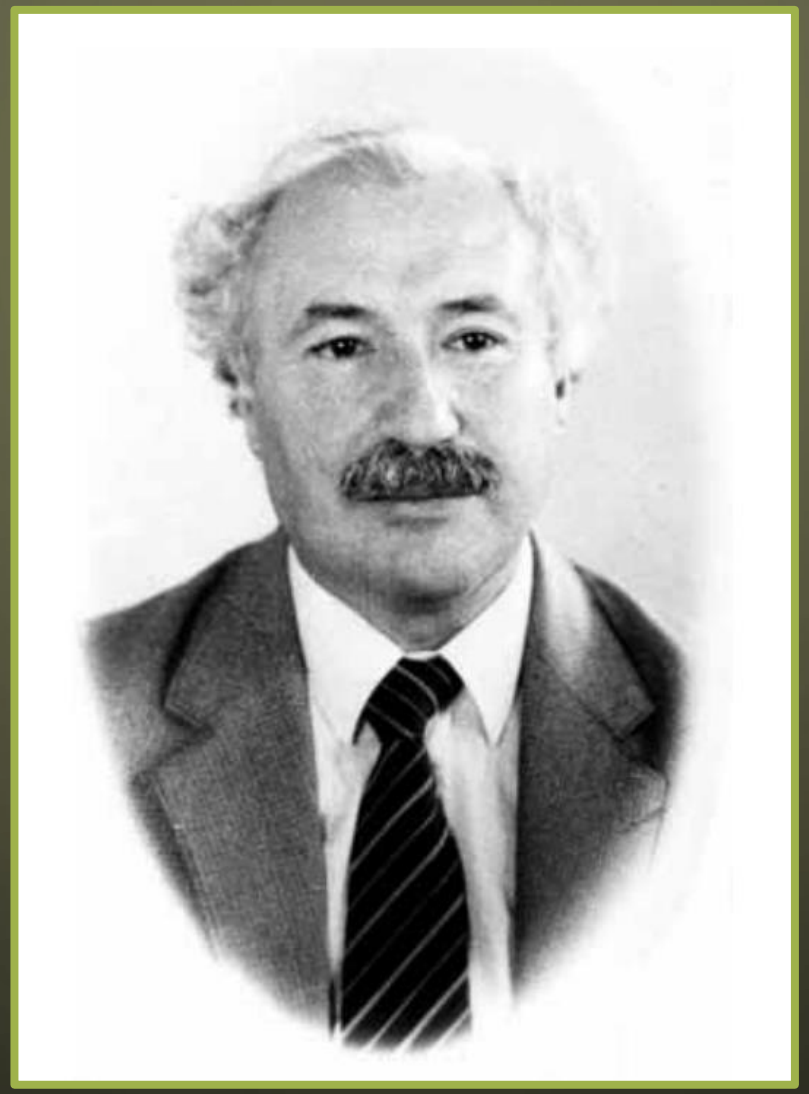
NON-SMOOTH OPTIMIZATION IN MACHINE LEARNING: REVIEW OF N.Z. SHOR'S METHODS AND THEIR APPLICATIONS

MYKOLA KORABLOV[1,2] M.M.KORABLOV@GMAIL.COM

[1]V.M. GLUSHKOV INSTITUTE OF CYBERNETICS OF THE NAS OF UKRAINE

[2]KYIV ACADEMIC UNIVERSITY

KAU DSS-2024: TOPICAL LECTURES ON MACHINE LEARNING, APRIL 13, 2024, KYIV, UKRAINE



Naum Zuselevych Shor (1937 – 2006)

Founder of Kyiv School of Nonsmooth Numerical Optimization

CONTENTS:

- Generalizing gradient descent using subgradients
  - Motivation for nonsmooth generalization of numerical optimization methods;
  - Attempts of such generalizations and arising problems;
  - Proper ways to generalize: main ideas behind Shor's subgradient descent methods;
- Improving convergence for ravine functions: Shor's space dilation operator
  - The problem of descending over ravine function;
  - Shor's space dilation operator;
  - Subgradient descent methods with space dilation:
    - Ellipsoid method;
    - Shor's r-algorithm;

CONTENTS:

- Application in supervised learning – regression
  - Quick overview of Least Squares Regression and why it is so popular;
  - Benefits of Least Moduli: situations when absolute errors save the day;
  - Example: nonsmooth regression model for searching defects in regular 3-D structures;
- Application in supervised learning – classification
  - Quick overview of Binary Linear Classification and Support Vector Machines;
  - In some problems you can not misclassify: testing linear separability of two sets of points;
  - Example: testing linear separability of "saw" dataset with gap $\varepsilon \to +0$;
- Concluding remarks

GENERALIZING GRADIENT DESCENT USING SUBGRADIENTS

MOTIVATION FOR NONSMOOTH OPTIMIZATION

- Main sources of nonsmooth optimization problems [1]:
  - Mathematical programming problems of large size that have block structure
  - Min-Max problems of the form $\min_{x \in D} \max_{i \in 1,\ldots,n} f_i(x)$
  - Nonlinear Programming problems that are solved using nonsmooth penalty functions
  - Optimal control problems with continuous/discrete time
  - Discrete or Discrete-Continuous Programming problems
- Also, from the purely numerical perspective, there is no such clear distinction between smooth and non-smooth functions: smooth functions that have quickly-oscilating gradients are “similar” to a nonsmooth functions in computation.

TWO DIRECTIONS OF RESEARCH

- Research towards solving special classes of problems that require minimizing a non-smooth objective function, the special structure of which is pre-defined (like Min-Max problems);
- Research towards developing general algorithms that are able to solve a vast amount of different types of nonsmooth problems, without the a priori knowledge about their structure:
  - Cutting hyperplanes methods;
  - Generalized gradient descent methods.

GRADIENT DESCENT: ATTEMPTING TO GENERALIZE

- Gradient descent method for minimization convex smooth function $f(x)$:

$$x_{k+1} = x_k - h_k \cdot \nabla f(x_k), k = 0,1,2, \ldots$$

x_0 – starting point

$\nabla f(x_k)$ – gradient of objective function, calculated at point x_k

h_k – step size on iteration k (learning rate)

- It would be nice to take this idea as a baseline for methods that minimize general type of convex objective function, smooth and non-smooth alike.
- However, it is not so easy…

GRADIENT DESCENT: ATTEMPTING TO GENERALIZE

- While trying to generalize gradient descent procedures for functions with discontinuous gradients, two main problems arise [1]:
  - Defining the analog of a gradient for those points, where ordinary derivatives do not exist, in such a manner that it can be easily used on practice;
  - Creating new ways to search for descent direction and step size, since ideas from smooth optimization will not work correctly when applied to nonsmooth functions minimization.

DEFINING THE ANALOG OF A GRADIENT

- There is such generalization – a subgradient $g_f(x)$ of a function $f(x)$;
- However, by definition there may exist infinitely many subgradients of a function at some point $\hat{x}$, leading to definition of a subgradient set:

$$G_f(\hat{x}) = \left\{ g_f(\hat{x}) : f(x) - f(\hat{x}) \geq \left\langle g_f(\hat{x}), x - \hat{x} \right\rangle \forall x \in \mathbb{R}^n \right\}$$

- Example:

$$f(x) = |x|; \quad G_f(0) = [-1; 1]$$

DEFINING THE ANALOG OF A GRADIENT

- The notion of a subgradient is a great theoretical generalization of a gradient for nonsmooth functions;
- But it can not be used in practice, mainly due to its non-uniqueness: if we have infinitely many subgradients at some point $\hat{x}$, which one of them to choose as a descent direction for an algorithm?

DEFINING THE ANALOG OF A GRADIENT

- Shor's approach [1]:
  - Restrict attention to some class of nonsmooth functions that excludes really "weird" ones that are hardly ever encountered in practice and define a generalization of a gradient for them ⇒ Shor's almost-gradients!
  - By definition all convex functions $f(x)$ have almost-gradients, and $\forall x \in \mathbb{R}^n$ such almost-gradient is some element of a subgradient set.
- But will they be suitable as a descent direction?
- **N.B.** further the term "subgradient" will actually mean "almost-gradient"

SEARCHING FOR DESCENT DIRECTION

- Shor's approach [1]:
  - Let $f(x) \in \mathbb{R}$ be a convex function $\forall x \in \mathbb{R}^n$, and let $x^* \in X^*$ be an element from set $X^* = \left\{x^* : f(x^*) = \min\limits_{x \in \mathbb{R}^n} f(x)\right\}$
  - By def. of subgradient at point $\hat{x}$:

$$f(x) - f(\hat{x}) \geq \langle g_f(\hat{x}), x - \hat{x} \rangle \quad \forall x \in \mathbb{R}^n$$

  - If $f(x) < f(\hat{x})$, then:

$$\langle -g_f(\hat{x}), x - \hat{x} \rangle > 0 \quad \forall x \in \mathbb{R}^n$$

SEARCH FOR DESCENT DIRECTION

- Geometric meaning of $\langle -g_f(\hat{x}), x - \hat{x} \rangle > 0 \quad \forall x \in \mathbb{R}^n$:
  - Anti-subgradient $-g_f(\hat{x})$ at point $\hat{x}$ forms an acute angle with any direction $x - \hat{x}$ from point $\hat{x}$ to such point $x \in \mathbb{R}^n$, where the value of $f(\cdot)$ is smaller;
  - Thus, if $X^* \neq \emptyset$ and $\hat{x} \notin X^*$, then moving from point $\hat{x}$ in the direction $-g_f(\hat{x})$ with a small enough step will reduce the distance to X^*!
- This simple fact is the main idea behind subgradient descent methods for nonsmooth functions minimization!

GENERALIZED GRADIENT DESCENT METHOD [2]

- a.k.a. subgradient descent, is a subgradient procedure for minimizing convex functions

$$x_{k+1} = x_k - h_k \cdot \frac{g_f(x_k)}{\|g_f(x_k)\|}, k = 0,1,2, \dots$$

x_0 – starting point

$g_f(x_k)$– subgradient of objective function, calculated at point x_k

h_k – step size on iteration k

- The method looks similar to smooth gradient descent, but it works differently;
- Also, we still need to choose step sizes h_k…

GENERALIZED GRADIENT DESCENT METHOD

- Theorem:

  Let $f(x)$ be a convex function with a bounded set X^*. Let also $\{h_k\}_{k=0}^{\infty}$ be a sequence s.t. $h_k > 0, \lim\limits_{k\to\infty} h_k = 0, \sum_{k=0}^{\infty} h_k = +\infty$;

  Then for $\forall x_0 \in \mathbb{R}^n$ for the sequence $\{x_k\}_{k=0}^{\infty}$, obtained from subgradient descent method, one of the following statements holds:

  - $\exists k = k^* \; s.t. \; x_{k^*} \in X^*$
  - $\lim\limits_{k\to\infty} \rho_k = 0, \lim\limits_{k\to\infty} f(x_k) = \min\limits_{x\in\mathbb{R}^n} f(x)$, where $\rho_k = \min\limits_{x\in X^*} \|x_k - x\|$

- Are there other ways to define h_k? Yes! (Polyak's step, Shor's adaptive step,...)

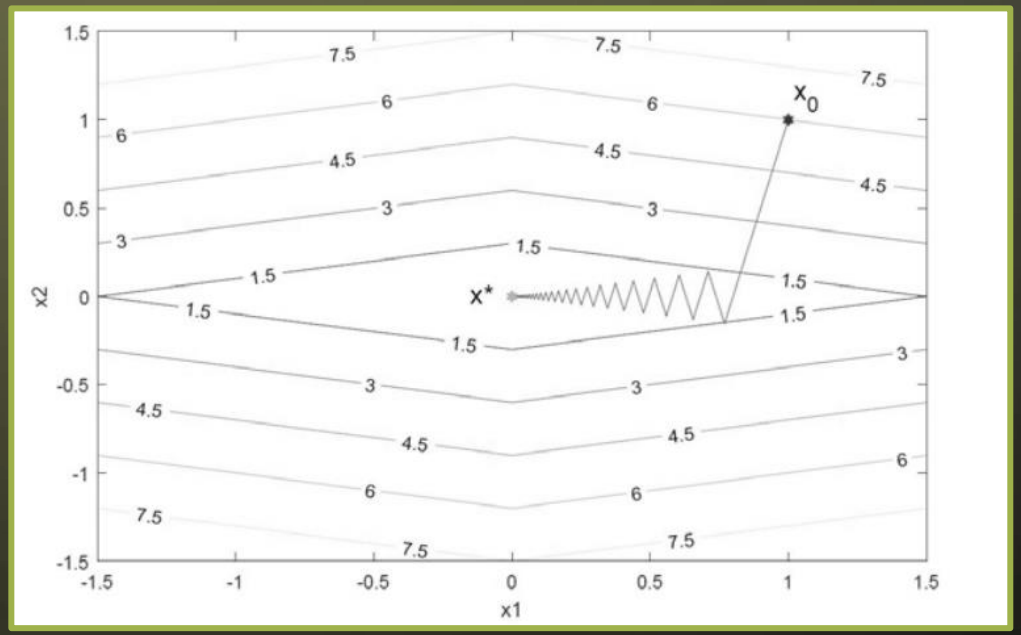
IMPROVING CONVERGENCE FOR RAVINE FUNCTIONS: SHOR'S SPACE DILATION OPERATOR

RAVINE FUNCTION

- A function of several real variables whose graph near a minimum has a ravine-type shape. Such functions cause difficulties in minimization problems
- Example:

$$f(x_1, x_2) = |x_1| + 5|x_2|$$

$$x_0 = (1,1), x^* = (0,0)$$



- It would be nice to help the procedure converge faster... Let's warp the space!

SPACE DILATION – MAIN IDEA [1]

- Let's make the following change of variables on k-th iteration of the subgradient method:

$$y = A_k x \Rightarrow x = B_k y, B_k = A_k^{-1}$$

- As we know, for the subgradient of a convex function $f(x)$ at point x_k, the following inequality holds:

$$f(x) \geq f(x_k) + \langle g_f(x_k), x - x_k \rangle \quad \forall x \in \mathbb{R}^n$$

SPACE DILATION – MAIN IDEA

- Substituting $x = B_k y$, we get:

$$\varphi(y) \geq \varphi(y_k) + \langle B_k^T g_f(x_k), y - y_k \rangle \; \forall y \in \mathbb{R}^n$$

- We see that $g_\varphi(y_k) = B_k^T g_f(x_k)$ satisfies an inequality:

$$\varphi(y) \geq \varphi(y_k) + \langle g_\varphi(y_k), y - y_k \rangle \; \forall y \in \mathbb{R}^n$$

thus showing us that $g_\varphi(y_k)$ is a subgradient of the convex function $\varphi(y) = f(B_k y)$ at point $y_k = A_k x_k$ of the transformed space of variables $y = A_k x$

SPACE DILATION – MAIN IDEA

- Now let's apply the subgradient procedure for minimizing $\varphi(y)$;
- In transformed space of variables $y = A_k x$ it will take form:

$$y_{k+1} = y_k - h_k \frac{g_\varphi(y_k)}{\|g_\varphi(y_k)\|} = y_k - h_k \frac{B_k^T g_f(x_k)}{\|B_k^T g_f(x_k)\|}$$

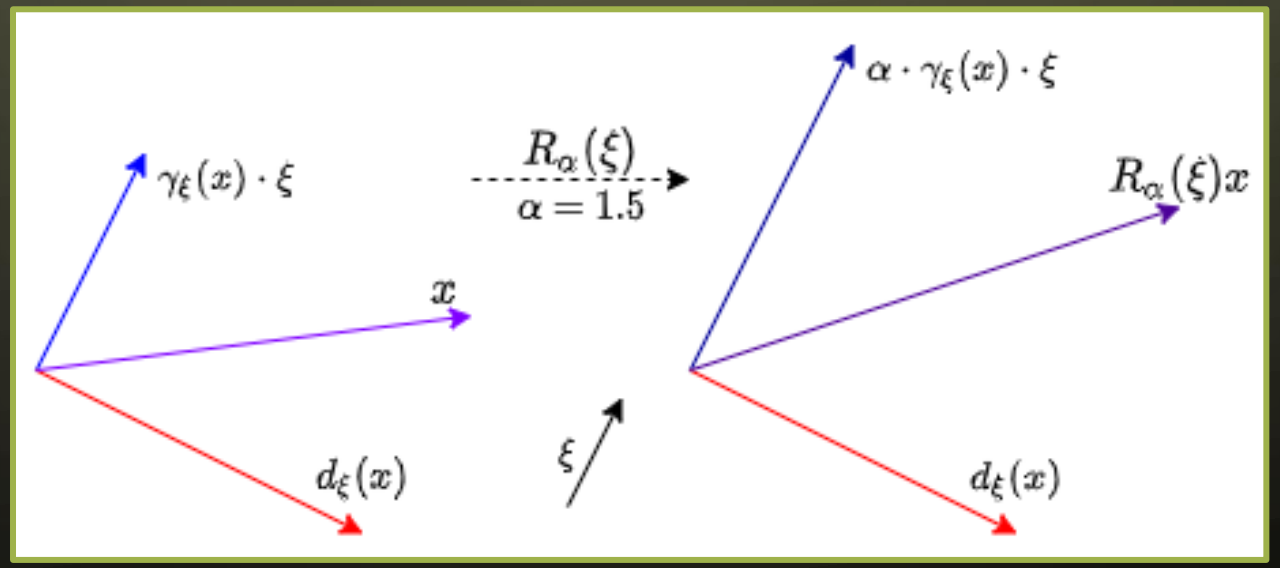
- And in the original space $x = B_k y$ it will become:

$$x_{k+1} = B_k y_{k+1} = B_k y_k - h_k B_k \frac{B_k^T g_f(x_k)}{\|B_k^T g_f(x_k)\|} = x_k - h_k B_k \frac{B_k^T g_f(x_k)}{\|B_k^T g_f(x_k)\|}$$

- Now we only need to define the way for iterative updating B_k
- The question is – what transformation of space is useful?

SHOR'S SPACE DILATION OPERATOR [1]

- Let's fix a vector $\xi \in \mathbb{R}^n, \|\xi\|_2 = 1$ and number $\alpha > 0$
- Then for $\forall x \in \mathbb{R}^n$ we have $x = \gamma_\xi(x) \cdot \xi + d_\xi(x)$, where $\langle \xi, d_\xi(x) \rangle = 0$
- Space dilation operator for space $\mathbb{R}^n$ in direction ξ with coefficient α is defined as $R_\alpha(\xi)x = \alpha \cdot \gamma_\xi(x) \cdot \xi + d_\xi(x)$ for $\forall x \in \mathbb{R}^n$



SHOR'S SPACE DILATION OPERATOR

- Some properties of the operator $R_\alpha(\xi)$:
  - Linear, symmetric;
  - Matrix form: $R_\alpha(\xi) = I + (\alpha - 1)\xi\xi^T$;
  - $R_{\alpha\beta}(\xi) = R_\alpha(\xi)R_\beta(\xi)$;
  - $R_\alpha(\xi)R_{\frac{1}{\alpha}}(\xi) = I$;
  - $R_0(\xi)$ is a projection operator on $\{\xi\}^\perp$

- So, which direction ξ we need to choose?

ELLIPSOID METHOD FOR CONVEX FUNCTIONS [3]

- Obtained using space dilation in direction of a subgradient and a certain choice of dilation coefficient and step size:

$$x_{k+1} = x_k - h_k B_k \xi_k, \quad \xi_k = \frac{B_k^T g_f(x_k)}{\|B_k^T g_f(x_k)\|}, \quad h_k = \frac{1}{n+1} r_k, \quad k = 0,1, \ldots$$

$$B_{k+1} = B_k R_{\beta_k}(\xi_k), \quad \beta_k = \sqrt{\frac{n-1}{n+1}}, \quad r_{k+1} = \frac{n}{\sqrt{n^2-1}} r_k$$

x_0 – starting point;

$r_0 > 0$ – radius of initial localizing ball, $\|x_0 - x^*\| \leq r_0$;

$B_0 = I_n$ – initial inverse space transformation matrix

ELLIPSOID METHOD FOR CONVEX FUNCTIONS

- Theorem:

  Let $\{x_k\}_{k=0}^{\infty}$ be the sequence of points generated by ellipsoid method for convex function. Then, the ratio of volumes of the localizing ellipsoids $\mathcal{E}_k$ and $\mathcal{E}_{k+1}$ obtained on iterations k and $k+1$ does not depend on k and equals

$$q_n = \frac{vol(\mathcal{E}_{k+1})}{vol(\mathcal{E}_k)} = \frac{n}{n+1}\left(\frac{n}{\sqrt{n^2-1}}\right)^{n-1} < e^{\frac{1}{2(n+1)}} < 1.$$

  Moreover, $x^* \in \mathcal{E}_k$ for $\forall k = 0,1,\dots,k^*$

ELLIPSOID METHOD FOR CONVEX FUNCTIONS

- Interesting facts:
  - The ellipsoid method was independently created by Nemirovski and Yudin [4] using completely different approach, while Shor came up with it as a type of subgradient descent method with space dilation
  - The ellipsoid method was used by Khachiyan [5] to construct and justify first polynomial algorithm for Linear Programming problems with rational coefficients, thus disproving their NP-hardness!
- However, you may find papers arguing that ellipsoid method is useless in practice and doesn't work even for functions of 2-5 variables. Why is that?
  - This algorithm has two forms: B-form that is computationally stable (mentioned above), and H-form, which is not. And many people have been using the wrong one for decades!

SHOR'S R-ALGORITHM [6]

- Obtained using space dilation in direction of a difference of two subsequent subgradients:

$$x_{k+1} = x_k - h_k B_k \xi_k, \quad B_{k+1} = B_k R_{\beta_k}(\eta_k), k = 0,1, \dots$$

$$\xi_k = \frac{B_k^T g_f(x_k)}{\|B_k^T g_f(x_k)\|}, \quad h_k \geq h_k^* = \underset{h \geq 0}{argmin}\, f(x_k - h B_k \xi_k)$$

$$\eta_k = \frac{B_k^T r_k}{\|B_k^T r_k\|}, \quad r_k = g_f(x_{k+1}) - g_f(x_k), \quad \beta_k = \frac{1}{\alpha} < 1$$

- A very powerful tool that allows to solve many hard optimization problems
- However, its convergence is theoretically proved only for some special cases

APPLICATION IN SUPERVISED LEARNING: REGRESSION

QUICK OVERVIEW: REGRESSION [7]

- Let $\left\{\left(x_1^{(j)}, \ldots, x_d^{(j)}, y^{(j)}\right) \in \mathbb{R}^{d+1} : j = \overline{1,n}\right\}$ be a dataset of size n, where for every measurement $j = \overline{1,n}$ observed values $y^{(j)}$ are somehow dependent on values of d factors $x_1^{(j)}, \ldots, x_d^{(j)}$.
- The regression problem is to use available data to build a model that:
  - Describes dependencies between observed values $y^{(j)}$ and factors $x_i^{(j)}$ good enough;
  - Is suitable for predicting values y corresponding to new, "unseen" factors x_i;

QUICK OVERVIEW: REGRESSION

- In the easiest case we assume that such dependence is linear on parameters of the model w, i.e.:

$$y = f(x_1, \dots, x_d) = \sum_{k=1}^{m} w_k \psi_k(x_1, \dots, x_d)$$

where $\psi_k, k = \overline{1, m}$ are some basis functions

- To find suitable parameter values w, i.e. construct a model that is good enough at approximating dependencies between y and x based on available data, we have to:
  - Pick basis functions (plain factors, polynomials, exponents, etc.);
  - Minimize observation errors $\varepsilon^{(j)} = y^{(j)} - \hat{y}^{(j)} = y^{(j)} - f\left(x_1^{(j)}, \dots, x_d^{(j)}\right), j = \overline{1, n}$;

QUICK OVERVIEW: LEAST SQUARES [7]

- The most popular way to solve linear regression problems:
  - Is statistically derived from assumptions that factors are independent and errors follow a Normal distribution $N(0, \sigma^2 I_n)$ using maximum likelihood principle;
  - An exact solution can be obtained using orthogonal projections in Hilbert space and such solution has great properties by Gauss-Markov theorem;
- Most of the time regression problems are treated from a functional approximation perspective, solving convex smooth optimization problem:

$$\min_{w \in \mathbb{R}^m} \sum_{j=1}^{n} \left(\varepsilon^{(j)}\right)^2 = \min_{w \in \mathbb{R}^m} \sum_{j=1}^{n} \left(y^{(j)} - \hat{y}^{(j)}\right)^2$$

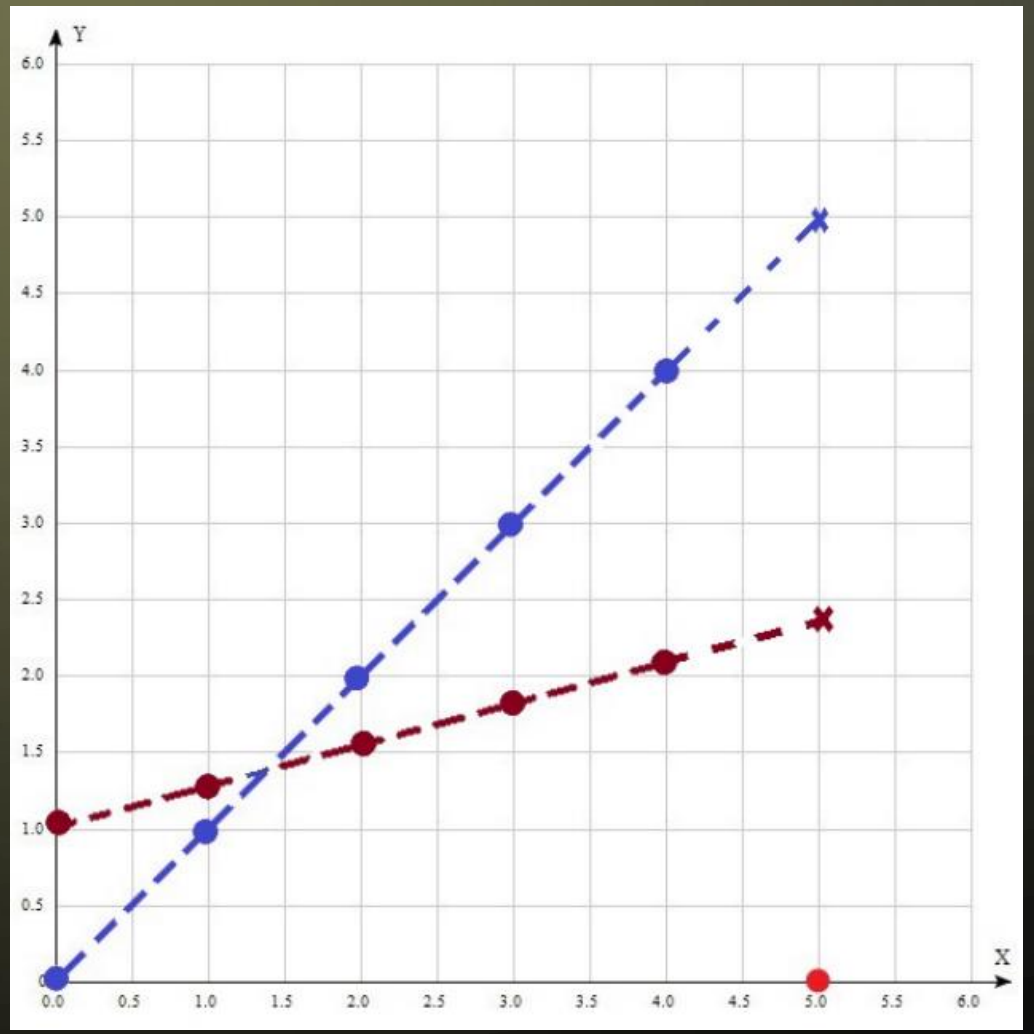
- It became a "state-of-the-art" approach, but many troubles usually appear applying it in practice

REGRESSION IN PRACTICE – DATA WITH OUTLIERS

- A "childish" example:
  - One-dimensional, six points

x	0	1	2	3	4	5
y	0	1	2	3	4	0

  - Red line – Least Squares estimation and its predicted $\hat{y}$ values
  - Blue line – correct estimation



PROBLEMS AND SOLUTIONS [7]

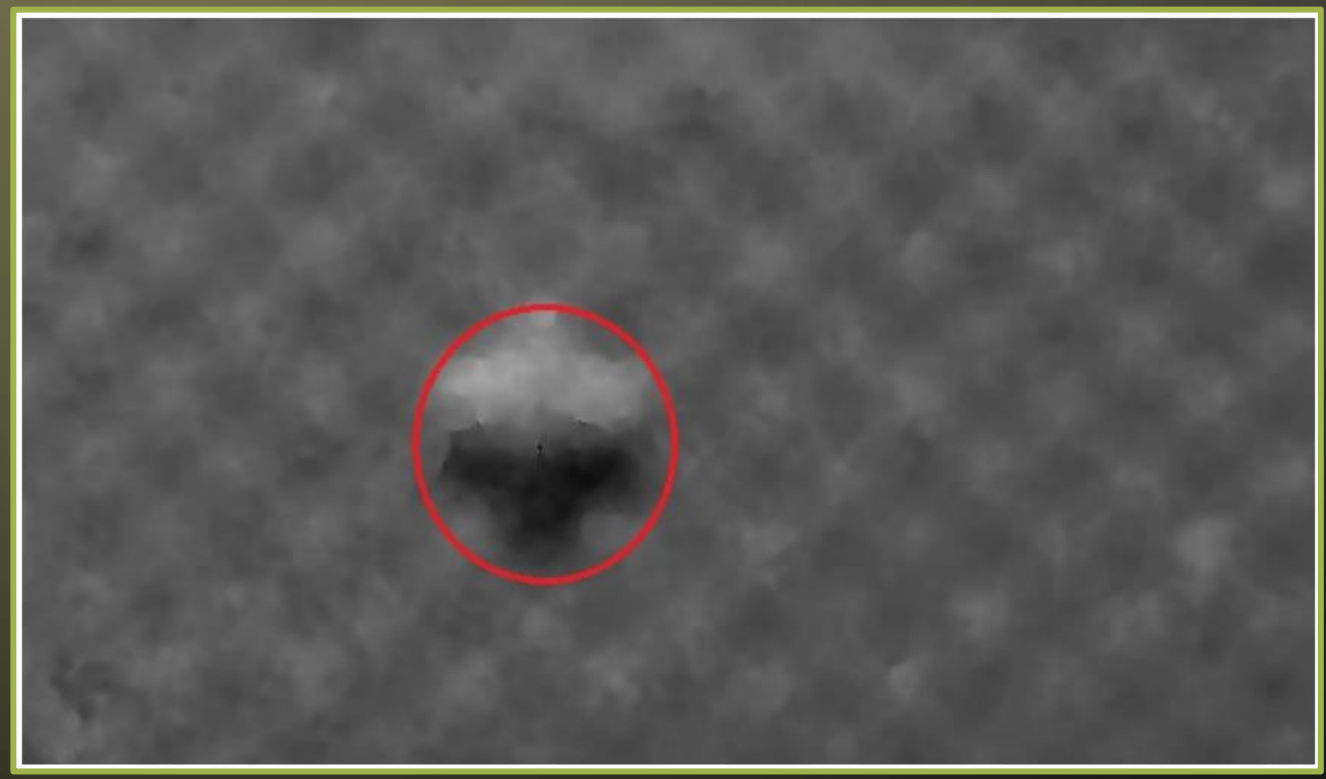
- Most problems in practice arise when:
  - We work with datasets of small size (e.g. medicine applications);
  - We work with datasets with many outliers or dependent features;
- One popular way out is regularization:
  - Ridge: $\min_{w \in \mathbb{R}^m} \sum_{j=1}^{n} \left(y^{(j)} - \hat{y}^{(j)}\right)^2 + \lambda \sum_{k=1}^{m} w_k^2$
  - Lasso: $\min_{w \in \mathbb{R}^m} \sum_{j=1}^{n} \left(y^{(j)} - \hat{y}^{(j)}\right)^2 + \lambda \sum_{k=1}^{m} |w_k|$
- But there is another great statistically justified way – Least Moduli (LM):

$$\min_{w \in \mathbb{R}^m} \sum_{j=1}^{n} \left|\varepsilon^{(j)}\right| = \min_{w \in \mathbb{R}^m} \sum_{j=1}^{n} \left|y^{(j)} - \hat{y}^{(j)}\right|$$

- It does require direct solving a nonsmooth problem though…

EXAMPLE: DEFECTS IN REGULAR 3-D STRUCTURES

- Research project by scientists from V.M. Glushkov Institute of Cybernetics and E.O. Paton Electric Welding Institute [8];
- Brief description: Create software for automatic non-destructive quality control (NDQ) of thin-walled multi-layer composite materials



REGULAR 3-D STRUCTURES

- A triple $\{A;\ u;\ v\}$ is called a regular 3-D structure, if $A \in \mathbb{R}^{m \times n}$, $u \in \mathbb{R}^m$, $v \in \mathbb{R}^n$, and $\forall i = \overline{1,m}, \forall j = \overline{1,n}$: $a_{ij} = u_i + v_j$;
- An elementary defect in a regular 3-D structure $\{A;\ u;\ v\}$ is such pair of indices (i,j) that $a_{ij} \neq u_i + v_j$;
- Suppose we have a matrix A that represents an image of a piece that's being checked for defects. The problem is to find such parameters u and v that coefficients $u_i + v_j$ have the smallest deviation from corresponding values a_{ij}

REGULAR 3-D STRUCTURES

- The smooth minimization problem (Least Squares):

$$\min_{u\in\mathbb{R}^m, v\in\mathbb{R}^n} \sum_{i=1}^{m}\sum_{j=1}^{n}\left(a_{ij}-u_i-v_j\right)^2$$

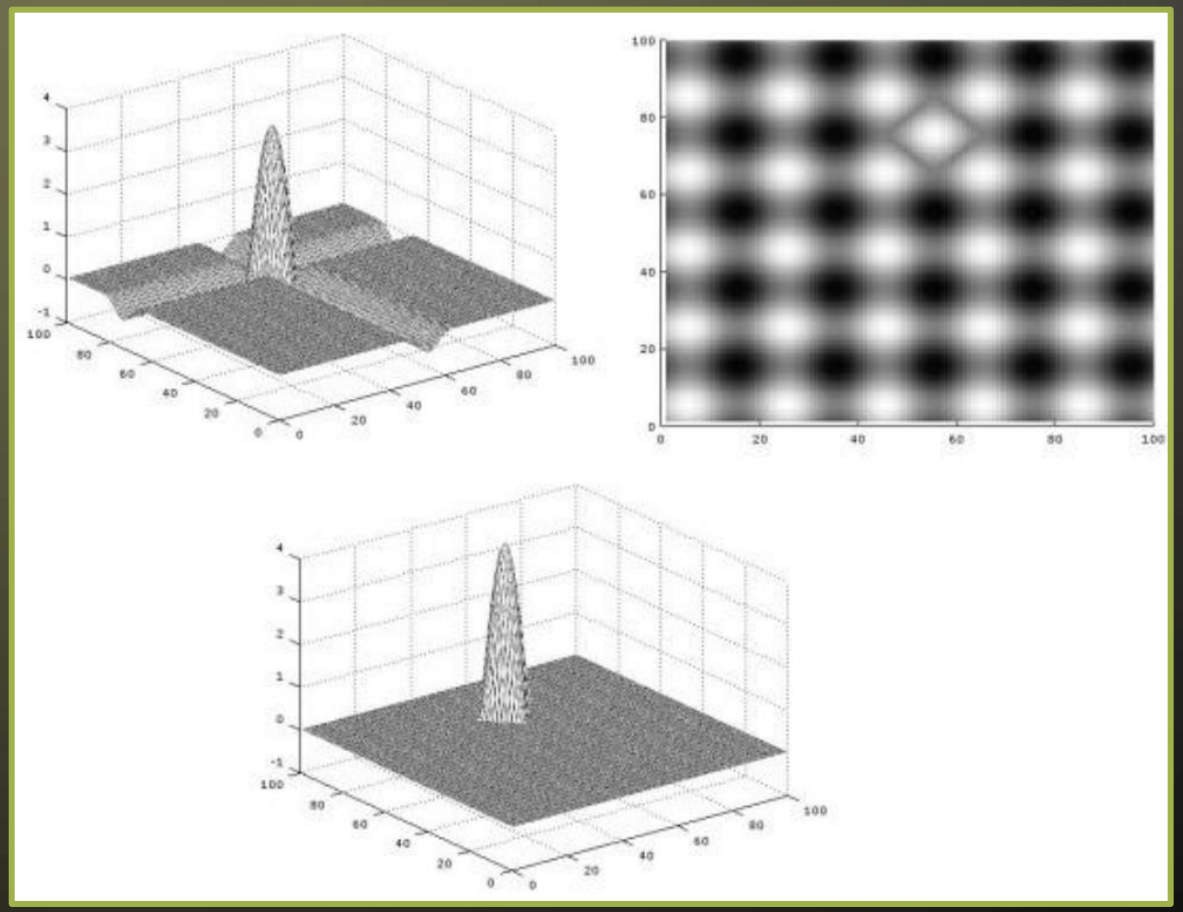
- The nonsmooth minimization problem (Least Moduli):

$$\min_{u\in\mathbb{R}^m, v\in\mathbb{R}^n} \sum_{i=1}^{m}\sum_{j=1}^{n}\left|a_{ij}-u_i-v_j\right|$$

- Performance of both models have been compared on various images, corresponding optimization problems were solved using r-algorithm

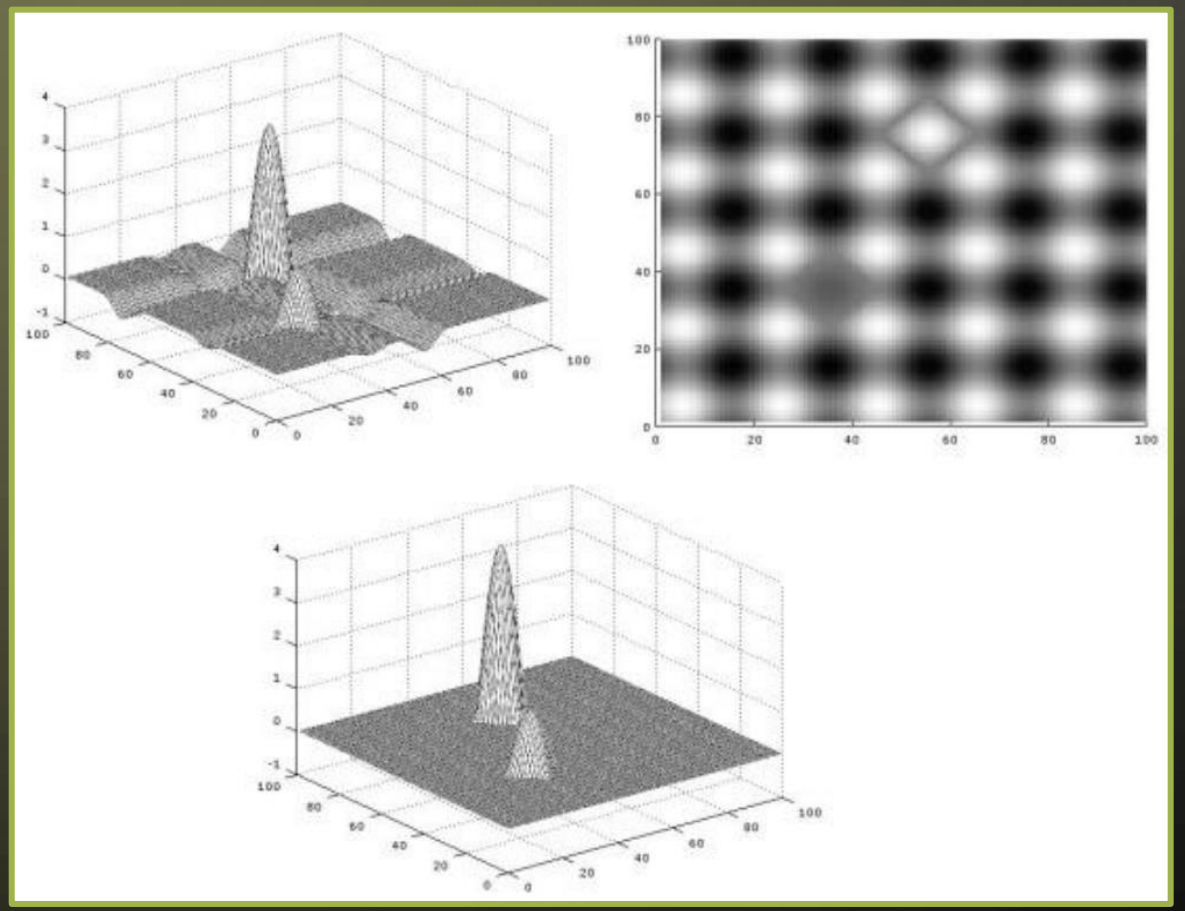
3-D STRUCTURE WITH 1 DEFECT

Defect detection with LS (top left),
the image itself (top right),
defect detection with LM (bottom)



3-D STRUCTURE WITH 2 DEFECTS

Defect detection with LS (top left),
the image itself (top right),
defect detection with LM (bottom)



APPLICATION IN SUPERVISED LEARNING: CLASSIFICATION

QUICK OVERVIEW: BINARY CLASSIFICATION [9]

- Let $\{(x_i, y_i), x_i \in \mathbb{R}^m, y_i \in \{\pm 1\}: i = \overline{1,n}\}$ be a dataset of size n, where for every measurement $i = \overline{1,n}$ elements x_i belong to one of two classes, denoted by values y_i
- The problem of linear classification is to use available data to find such a hyperplane $\langle w, x \rangle + b = 0$ dividing two classes that has the biggest margin:

$$\max_{w \in \mathbb{R}^m, b \in \mathbb{R}, r > 0} r$$

$$y_i \cdot (\langle w, x_i \rangle + b) \geq r, \ i = \overline{1,n}$$

$$\|w\| = 1$$

QUICK OVERVIEW: SUPPORT VECTOR MACHINES

- Using scaling we can set $r = 1$ w.l.o.g. and consider corresponding models [9]:
- Hard SVM forbids misclassifications:

$$\min_{w \in \mathbb{R}^m, b \in \mathbb{R}} \frac{1}{2} \|w\|^2$$

$$y_i \cdot (\langle w, x_i \rangle + b) \geq 1, \quad i = \overline{1, n}$$

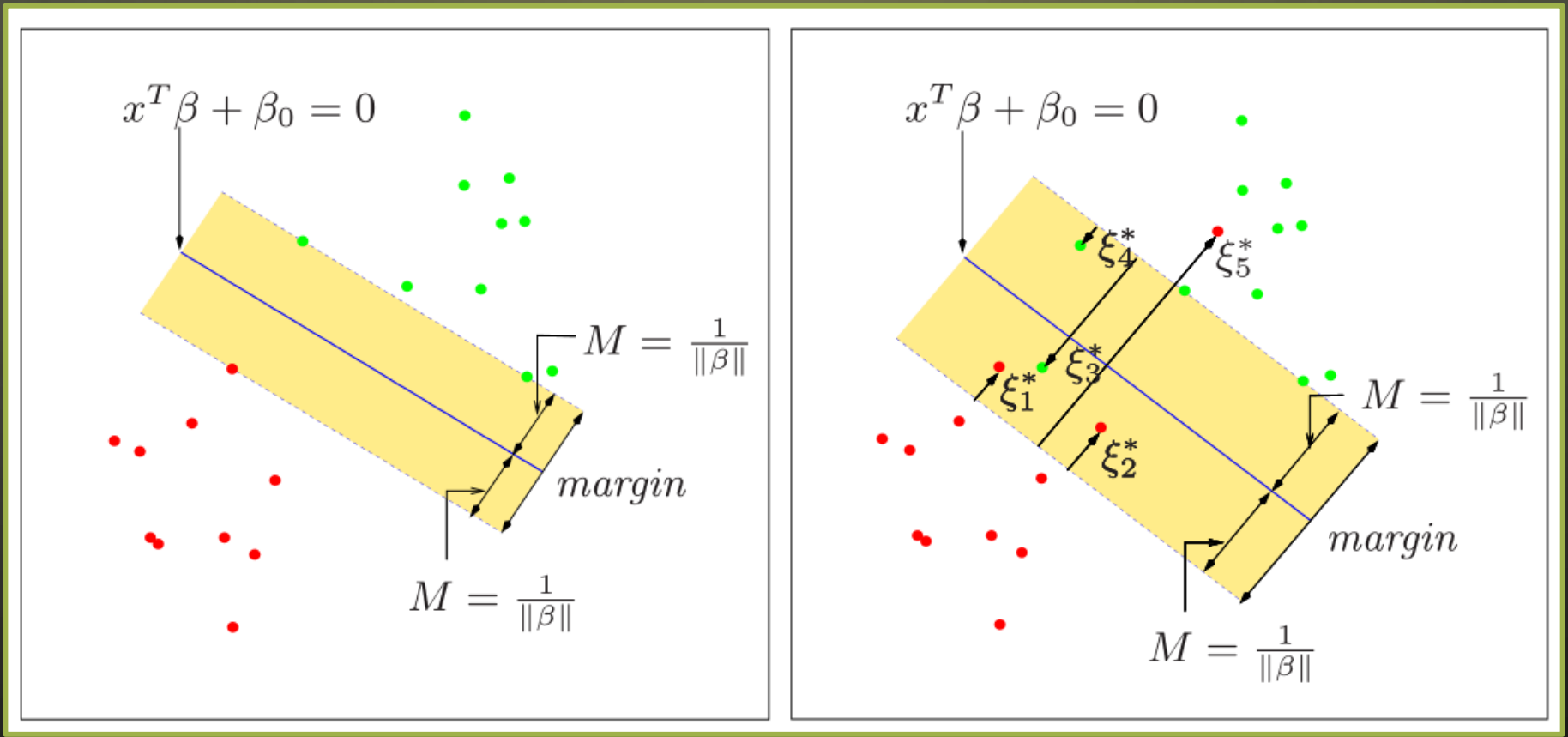
- Soft SVM allows some misclassifications:

$$\min_{w \in \mathbb{R}^m, b \in \mathbb{R}} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

$$y_i \cdot (\langle w, x_i \rangle + b) \geq 1 - \xi_i, \quad i = \overline{1, n}$$

$$\xi_i \geq 0, \qquad i = \overline{1, n}$$

COMPARING HARD SVM AND SOFT SVM

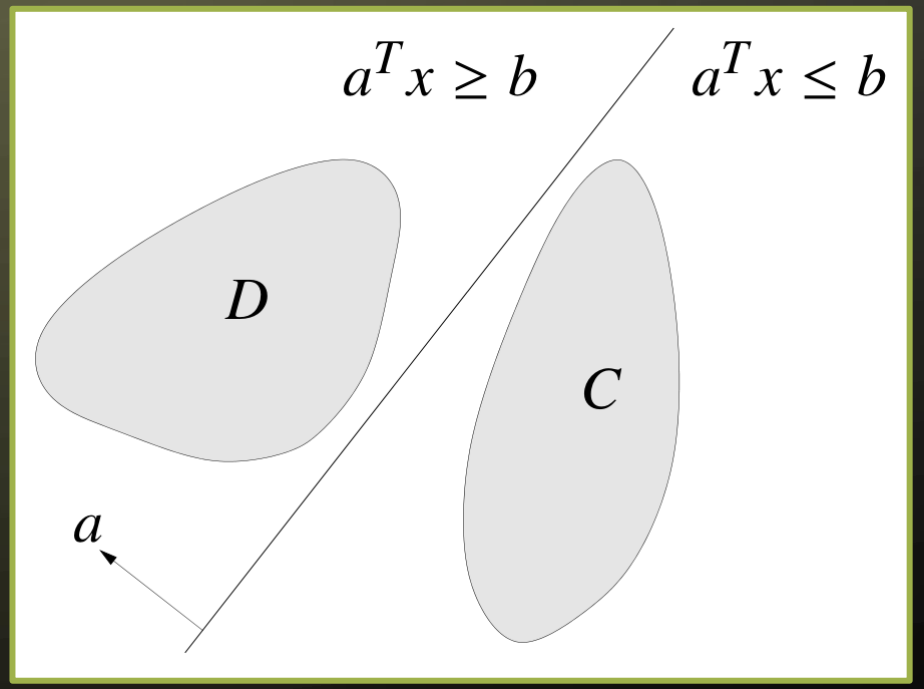


Hard SVM classification (left), Soft SVM classification (right) [7]

(in our notation $M = r, \beta = w, \beta_0 = b$)

LINEAR SEPARABILITY OF SETS

- In order to apply linear classification and expect it to work correctly, we firstly need to know whether the classes are linearly separable
- It is a well known theoretical question in mathematical programming [10], and many great theorems exist for convex sets
- Example:
  - Hyperplane $\{x: \langle a, x \rangle = b\}$ separates C and D



LINEAR SEPARABILITY OF SETS

- But in practice, we do not know whether two "clouds of datapoints" indeed form disjoint convex sets – and thus can be separated
- So the need arises for some numerical algorithm that will be able to test linear separability of a given labeled dataset
- In most cases, Hard SVM will tackle this task just fine:
  - If you can solve it with zero training error, then linear separation exists
  - If the Hard SVM does not have a solution, then the dataset is not linearly separable
- However, this approach is not reliable

MAXIMUM MARGIN LINEAR CLASSIFIER [11]

- Problem: to find such hyperplane $\langle w, x \rangle + b = 0$ that there will be no datapoints between $\langle w, x \rangle + b = 1$ and $\langle w, x \rangle + b = -1$:

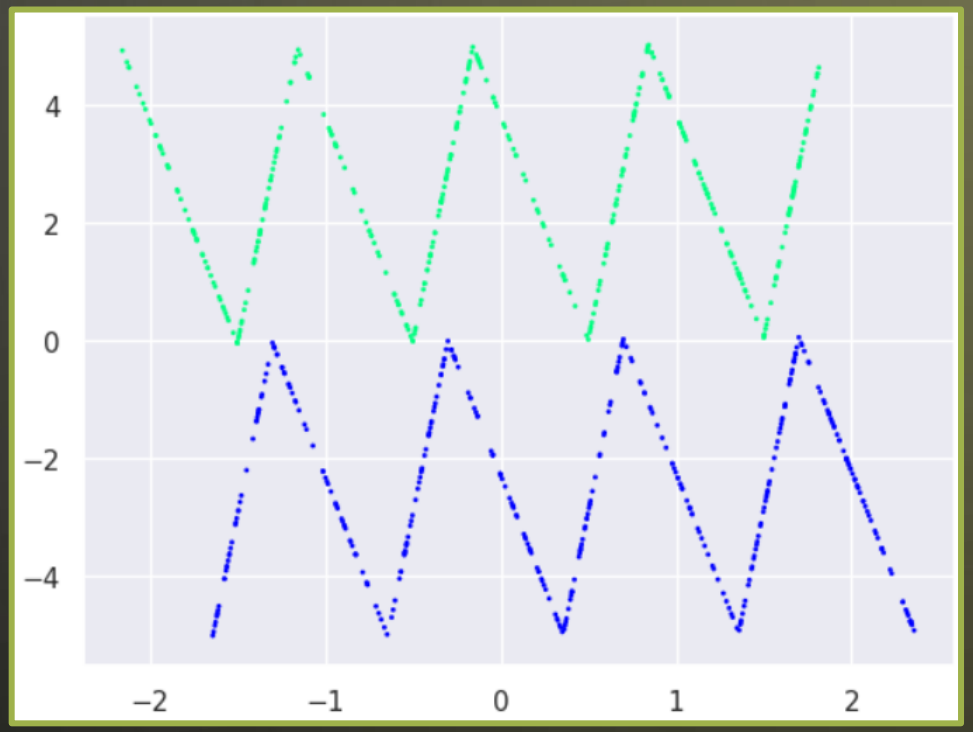
$$\min_{w \in \mathbb{R}^m, b \in \mathbb{R}} \left\{ \max_{i \in \overline{1,n}} -y_i(\langle w, x_i \rangle + b) \right\}$$

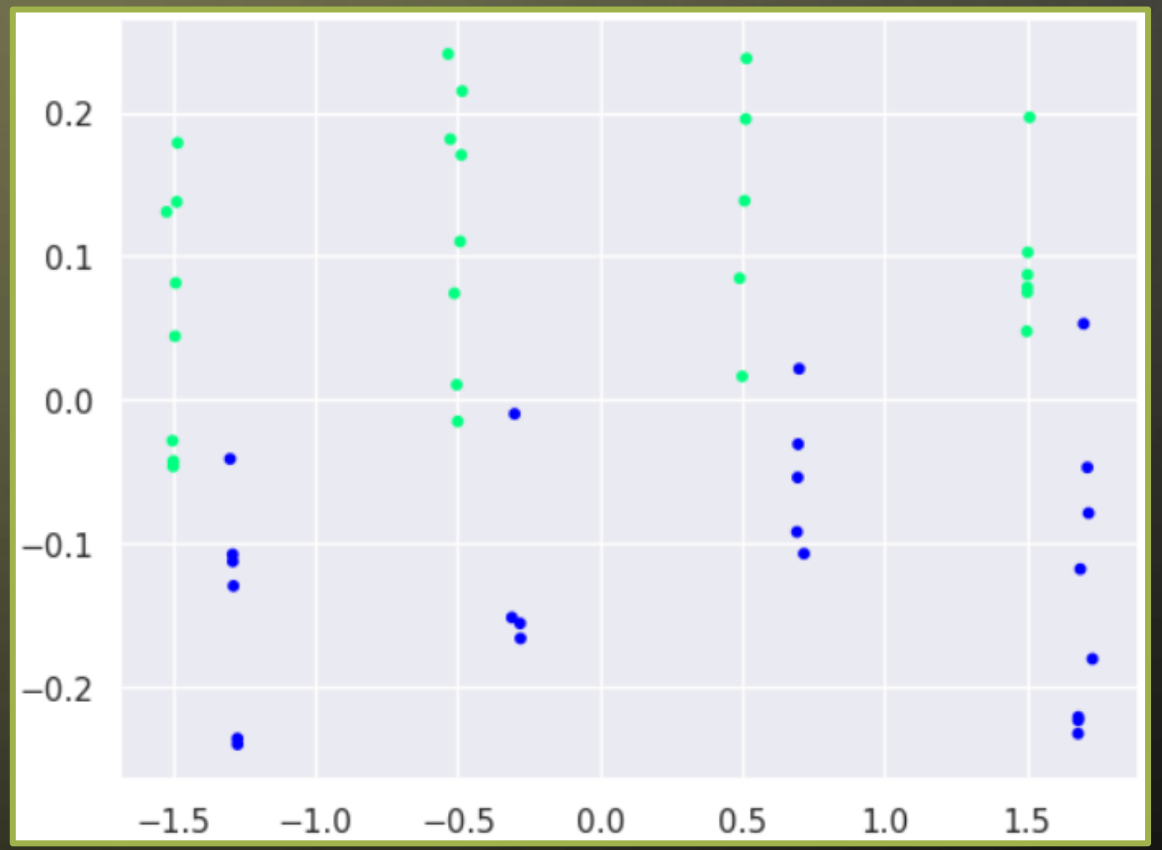
$$\|w\|^2 \leq 1$$

- This problem is nonsmooth and constrained, and can be solved using ellipsoid method and nonsmooth penalty function [12]:

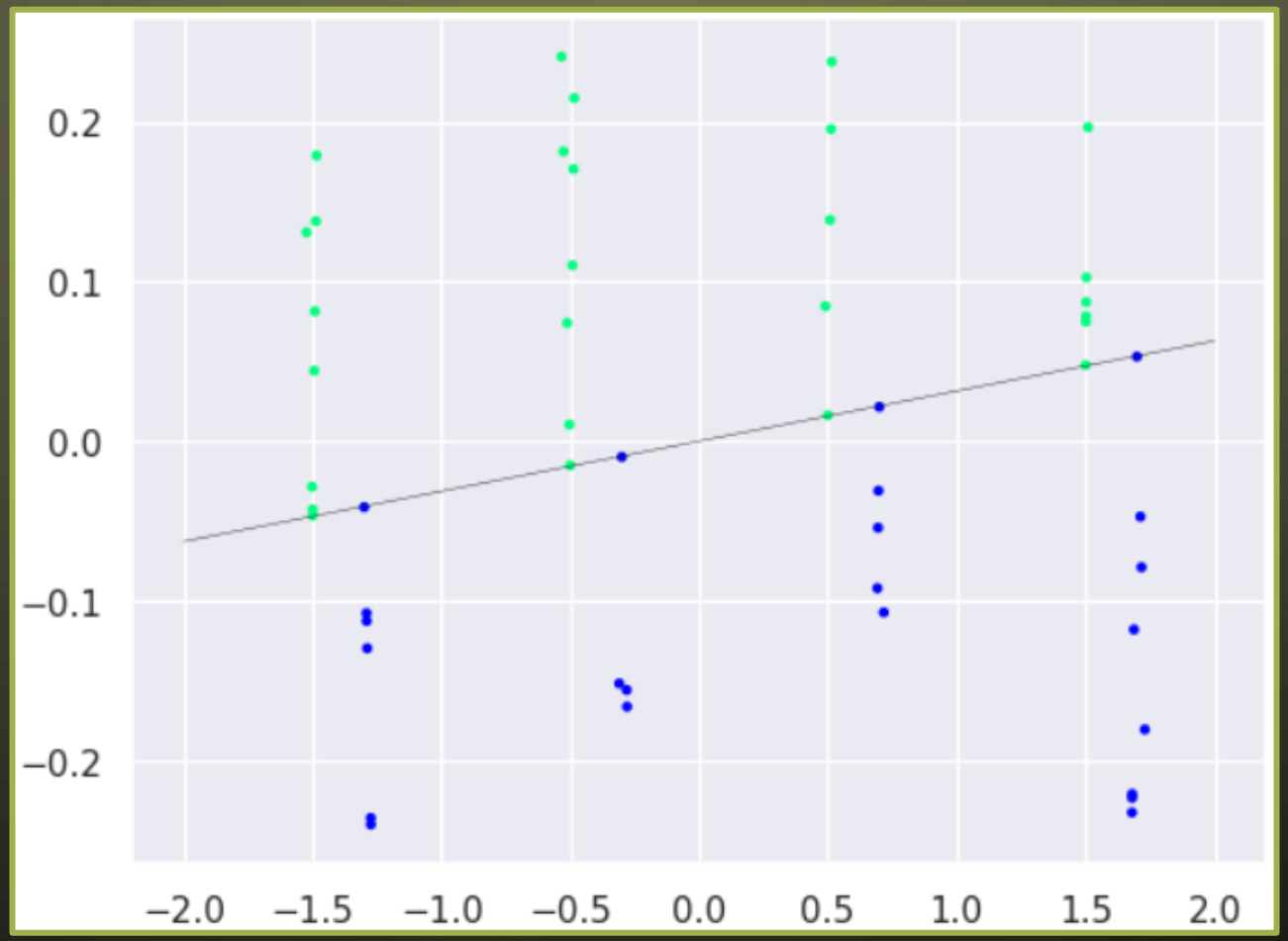
$$\min_{w \in \mathbb{R}^m, b \in \mathbb{R}} \left\{ \max_{i=1,..,n} \{-y_i(\langle w, x_i \rangle + b)\} + P \cdot max\left\{0, \sum_{j=1}^{m} w_j^2 - 1\right\} \right\}$$

"SAW" DATASET WITH GAP $\varepsilon = 10^{-3}$



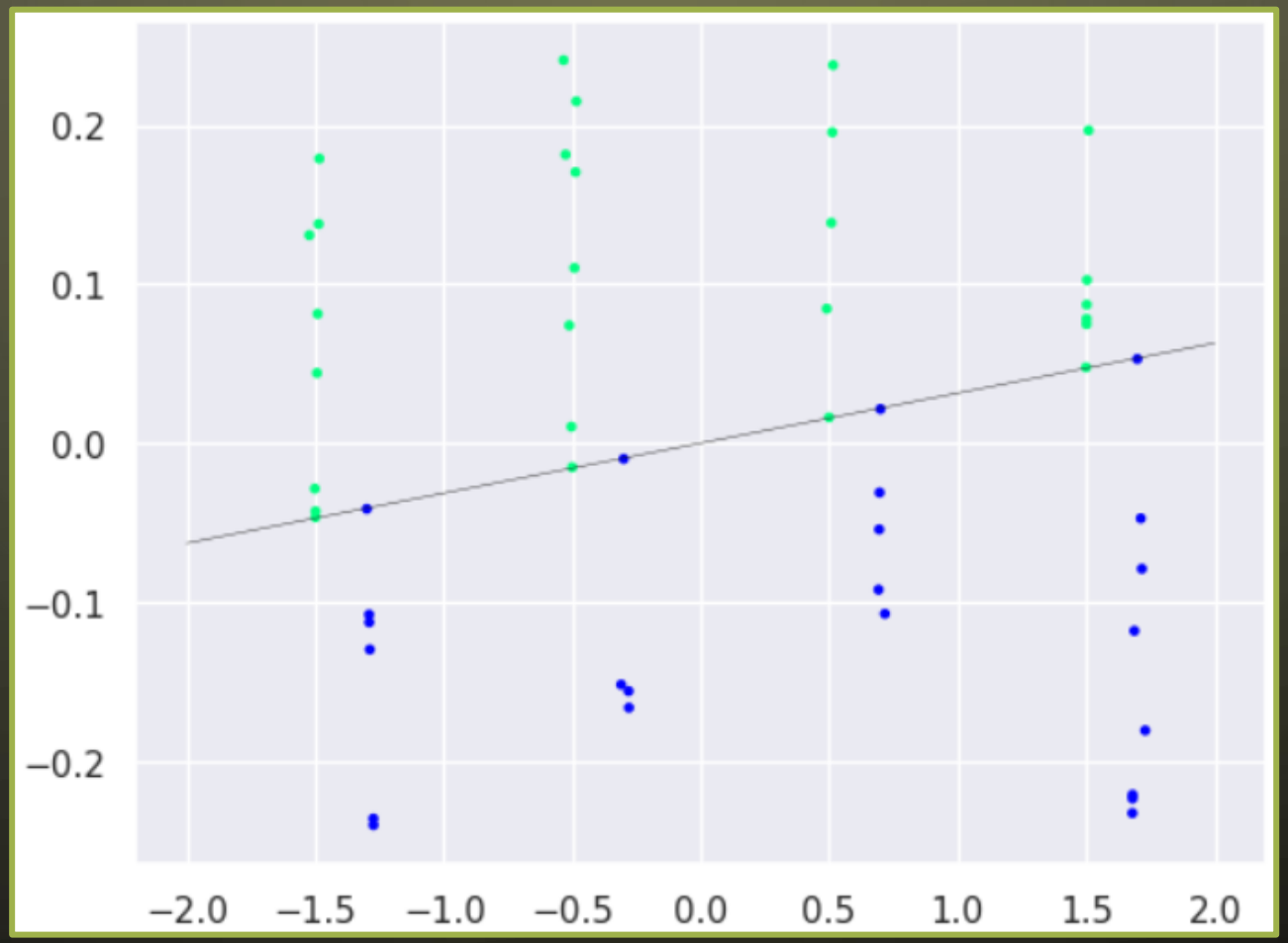


"SAW" DATASET WITH GAP $\varepsilon = 10^{-3}$



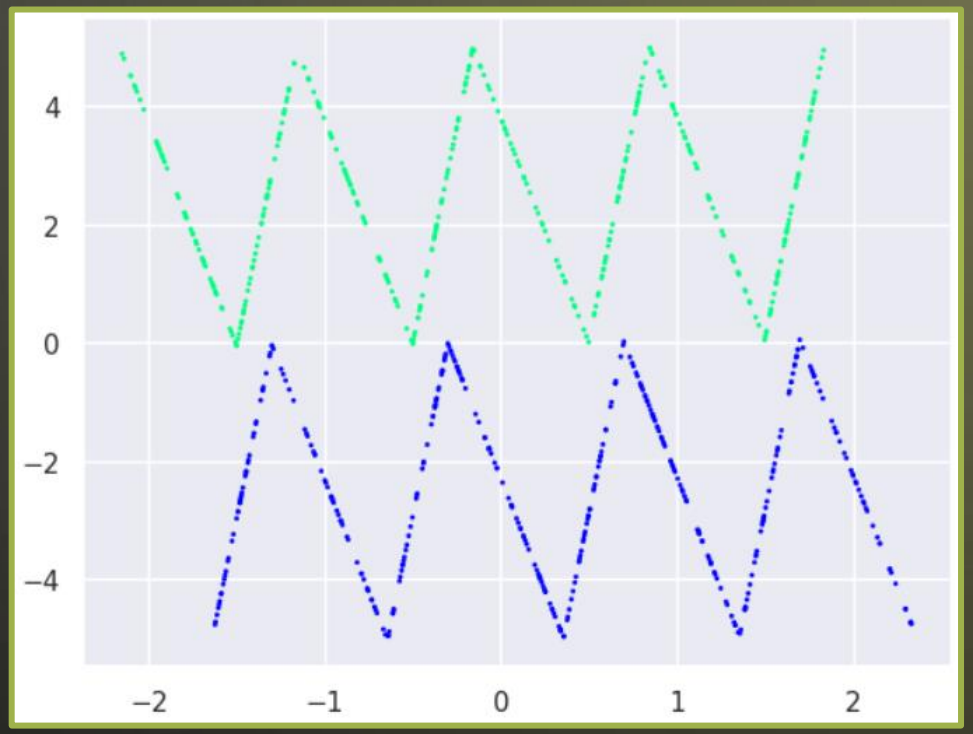
Solution using nonsmooth penalty function and ellipsoid method

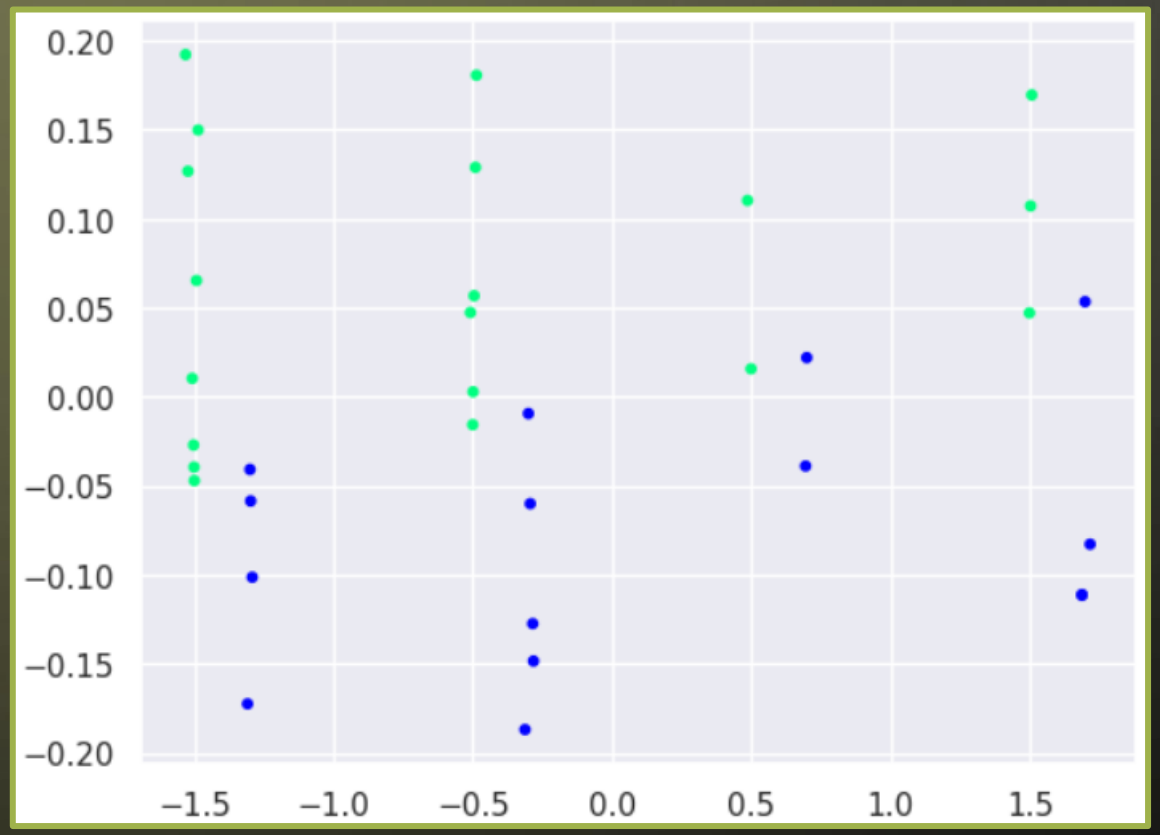
"SAW" DATASET WITH GAP $\varepsilon = 10^{-3}$



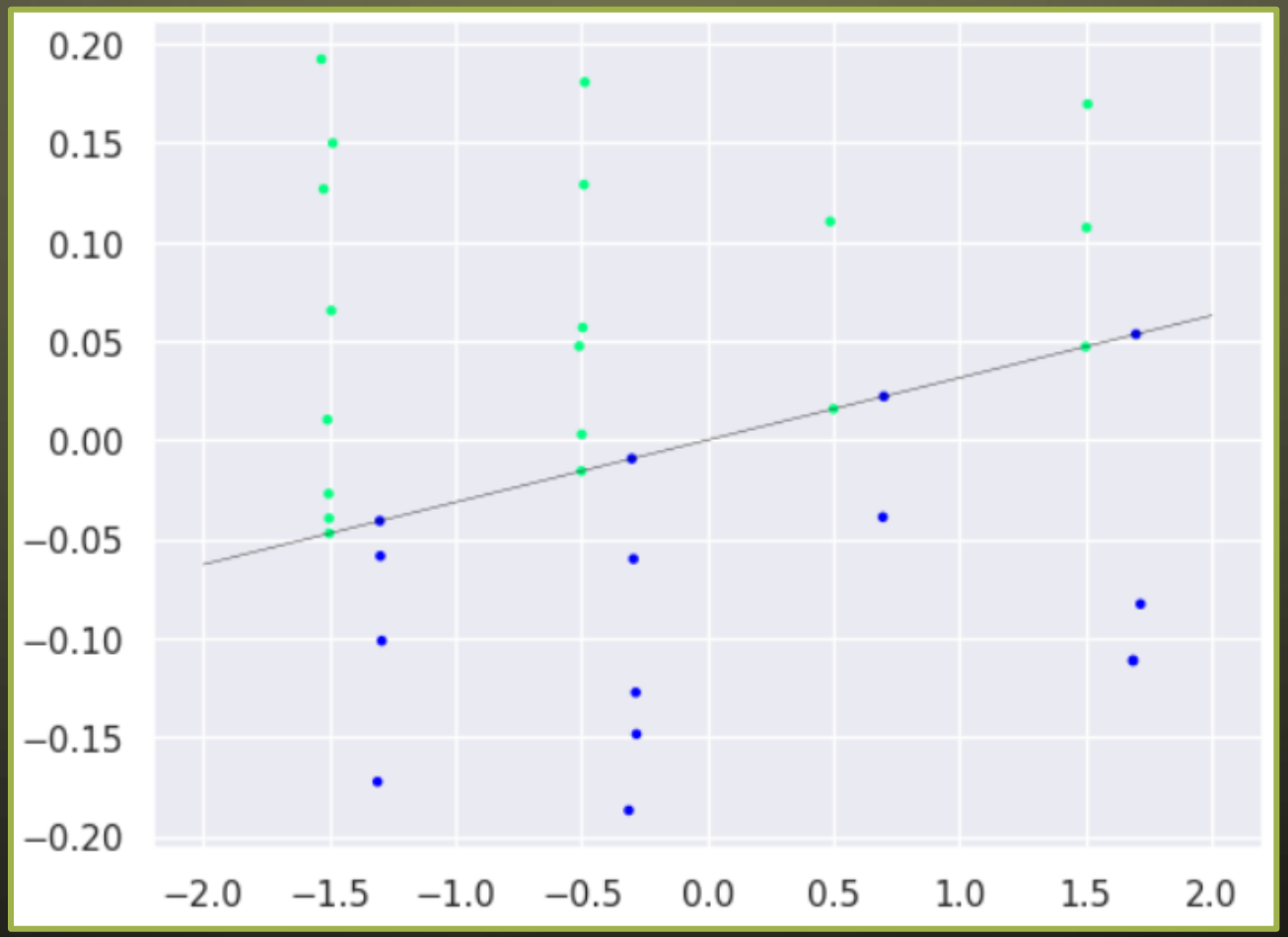
Solution using sklearn.svm.SVC(C=1e9, kernel='linear')

"SAW" DATASET WITH GAP $\varepsilon = 10^{-6}$



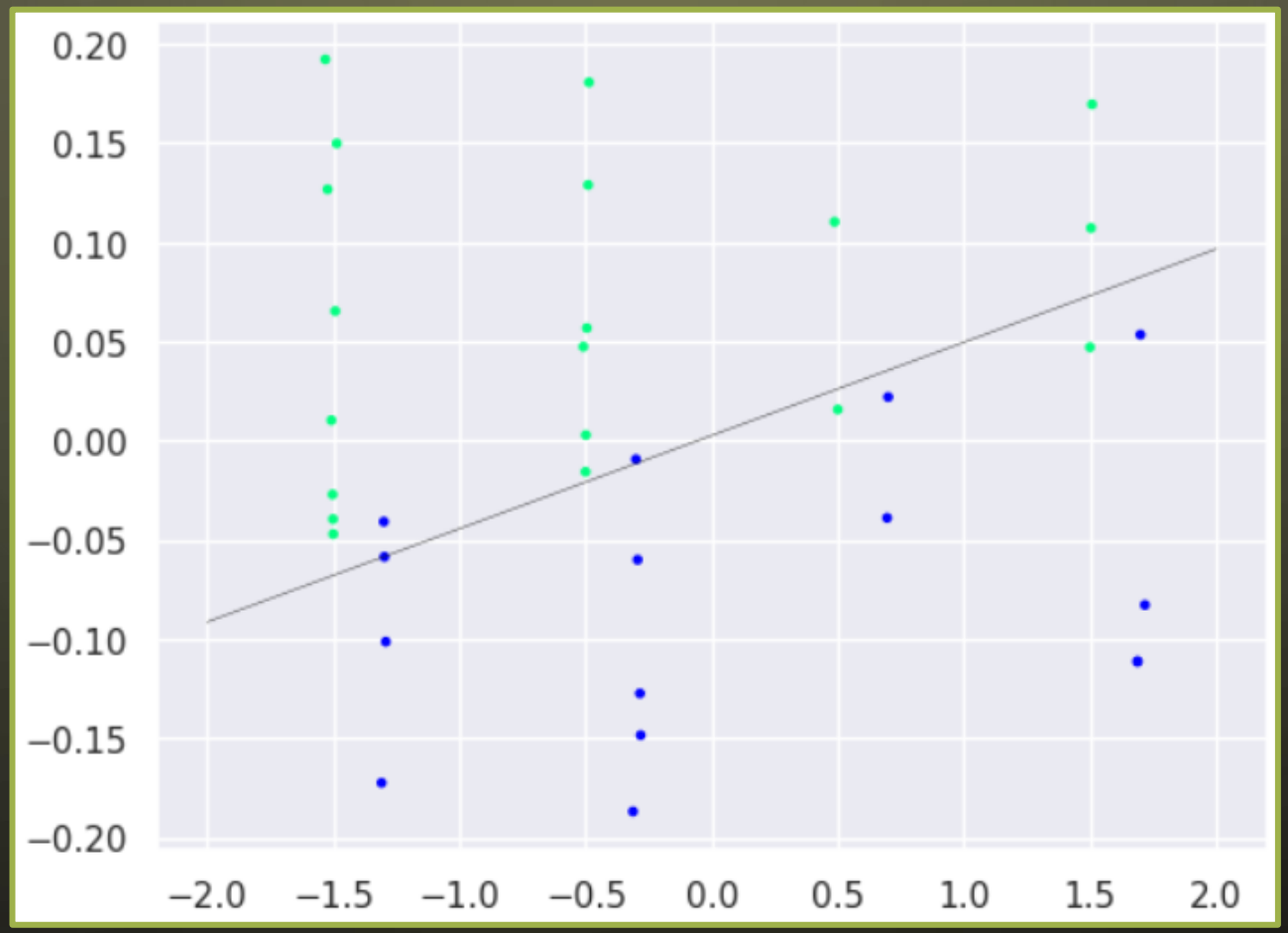


"SAW" DATASET WITH GAP $\varepsilon = 10^{-6}$



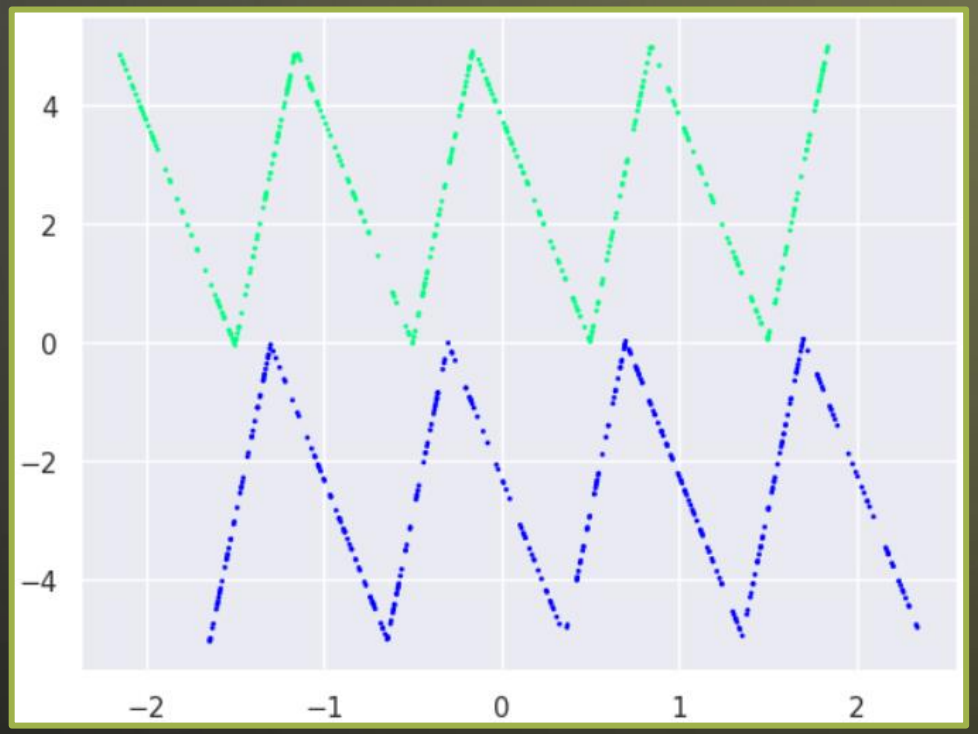
Solution using nonsmooth penalty function and ellipsoid method

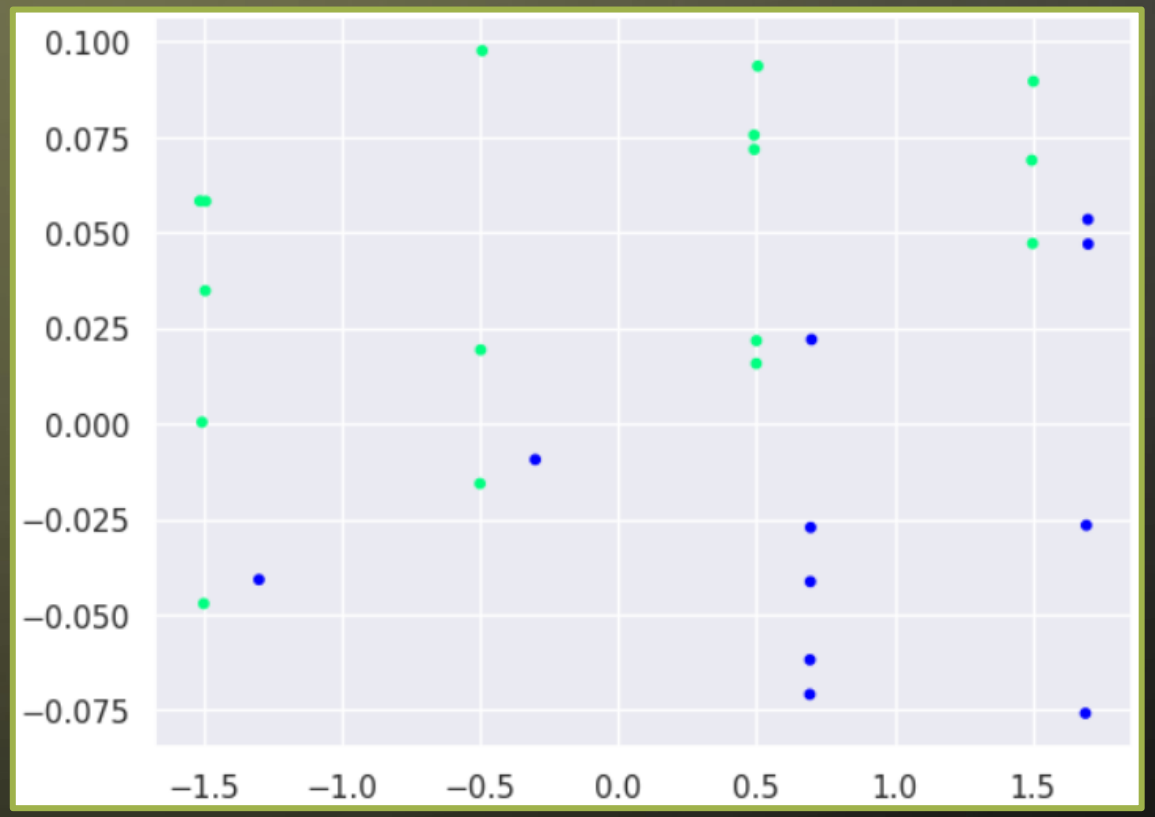
"SAW" DATASET WITH GAP $\varepsilon = 10^{-6}$



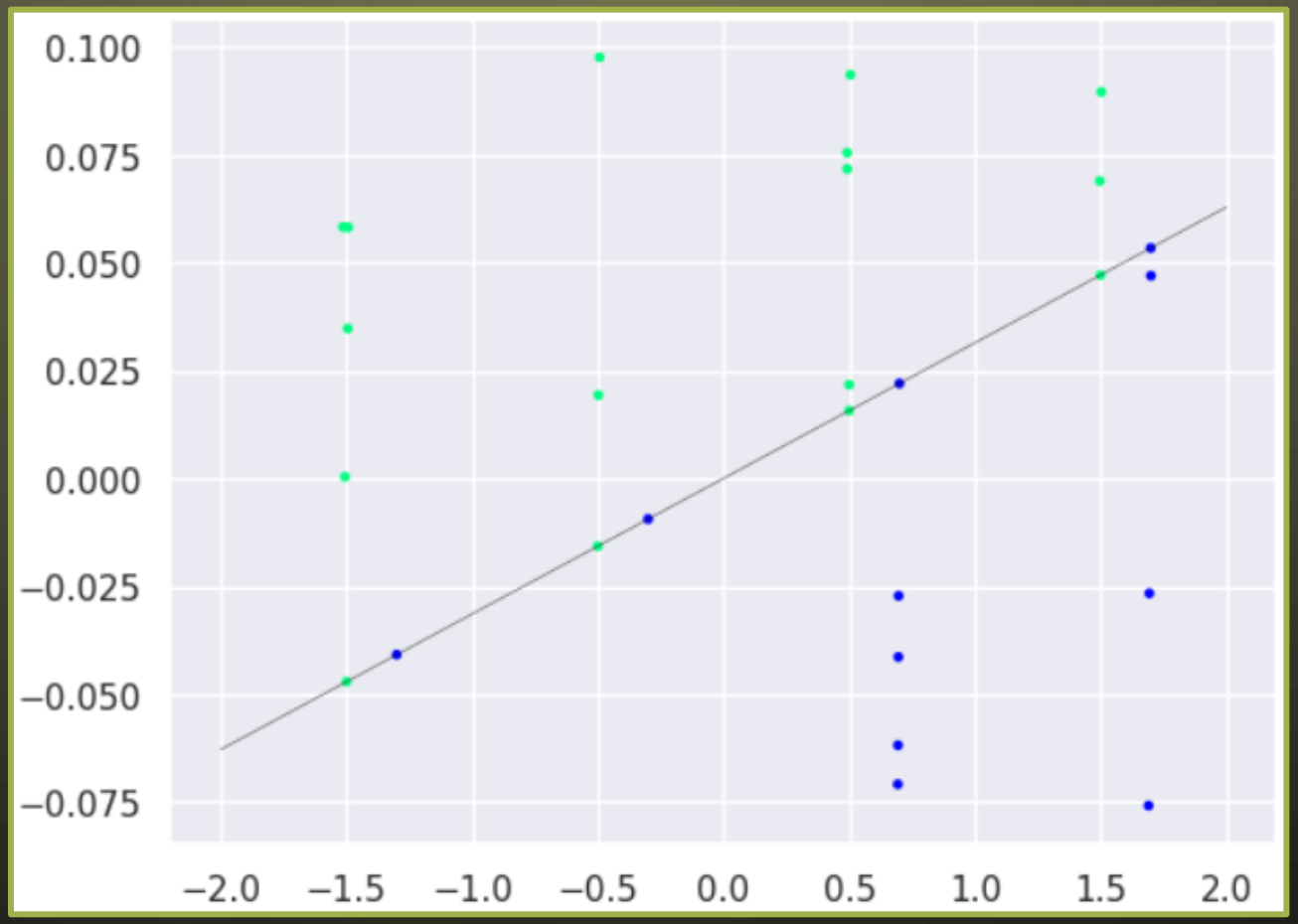
Solution using sklearn.svm.SVC(C=1e9, kernel='linear')

“SAW” DATASET WITH GAP $\varepsilon = 10^{-12}$



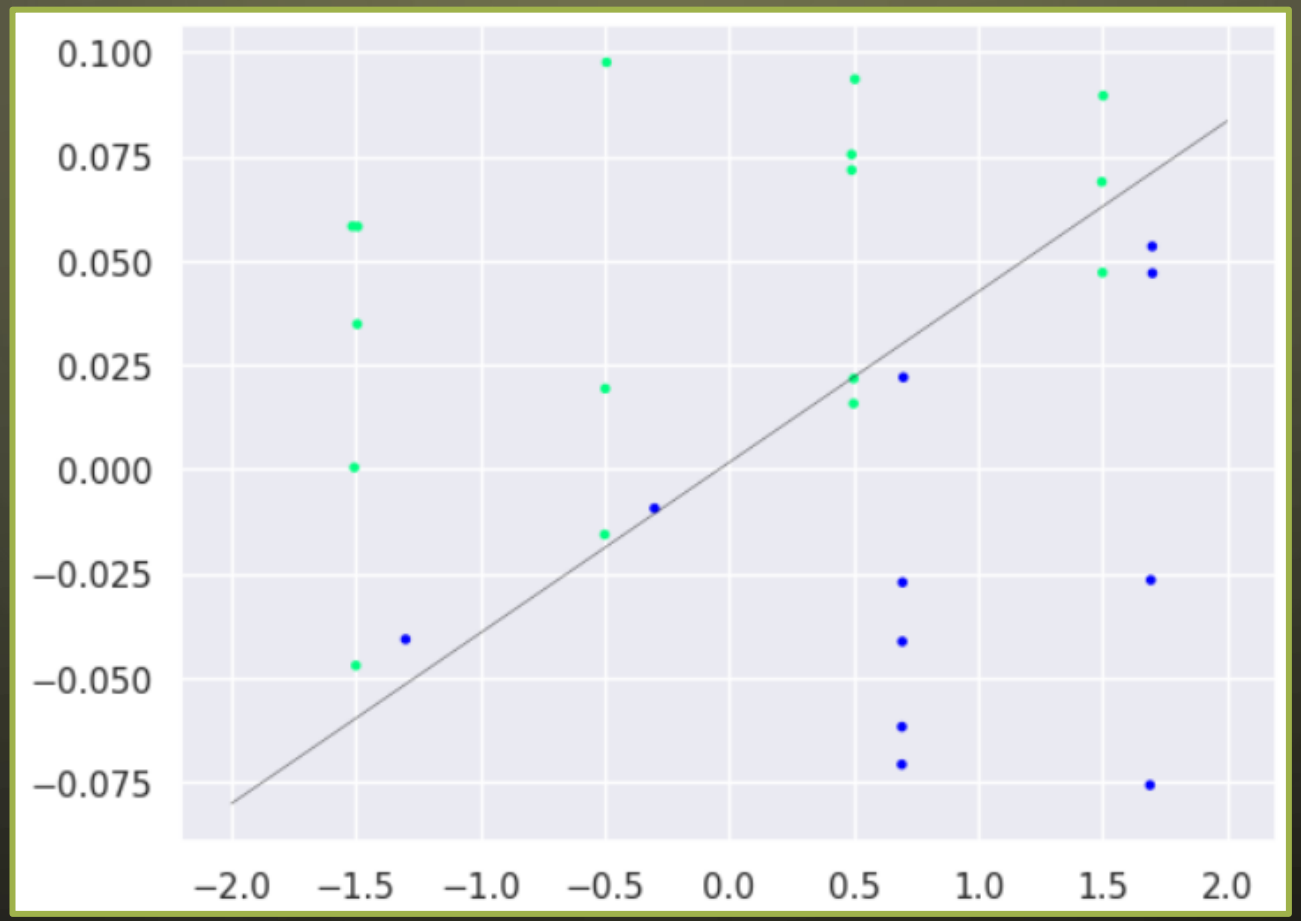


"SAW" DATASET WITH GAP $\varepsilon = 10^{-12}$



Solution using nonsmooth penalty function and ellipsoid method

"SAW" DATASET WITH GAP $\varepsilon = 10^{-12}$



Solution using sklearn.svm.SVC(C=1e9, kernel='linear')

CONCLUDING REMARKS

April 15, 2005

Dear Professor Shor,

We have never met, but your work has very much influenced me for many years now. I started with your small 1985 Springer book on subgradient methods, which I read as a PhD student. I recently read your newer book on nondifferentiable optimization (1998), which I enjoyed very much.

I'm enclosing copies of the three books I've written. The first concerns the design of linear controllers via convex optimization; the second is on linear matrix inequalities; and the third one is a basic textbook on convex optimization. [. . .] I hope you can see your strong influence in all of these books.

With the best regards,
Stephen P. Boyd

"Theory of the entire class of algorithms with space dilation is still far from being perfect. It seems to us that it is rather realistic goal to create such an algorithm, that would not be inferior to r-algorithm in terms of its practical effectiveness, and also would be as well justified as an ellipsoid method."

N.Z. Shor

REFERENCES

1. N.Z. Shor Minimization Methods for Non-Differentiable Functions, 1985, 164 p. (translated monograph)
2. N.Z. Shor Application of the gradient descent method for solving the network transport problem, Materials of a scientific seminar on theoretical and applied issues of cybernetics and operations research 1, pp. 9-17 (1962) (in russian)
3. N.Z. Shor Cut-off method with space extension in convex programming problems, Cybernetics 13, pp. 94-96 (1977)
4. D.B. Yudin, A.S. Nemirovskii, Informational complexity and efficient methods for the solution of convex extremal problems, Matekon 13, pp. 25-45 (1976)
5. L.G. Khachiyan, Polynomial algorithms in linear programming, USSR Comput. Math. Math. Phys. 20 (1), pp. 53-72 (1980)
6. N.Z. Shor, M.G. Zhurbenko, A minimization method using the operation of extension of the space in the direction of the difference of two successive gradients, Cybernetics 7 (3), pp. 450-459 (1971)

REFERENCES

7. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Springer Series in Statistics): textbook. Springer, 2nd Edition. 2016. 767 p.
8. P.I. Stetsyuk, V.V. Savitsky, On Defects Searching in Regular 3D-Structures, Journal of Automation and Information Sciences 50 (3), pp. 21-37 (2018)
9. Deisenroth M., Faisal A., Soon Ong C. Mathematics for Machine Learning: textbook. Cambridge, 1st Edition. 2020. 398 p.
10. Boyd S., Vandenberghe L., Convex Optimization: textbook, Cambridge University Press, 2004 https://web.stanford.edu/~boyd/cvxbook/
11. Stetsyuk P.I., Berezovskyi O.A., Zhurbenko M.G., Kropotov D.O. Non-smooth optimization methods in special classification problems, preprint, V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, 2009-1, 28 p. (in Ukrainian)
12. Pshenychnyy B.M., Linearization method, Nauka, 1983, 136 p. (in russian)

ACKNOWLEDGEMENTS

This report is supported by:

- DTT TS KNU NASU grant №2M-2024
- Volkswagen Foundation grant № 97775
- The project of research works of young scientists №07-02/03-2023

THANK YOU FOR YOUR ATTENTION!