i.i.d. with $\sigma_v < \infty$
- Arbitrary external noise

Data Averaging

If our registering apparatus averages signals f_t coming at $t = 1, \dots, T$ then at the output we receive

$$y = \frac{1}{N} \sum_{t=1}^{T} f_t + \frac{1}{N} \sum_{t=1}^{T} v_t \qquad (7)$$

If v_t, $t = 1, \dots, T$, is i.i.d. with mean value M_v and variance $\sigma_v < \infty$ then

$$\mathrm{Prob} \left\{ |\frac{1}{N} \sum_{t=1}^{T} v_t - M_v| > \varepsilon \right\} \to 0 \;\; as \;\; t \to \infty.$$

Hence, we can use estimates

$$\hat{f} = \frac{1}{N} \sum_{t=1}^{T} y_t - M_v.$$

Estimation under Arbitrary Noise

Is it possible to get smart estimates?

Modernize the problem by including into the observations model the controllable input u. Following the paradigm inseparability of an information and control, we assume that the measured signal f at time t is directly determined by the current input u_t and some unknown parameter $\theta_\star$ (an unknown coefficient of gain/attenuation inputs).

$$f_t = u_t \theta_\star. \tag{8}$$

The problem is *to find or estimate* the unknown parameter $\theta_\star \in \mathbb{R}$ by the sequence of inputs and outputs $\{u_t, y_t\}$ without any restrictions for the sequence $\{v_t\}$ of external noises.

Problem Description

The model of observations (6) can be rewritten as:

$$y_t = u_t \theta_\star + v_t. \tag{9}$$

And we can

- chose the inputs (controls) u_t, $t = 1, 2, ..., T$,
- measure the outputs y_t (see Fig. 3).

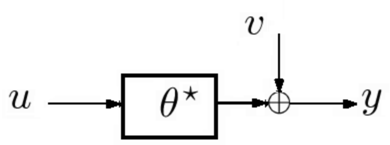
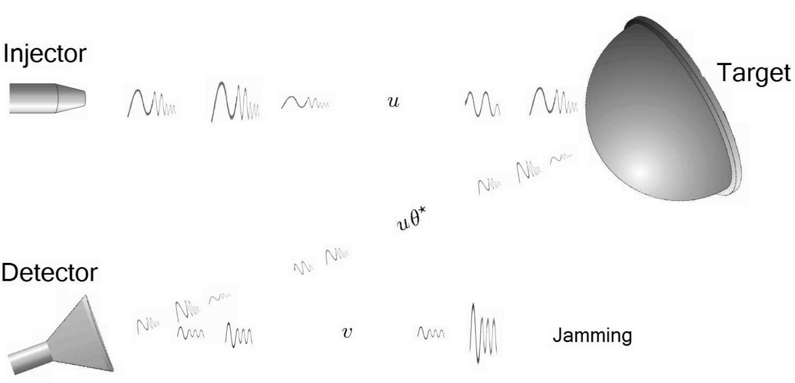


Figure: The model of observations

If we use $u_t \equiv 1$, we obtain the traditional problem of estimating of unknown parameter $\theta_\star$ observed with noise

The Source, Target and Detector of the Reflected Signal



An Algorithm of Estimation of $\theta_\star$

1. Control u_t selection and feeding it to the system input.
2. Receive the response from the system y_t.
3. Estimate the parameter $\theta_\star$ based on the data obtained u_t, y_t (for example, calculation of an estimate $\widehat{\theta}_t$ or set $\widehat{\Theta}_t$ containing $\theta_\star$).
4. Repeat steps 1–3.

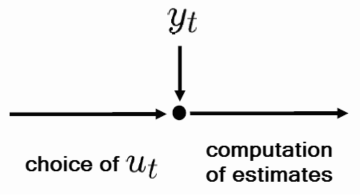


Figure: A model of an estimation algorithm.

Deterministic Algorithm

Definition

An algorithm is called *a deterministic algorithm* if each of its steps defined by the user is given by deterministic rules using the results of the previous steps, and obtained new data (output) is returned for using in subsequent steps of the algorithm.

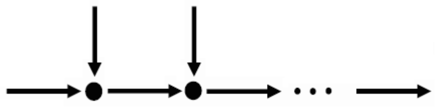


Figure: A model of a deterministic algorithm.

Deterministic Approaches are Often Fail !

- In theory and practice many difficulties arise when we try to make analytical investigation of “complex” systems.
- In many practical applications traditionally efficient deterministic methods fail to yield a result when the system is complex.
- In particular, this leads to the notion of *NP*-hard problems.

There are no Deterministic Algorithm Under Arbitrary External Noise!

Let be $\theta_\star = 3$

$$\widehat{\theta}_t = \frac{1}{t}\sum_{i=1}^{t} y_i$$

Table:

t	1	2	3	4	5	6	7
u_t	1	1	1	1	1	1	1
$v_t = rand() - 0.5$							
y_t	2.9	2.8	3.2	3.3	2.6	3.4	2.7
$\widehat{\theta}_t$	2.9	2.85	2.97	3.05	2.96	3.03	2.99
$v_t = rand() - 0.5 + m,\ m = 1$							
y_t	3.9	3.8	4.2	4.3	3.6	3.9	4.2
$\widehat{\theta}_t$	3.9	3.85	3.97	4.05	3.96	4.03	3.99

Randomized Algorithms

Randomization is a powerful tool for solving a number of problems deemed unsolvable with deterministic methods.

Definition

An algorithm is called a *randomized algorithm* when the execution of one or more steps, which are defined by the user, is based on a random rule (that is, among many deterministic rules one is chosen randomly according to a probability P).

"Enriched" Observations

Consider the following rule of a random input selection for the first step

$$u_t = \begin{cases} +1, & \text{with probability } \frac{1}{2}, \\ -1, & \text{with probability } \frac{1}{2}. \end{cases} \tag{10}$$

At the second step from the known values (u_t, y_t) we form value

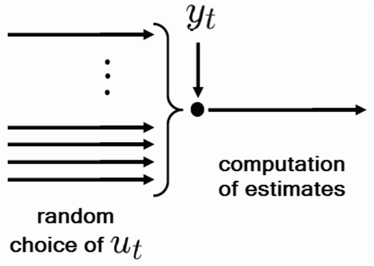
$$\tilde{y}_t = u_t \cdot y_t.$$

For the "new" sequence of observations we have a similar to (9) model

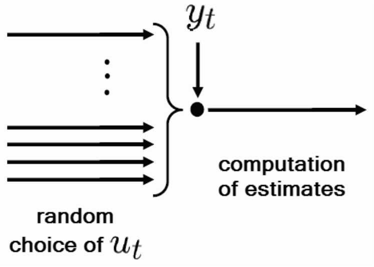
$$\tilde{y}_t = \tilde{u}_t \cdot \theta_\star + \tilde{v}_t,$$

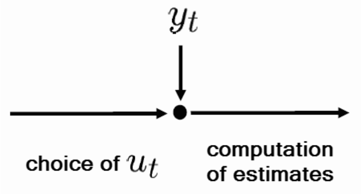
where $\tilde{u}_t = u_t^2$ and $\tilde{v}_t = u_t \cdot v_t$.

Diagram of Randomized Algorithm



Two Kinds of Algorithms





Results of Simulation

$$\theta_\star = 3, \qquad \widehat{\theta}_t = \frac{1}{t}\sum_{i=1}^{t} \tilde{y}_i = \frac{1}{t}\sum_{i=1}^{t} u_i y_i$$

Table:

<table>
<tr><th>t</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td>u_t</td><td>-1</td><td>1</td><td>-1</td><td>1</td><td>1</td><td>1</td><td>-1</td></tr>
<tr><td colspan="8">$v_t = rand() - 0.5 + m, \ m = 1$</td></tr>
<tr><td>y_t</td><td>-2.1</td><td>3.8</td><td>-1.8</td><td>4.3</td><td>3.6</td><td>4.4</td><td>-2.3</td></tr>
<tr><td>$\tilde{u}_t$</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>$\tilde{y}_t$</td><td>2.1</td><td>3.8</td><td>1.8</td><td>4.3</td><td>3.6</td><td>4.4</td><td>2.3</td></tr>
<tr><td>$\widehat{\theta}_t$</td><td>2.1</td><td>2.95</td><td>2.57</td><td>3.00</td><td>3.12</td><td>3.33</td><td>3.19</td></tr>
</table>

$$\forall t,\ \forall \varepsilon > 0 \ \ Prob\{|\widehat{\theta}_t - \theta_\star| \geq \varepsilon\} \leq \frac{1}{t}\frac{\mathrm{E}\{v_t^2\}}{\varepsilon^2} + o\left(\frac{1}{t}\right).$$

[Granichin, TAC, 2004]

Non-Asymptotic Result

For the finite number of observations ($N = 7$) a new rigorous mathematical result of a guaranteed set of possible values of the unknown parameter $\theta_\star$ can be obtained for arbitrary external noise v_t following by the method described by M. Campi [EJC, 2010]:

1. Let be $M = 8$ and select randomly seven ($= M - 1$) different groups of four indexes $T_1, \ldots, T_7$.
2. Compute the partial sums $\bar{s}_i = \frac{1}{4} \sum_{j \in T_i} \bar{y}_j$, $i = 1, \ldots, 7$.
3. Build the confidence interval

$$\widehat{\Theta} = [\min_{i \in 1:7} \bar{s}_i ; \max_{i \in 1:7} \bar{s}_i],$$

which contains $\theta_\star$ with the probability $p = 75\%$ ($= 1 - 2 \cdot 1/M$).

Confidence Interval

For the previous data $\{(u_t, y_t)\}$ we obtain by the described method:

Table:

i	T_i	$\bar{s}_i$
1	$\{2, 3, 4, 5\}$	3.375
2	$\{1, 3, 4, 6\}$	3.15
3	$\{2, 3, 5, 6\}$	3.4
4	$\{1, 2, 6, 7\}$	3.15
5	$\{1, 4, 5, 7\}$	3.075
6	$\{2, 3, 5, 7\}$	2.875
7	$\{1, 4, 6, 7\}$	3.275

Hence,

- unknown parameter $\theta_\star$ belongs to interval $\widehat{\Theta} = [2.875; 3.4]$ with probability $p = 75\%$.

The randomization in the process of the input data selection can get quite reasonable results.

Randomized and Bayesian Approaches

An alternative probabilistic approach is a Bayesian estimation when noise's v_t probability is attributed a priori to nature Q.

However, Bayesian and randomized approaches are quite different from the practical point of view.

In Bayesian approach probability Q describes a probability of a value of v_t in a comparison with other, i.e. the choice of Q is a part of the problem model.

In contrast, the probability P in a randomized approach is selected artificially. P exists only in our algorithm, and therefore there is no a traditional problem of "a bad model" as can happen with Q in a Bayesian approach.

Randomization for Reducing a Computational Complexity

The general idea is:
Suppose that a deterministic algorithm requires a huge amount of computing resources to process all available information.
Then we can intentionally give up part of the information and proceed to the solution of the simplified problem with partial information.
In this case, however, a deterministic solvability may be impossible to achieve, but as above we can consider a randomized approach to defining solutions with a high probability of success. The end result is a compromise between the full guarantee of success and computational feasibility (the opportunity to get a real answer for a limited time).

Other results

- Linear regression
- Filtering
- Machine learning

Applications

- Photoemission Experiment
- UAV Soaring
- Adaptive Optimization of a Server
- Load Balancing

Randomization in Photoemission Experiment

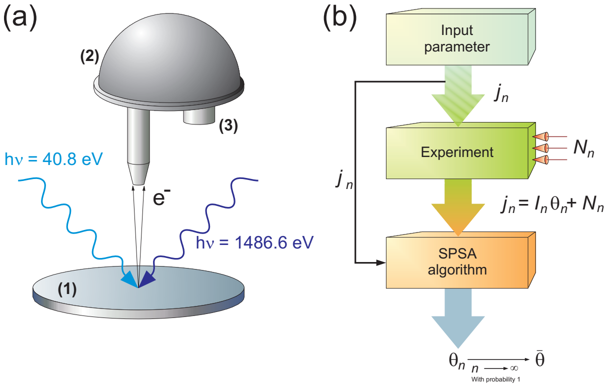


Figure: (a) Scheme of the PE experiment, where sample (1) is illuminated by two light sources, He IIα and Al Kα, electrons are analyzed by photoelectron spectrometer (2) and detector (3) registers the signal j_n. (b) Layout of the present study using the rand. algorithm for eliminating noises of unknown nature.

Results

Granichin O., Molodtsov S. *et al.* Review of Scientific Instruments. 79, 036103. 2008.

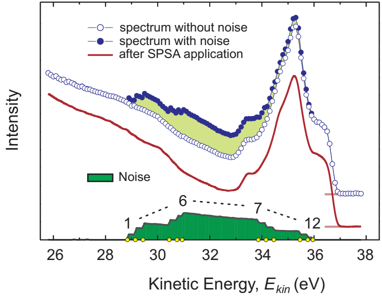


Figure: Experimental PE spectra of the valence band of W(110) obtained with He IIα radiation without (open circles) and with (filled circles) systematic noise. Spectrum obtained after application of the rand. algorithm to a series of 50 experimental single-scan spectra is shown by thin line. The shaded area in the bottom is systematic noise measured separately.

UAV Soaring

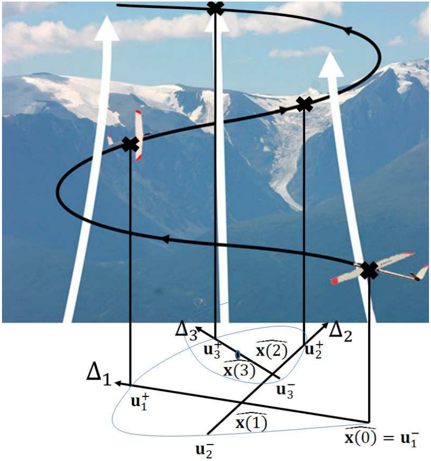
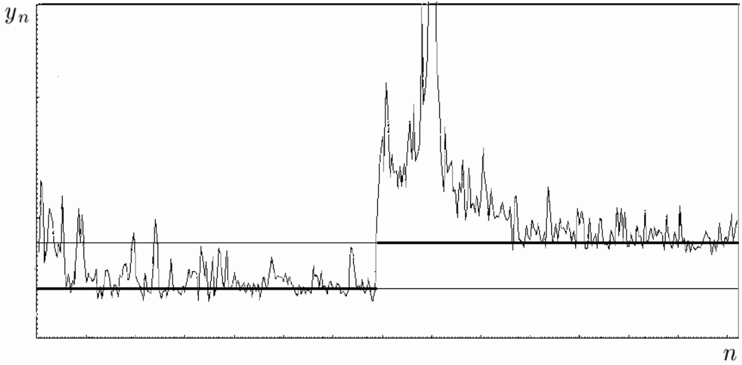

Δ_3
$\mathbf{u}_3^+$
$\widehat{\mathbf{x}(2)}$
$\mathbf{u}_2^+$
Δ_2
$\widehat{\mathbf{x}(3)}$
Δ_1
$\mathbf{u}_3^-$
$\mathbf{u}_1^+$
$\widehat{\mathbf{x}(1)}$
$\widehat{\mathbf{x}(0)} = \mathbf{u}_1^-$
$\mathbf{u}_2^-$

Figure: The sequence of estimates and waypoints.

Adaptive Optimization of a Server

$$F(x) = q(x) + L(x) \equiv q(x) + \lim_{T} \frac{1}{T} \sum_{t=1}^{T} Ey_t(x) \to \min_{x},$$

$$y_n = \frac{t_n - t_{n-1}}{x} d_{load} + \frac{1}{N} \sum_{t_i^{out} \in [t_{n-1}, t_n]} (t_i^{out} - t_i^{in} - d_i). \qquad (11)$$



Load Balancing

$$\sum_{j=1}^{m} u_j = z_n,$$

$$T(\mathbf{u}) = ||\mathbf{t}(\mathbf{u})||_\infty = \max_{j\in 1..m} t_j(u_j) \to \min_{\mathbf{u}}. \tag{12}$$

$$x_1 u_1 = x_2 u_2 = \cdots = x_m u_m.$$

$$F(\widehat{\mathbf{x}}_n) = \sum_{j,k=1}^{m} (\bar{t}_n^j - \bar{t}_n^k)^2 \ \to \ \min_{\widehat{\mathbf{x}}_n}, \tag{13}$$

$$\widehat{\mathbf{x}}(n) = \widehat{\mathbf{x}}(n-1) - \frac{\alpha}{\beta}\mathbf{\Delta}_n \sum_{j,k=1}^{m} (\Delta_n^j - \Delta_n^k)(\bar{t}_{2n}^j + \bar{t}_{2n-1}^j - \bar{t}_{2n}^k - \bar{t}_{2n-1}^k).$$

Randomization ...

1928-30 ...

- von Neumann (minimax theorem), Fisher (remove bias)

1950 ... 1975

- Metropolis, Ulam (method Monte-Carlo)
- Rastrigin, Kirkpatrick, Holland (random search, simulation annealing, genetic algorithm)

1980 ... 1999

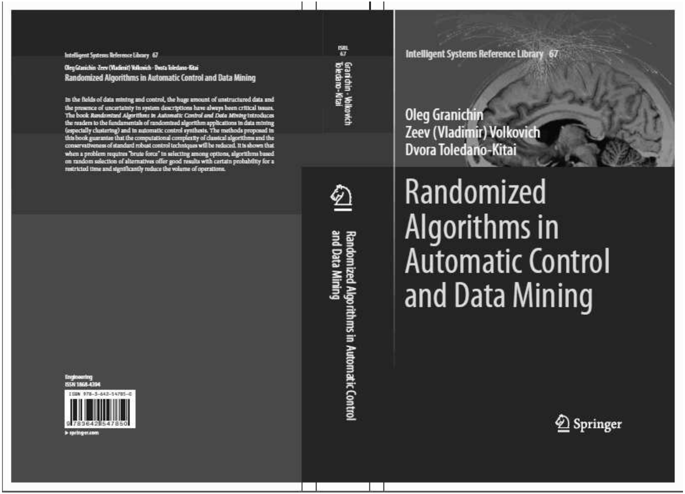
- Granichin, Fomin, Chen, Guo (randomized control strategies)
- Polyak, Thzubakov, Luing, Guffi, Spall (fast algorithms)
- Granichin (arbitrary noise)
- Vadiyasagar (randomized learning theory)

2000 ...

- Tempo, Campi, Calafiore, Dabbene, Polyak, Sherbakov *etc.* (probabilistic methods in a control syntheses, scenario approach)
- Candes, Donoho, Romberg, Tao (compressive sensing)

Best Features

- Significantly decreasing the number of operations
- Annihilating the systematic errors (the bias effect or an arbitrary noise)
- Accuracy usually not depend on the dimension of data



Intelligent Systems Reference Library 67

Oleg Granichin · Zeev (Vladimir) Volkovich · Dvora Toledano-Kitai

Randomized Algorithms in Automatic Control and Data Mining

In the fields of data mining and control, the huge amount of unstructured data and the presence of uncertainty in system descriptions have always been critical issues. The book *Randomized Algorithms in Automatic Control and Data Mining* introduces the readers to the fundamentals of randomized algorithm applications in data mining (especially clustering) and in automatic control synthesis. The methods proposed in this book guarantee that the computational complexity of classical algorithms and the conservativeness of standard robust control techniques will be reduced. It is shown that when a problem requires "brute force" in selecting among options, algorithms based on random selection of alternatives offer good results with certain probability for a restricted time and significantly reduce the volume of operations.

Engineering
ISSN 1868-4394

ISBN 978-3-642-54785-0

9 783642 547850

▸ springer.com

ISRL 67

Granichin · Volkovich
Toledano-Kitai

Randomized Algorithms in Automatic Control
and Data Mining

Intelligent Systems Reference Library 67

Oleg Granichin
Zeev (Vladimir) Volkovich
Dvora Toledano-Kitai

Randomized Algorithms in Automatic Control and Data Mining

Springer

Figure: New book.

Thank you for your attention!