Contents

We consider some new approach to method convex 2-dimensional optimization on a fixed square recently proposed by Yu. E. Nesterov (see ex.1.5 in https://arxiv.org/pdf/1711.00394.pdf). The method can be interested for to solve a dual problem for a convex problem with two functional constraints. The idea of the method consists in the narrowing of the domain until we achieve an acceptable quality of the solution. This method has to search minimum on the separating segments and we propose some non-depended on required accuracy stop condition for it. Estimations for the iterations number are proved for the cases of smooth and non-smooth functions. Besides there is an experimental comparison of our method with other inexact methods of convex optimization such as ellipsoid method with epsilon-subgradient and primal and fast gradient method with delta-L-oracle. The experiments were made on dual problems for problems with two constraints As a result, our method with provided in this work strategy has the best result.

1 Introduction

In this paper we research one method of optimization on a square in $\mathbb{R}^2$. The method was offered by Nesterov (see ex. 4 from [1]). interested for to solve a dual problem for a convex problem with two functional constraints. This method is reffered as Halving Square Method.

The idea of the method consists in the narrowing of the domain until we achieve an acceptable quality of the solution. This method has to search minimum on the separating segments. In the next section there is description of method and pseudocode for it.

In the paper [2] there are some results for this method. Namely, there is an estimate for iterations number. Additionally, the authors provided strategy for subproblem on segment. Also there are comparison of this method with method of ellipsoids. Discussed method showed better results on time.

But the provided strategy has several disadvantages. Firstly, the required accuracy for the problem on segment significantly depends on the required accuracy for initial task on segment. It is essential disadvantage when one want to solve initial task extra accurately. Secondly, the method has convergence only on function and is able to not converge to solution on argument.

In this paper we continue this research and provide a new strategy for separating segments. According to it there is stop condition for one-dimensional problem. This stop condition does not depend on required initial accuracy

and the HSM with this strategy converges to the solution on argument. The discussion of this strategy is in the section 4.

Moreover, the new estimates for iterations number was provided in this work. There are estimates for smooth and non-smooth problem when on each iteration one chooses rectangle that include the global solution. Additionally, there are theoretical and experimental comparisons of this estimates.

Also we carried out some numerical expirements for to compare the Halving Square Method with other methods. Firstly, it is comparison of different strategies for the Halving Square Method on problems when we can calculate object function's value analitacally. Secondly, it is comparison on dual problem when we can not do it. In this case we compare our method with other such inexact methods as primal gradient method, fast gradient method and ellipsoid method.

This experiments showed that the Halving Square Method with provided in this work strategy is optimal method for to solve dual problem with high accuracy. And the gain of using this method grows with the growth of required accuracy.

2 Method Description

Let's consider a following task:

$$\min_{(x,y)} \{f(x,y)|(x,y) \in Q\},$$

where f is a convex function, Q - is a square on the plane.

Let's consider a following method. One solves task of minimization for a function $g(x) = f\left(x, y_0 = \frac{a}{2}\right)$ on a segment $[0, a]$ with an accuracy δ on function. After that one calculates a subgradient in a received point and chooses the rectangle which the subgradient "does not look" in. Similar actions are repeated for a vertical segment. As a result we have the square decreased twice. Let's find a possible value of error δ_0 for task on segment and a sufficient iteration's number N to solve the initial task with accuracy ϵ on function.

Let's describe an algorithm formally. See pseudo-code 1.

Algorithm 1 Halving Square Method

function METHOD(convex function f, square $Q = [a, b] \times [c, d]$)
  $x_* \leftarrow \arg\min_{x \in [a,b]} f(x, \frac{c+d}{2})$
  $g \leftarrow \partial f(x_*, \frac{c+d}{2})$
  if g[1] > 0 **then**
    $Q := [a, b] \times [c, \frac{c+d}{2}]$
  else
    $Q := [a, b] \times [\frac{c+d}{2}, d]$
  end if
  $y_* \leftarrow \arg\min_{y \in [c,d]} f(\frac{a+b}{2}), y)$
  $g \leftarrow \partial f(\frac{a+b}{2}), y_*)$
  if g[0] > 0 **then**
    $Q := [a, \frac{a+b}{2}] \times [c, d]$
  else
    $Q := [\frac{a+b}{2}, b] \times [c, d]$
  end if
  if $f(\frac{a+b}{2}, \frac{c+d}{2}) - f^* > \epsilon$ **then**
    Method(f, Q)
  end if
  return $(\frac{a+b}{2}, \frac{c+d}{2})$
end function

The arg min will be find through dichotomy method that uses derivative, i.e. it is method that take the derivative in the center of curent segment and select the part segment that anti-gradient looks at. The interesting remark is that it is one-dimensional analog of our method.

The sufficient accuracy for arg min will be discussed in the section 4.

3 Algorithm correctness

In this section there are profes for the fact that our method works for some accuracy for one-dimensional problem on segment.

Now we introduce the following notation. If we solve the one-dimensional problem then $f'_\perp$ is a perpendicular to the segment subgradient's component, i.e. it is a derivative on the variable that is fixed on this segment. In this case $f'_\parallel$ is a parallel to the segment subgradient's component.

We will use the following enough obvious lemma.

Lemma 3.1. *If $\boldsymbol{x}_*$ is a solution of one-dimensional problem on a horizontal segment then*

$$\exists g \in \partial f(\boldsymbol{x}_*) : g_{\|} = 0$$

Proof. ... □

Theorem 3.1. *Let's the f has continuous derivative on the square. Then there is a neighbourhood of a solution of optimization task on segment such as a choice of rectangle will not change if one use any point from the neighbourhood.*

Proof. ... □

The method does not work for all convex functions even for zero error on segment. There is an example of non-smooth convex problem in [2] when this method can not converge to the solution more accurately then one constant.

4 One-Dimensional Problem

In this section we describe a sufficient conditions for to stop to solve one-dimensinal problem on segment. We will use the following notation:

$$(x_*, y_*) \text{ – solution of one-dimensional problem}$$

$$\delta = |x - x_*| \text{ – distance}$$

In the work [2] there are proves for the following theorem:

Theorem 4.1. *Let f be convex M-Lipschitz continious function with L-Lipschitz continious gradient. Then if each one-dimensional task was solved with the following accuracy*

$$\delta \leq \frac{\epsilon}{2Ma(\sqrt{2}+\sqrt{5})(1-\frac{\epsilon}{La\sqrt{2}})} \tag{1}$$

then this method converge to minimum of f on square with accuracy ϵ on function.

This strategy requires to solve each one-dimensional problem with accuracy of order of ϵ. It can be sufficient disadvantage for iterations on the big segments in the start of method's work. Also this strategy does not give convergence on argument. We will call this strategy **ConstEst**(Constant Estimate).

Let's develop strategy where there is convergence on argument. Rectangles are defined correctly for a horizontal optimization task, if:

$$f'_y(\mathbf{x}_*)f'_y(x_* + \delta, y_*) > 0 \tag{2}$$

Analogically, for a vertical segment:

$$f'_x(\mathbf{x}_*)f'_x(x_*, y_* + \delta) > 0 \tag{3}$$

Theorem 4.2. *Let function f be convex and has L-Lipschitz continuous gradient and a point $\boldsymbol{x}_*$ is a solution of optimization's task on a current segment.*

The current segment is horizontal and $M = |f'_y(\boldsymbol{x}_{current})|$ or the current segment is vertical and $M = |f'_y(\boldsymbol{x}_{current})|$. Then rectangle is defined correctly if a distance between $\boldsymbol{x}_{cur}$ and accurate solution on segment is less than $\frac{M}{L}$.

Proof. Condition (2) is met if there is a derivative $f'_y(x_0 + \delta, y_0)$ in a neighbourhood of $f'_y(\mathbf{x}_*)$ with radius $\left|f'_y(\mathbf{x}_{cur})\right|$:

$$\left|f'_y(\mathbf{x}_*) - f'_y(\mathbf{x}_{cur})\right| < \left|f'_y(\mathbf{x}_{cur})\right|$$

The L-Lipschitz continuity gives following inequality:

$$\left|f'_y(\mathbf{x}_*) - f'_y(\mathbf{x}_{cur})\right| \leq L|\delta|$$

Therefore the following possible value is sufficient to select rectangle correctly:

$$\delta_0 < \frac{M}{L} \leq \frac{\left|f'_y(\mathbf{x})\right|}{L}$$

Statement for vertical segment is proved similarly. □

Theorem 4.2 gives stop conditions in the case when gradient in point-solution on segment and close points is large. But what should we do if gradient in this point is small?

Theorem 4.3. *Let f be M-Lipschitz continuous convex and have L-Lipschitz continuous gradient. The points $\boldsymbol{x}_*$ is one-dimensional problem's solution and $\boldsymbol{x}$ is its approximation, $\delta = \|\boldsymbol{x}_* - \boldsymbol{x}\|$ is a distance between them.*

Then for accuracy on function ϵ following condition in point $\boldsymbol{x}$ is sufficient:

$$\delta \leq \frac{\epsilon - LR|f'_\perp(x_*)|}{L + MR},$$

where $R = a\sqrt{2}$ is size of current square.

Proof. From the lemma 3.1 we have:

$$g \in \partial f(\mathbf{x}_*) : g_\| = 0.$$

Then the following inequallity is true by diffenition of subgradient:

$$f(\mathbf{x}^*) - f(\mathbf{x}_*) \geq (g, \mathbf{x}^* - \mathbf{x}_*)$$

Using Cauchy–Bunyakovsky–Schwarz inequality one has following inequality

$$f(\mathbf{x}_*) - f(\mathbf{x}^*) \leq -(g, \mathbf{x}^* - \mathbf{x}_*) \leq$$
$$\leq \|g\|\|\mathbf{x}^* - \mathbf{x}_*\| \leq \|g\|a\sqrt{2}$$

On the other hand, we have:

$$f(\mathbf{x}) - f(\mathbf{x}_*) \leq M\delta$$

Therefore we have the following estimate:

$$f(\mathbf{x}) - f(\mathbf{x}^*) \leq M\delta + \|g\|a\sqrt{2} = M\delta + |f'_\perp(\mathbf{x}_*)|R$$

From Lipschitz continious from gradient we have:

$$f(\mathbf{x}) - f(\mathbf{x}^*) \leq M\delta + (|f'_\perp(\mathbf{x})| + L\delta)\, R$$

Then for to approach accuracy ϵ at the point x the following condition is sufficient:

$$M\delta + \|g\|a\sqrt{2} = M\delta + (|f'_\perp(\mathbf{x})| + L\delta)\, R \leq \epsilon$$

$$\delta \leq \frac{\epsilon - |f'_\perp(\mathbf{x})|}{M + LR}$$

□

Then our addaptive strategy is following. One is to calculate untill the following condition is met:

$$\delta \leq \max\left\{\frac{|f'_{\perp}(\mathbf{x})|}{L}, \frac{\epsilon - |f'_{\perp}(\mathbf{x})|}{M + LR}\right\}. \tag{4}$$

And if the condition from theorem 4.3 we stop our method. This strategy is reffered as **CurGrad**(Current Gradient).

Let's make a couple of remarks.

Firstly, we can replace the Lipschitz condition on the square by the Lipschitz condition on the segments in the written above theorems.

Secondly, we can use in this theorems Lipschitz constants not for gradients but for partitial derivative on segments. Its proves are obvious enough and almost repeat proves for this theorem.

5 Convergence

Below proved estimates are correct if each iterations was correct, i.e. after each iteration there is the solution in the selected rectangle. It is important condition

Theorem 5.1. *If function f is convex and L_f-Lipschitz continuous, then for to solve initial task with accuracy ϵ on function one should make following iteration's numbers:*

$$N = \left\lceil \log_2 \frac{\sqrt{2} L_f a}{\epsilon} \right\rceil \tag{5}$$

where a is a size of the initial square Q.

Proof. The square's size equals $\frac{a}{2^k}$ on k'th iteration. If each irerations was correct then

$$\|\mathbf{x}_k - \mathbf{x}^*\| \leq \frac{a}{2^k}\sqrt{2}$$

Using the fact that function is Lipschitz continuous we have estimate 5 □

The similar estimate was proven for the fist strategy without convergence on argument too. This proof can be found in the work [2].

There are functions which estimates from written above theorem are very accurate for.

Example 5. Let's consider following task with positive constant α:

$$\min\left\{\alpha(x+y)|Q=[0,1]^2\right\}$$

If one take a center of a current solution as approximate solution one have value $\frac{\alpha}{2^N}$ after N iterations. Therefore, for accuracy ϵ one has to $\lceil\log_2\frac{\alpha}{\epsilon}\rceil$. For this function $L_f=2\alpha$. Therefore, estimate (5) is accurate for such tasks with little error that not more one iteration.

We can improve written above estimates if to add new conditions:

Theorem 5.2. *Let function f be convex.*

If

1. Function f has L-Lipschitz continious gradient

2. $\exists \boldsymbol{x}^*\in Q:\nabla f(\boldsymbol{x}^*)=\boldsymbol{0}$

3. Strategy gives a convergence on argument

then for to approach accuracy ϵ on function the following iterations number is sufficient:

$$N=\left\lceil\frac{1}{2}\log_2\frac{Ma^2}{4\epsilon}\right\rceil, \tag{6}$$

where a is a size of the initial square Q.

Proof. For all convex functions there is following inequality (one may find proof in [3]):

$$f(\mathbf{x})-f(\mathbf{x}^*)-(f'(\mathbf{x}^*),\mathbf{x}-\mathbf{x}^*)\leq\frac{L}{2}\|\mathbf{x}-\mathbf{x}^*\|^2$$

If $\mathbf{x}^*$ is a solution and an internal point, then $f'(\mathbf{x}^*)=0$:

$$f(\mathbf{x})-f(\mathbf{x}^*)\leq\frac{L}{2}\|\mathbf{x}-\mathbf{x}^*\|^2$$

After N iterations we have the estimate:

$$f(\mathbf{x})-f(\mathbf{x}^*)\leq L\left(\frac{a}{2^N}\right)^2$$

Using it we have estimate (6). □

Estimate 5 works in all cases and better then estimate 6 when following condition is met:

$$\frac{2L_f^2}{L_g} \leq \epsilon,$$

where L_f, L_g is Lipschitz constants for function and for gradient. On the other hand, the estimate 6 works if point with zero gradient exists in the square.

6 Halving Cube Method

This Halving Square Method is optimization method for convex problem on square. But there is obvious generalization for convex problems in high dimensional spaces on n-dimensional hypercube. Let's describe it:

1. Take $(n-1)$-dimensional hyperplane in hypercube that includes hypercube center and is parallel to one of its face.

2. To solve optimization problem on this set with accaracy that is met to condition from theorem 4.2.

3. Calculate gradient in this point and select rectangle which antigradient look into.

4. Repeat the third step n times for each hyperplane in hypercube that includes hypercube center and is parallel to one of its face.

5. Repeat the previous step until required accuracy is achieved.

The proves for convergence for smooth function and for strategy repeat proves in the case of 2-dimensional space. Moreover, obvious enough that the following theorems are correct when on the fourth algorithm step rectangle was selected correctly.

Theorem 6.1. *If function f is convex and L_f-Lipschitz continuous, then for to solve initial task with accuracy ϵ on function one should make following iteration's numbers:*

$$N = \left\lceil \log_2 \frac{\sqrt{n} L_f a}{\epsilon} \right\rceil \tag{7}$$

where a is a size of the initial square Q.

Theorem 6.2. *Let function f be convex.*

If

1. Function f has L-Lipschitz continious gradient
2. $\exists \boldsymbol{x}^* \in Q : \nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$
3. Strategy gives a convergence on argument
then for to approach accuracy ϵ on function the following iterations number is sufficient:

$$N = \left\lceil \frac{1}{2} \log_2 \frac{Ma^2}{4\epsilon} \right\rceil, \tag{8}$$

where a is a size of the initial square Q.

Note the hyperplane in hypercube in the first step is $(n-1)$-dimensional hypercube. And we can use for it this method too. So, we solve the convex problem on n-dimensional hypercube recursively.

But this method has two essential disadvanteges for enough high dimensional spaces. Fisrtly, the required iterations number depend on logarithm of dimension. Of course, it is essential only for very high dimensional problems. Secondly, recursive algorithm will be extra slow because each the complexity of each iteration for this method is factotial of dimension.

We don't expect inspired result for each modification and we will not test it.

7 The Dual Problems

This method is interesting for to solve dual problems for problems with two constraints. Namely, we are interesting in solution of the following problem:

$$\phi(\lambda_1, \lambda_2) \to \min_{\lambda \geq 0}, \tag{9}$$

$$\text{where } \phi = -\min_{\mathbf{x}} \left(f(\mathbf{x}) + \lambda_1 g_1(\mathbf{x}) + \lambda_2 g_2(\mathbf{x}) \right) \tag{10}$$

$$\mathbf{x}(\lambda) = \arg\min_{\mathbf{x}} \Phi(\mathbf{x}, \lambda)$$

In this section we will discuss how transform this task to task of the task of minimization on square, what is derivative and lipschitz constants. Also, there is description of way for to calculate the value of function ϕ and its derivative.

Fitstly, let's transform this problem to the problem of minimization on square. According to [1] (see ex. 4.1), we can add following restraint for the dual variables:

$$\|\lambda\|_1 \leq a = \frac{1}{\gamma}\left(f(\overline{\mathbf{x}}) - \min_{\mathbf{x}} f(\mathbf{x})\right), \tag{11}$$

$$\text{where } \overline{\mathbf{x}} : g_i(\overline{\mathbf{x}}) < 0, \gamma = \min_i[-g_i(\overline{\mathbf{x}})] \tag{12}$$

According to this statemant ther is the λ^* in square $Q = [0, a]^2$. And we have following optimization task:

$$\phi(\lambda_1, \lambda_2) \to \min_{\lambda \in Q} \tag{13}$$

The next point is a gradient of function ϕ. It will be calculated according to well-known Demyanov-Danskin-Rubinov Theorem, see [5].

Theorem 7.1. *Let $\phi(\lambda) = \min\limits_{x \in X} \Phi(x, \lambda)$ for all $\lambda \geq 0$, where Φ is a smooth convex function with respect to λ and $x(\lambda)$ is the only maximum point. Then*

$$\nabla\phi(\lambda) = F'_\lambda\left(x(\lambda), \lambda\right)$$

If theorem's conditions is met for our case then we have derivative value:

$$\phi'_{\lambda_k}(\lambda) = g_k\left(\mathbf{x}(\lambda)\right) \tag{14}$$

Additionally, we need a Lipschitz constant for gradient. In the work [6] there is following theorem:

Theorem 7.2. *Let $f(x)$ be a μ_f-strongly convex function, the function $g(x)$ satisfies the Lipschitz condition with a constant M_g. Then the function $\phi(\lambda) = \min_{\boldsymbol{x}}\left(f(\boldsymbol{x} + \lambda_1 g_1(\boldsymbol{x}) + \lambda_2 g_2(\boldsymbol{x})\right)$ defined in 20, where $x(\lambda) = \arg\min\limits_x(f(x) + \lambda g(x))$, has Lipschitz smooth gradient with constant $L_{\phi'} = \frac{M_g^2}{\mu_f}$*

This theorem can be proved easy for the 2-dimensional space. In this case g is a vector-function.

The next subsections are devoted to efficient calculation of function's and derivative's values.

The main problem for such saddle-points problems is that one can not calculate $\mathbf{x}(\lambda)$ presicely. Therefore, one have not precise value of the gradient in our method.

We need $\mathbf{x}(\lambda)$ when we solve one-dimensional problem. There are three following cases:

1. The step of dichotomy on segment.
2. The test of stop-condition.
3. The select of rectangle.

Let's assume that we solve one dimensional problem on segment that is paralell to the first axis. For the case of the second axis the results will be similar. In the each above written case we are interesting in not full gradient but :

1. The first component, i.e. $g_1(\mathbf{x}(\lambda))$.
2. (for the strategy through current gradient) the value of difference $\delta - \frac{|g_2(\mathbf{x}(\lambda))|}{L}$, where L is a Lipschitz constant for gradient of ϕ.
3. The second component, i.e. $g_2(\mathbf{x}(\lambda))$.

According to [2] we can calculate derivatives inexactly for to select rectangle with ϵ-solution. So if δ is accuracy on argument of one-dimensional problem solution and Δ is accuracy of calculating gradient at point then if the following condition is met:

$$2\Delta + L\delta \leq \frac{\epsilon}{2a(\sqrt{2}+\sqrt{5})},$$

where L is a Lipshitz constant of gradient for dual problem then the rectangle with ϵ-solution will be selected. We can see that the task of selecting segment with one-dimensional problem's solution is similar to the task of selecting rectangle with global ϵ-solution where $\delta = 0$. Then we have the following theorem:

Theorem 7.3. *If one-dimensional problem is solved with accuracy delta* $= \frac{\epsilon}{4La(\sqrt{2}+\sqrt{5})}$ *on argument then for to select rectangle correctly one needs to calculate* $\boldsymbol{x}(\lambda)$ *at this point with the following accuracy:*

$$\|\boldsymbol{x} - \boldsymbol{x}(\lambda)\| \leq \frac{1}{M_g}\frac{\epsilon}{8a(\sqrt{2}+\sqrt{5})} \tag{15}$$

For to select segment with one-dimensional problem's solution one needs to calculate $\boldsymbol{x}(\lambda)$ *at the center of current segment with the following accuracy:*

$$\|\boldsymbol{x} - \boldsymbol{x}(\lambda)\| \leq \frac{1}{M_g}\frac{\epsilon}{4a(\sqrt{2}+\sqrt{5})} \tag{16}$$

On the other hand, for each case only sign of corresponding expression. For it we will use the following lemma.

Lemma 7.1. $\forall a, b \in \mathbb{R} : b \neq 0, |a - b| \leq |b| \Rightarrow \operatorname{sign} a = \operatorname{sign} b$

Proof. If $b > 0$ and $a \leq b$ then the lemma's condition is equivalent to the following condition:

$$b - a \leq b \Rightarrow a \geq 0.$$

If $b > 0$ and $a \geq b$ then $a \geq 0$.

The case of negative b is proven similary. □

According this lemma we have the following stop conditions for calculating $\mathbf{x}(\lambda)$ for the cases:

1. $|g_1(\mathbf{x}) - g_1(\mathbf{x})| \leq |g_1(\mathbf{x})|$
2. $\frac{1}{L}\Big||g_2(\mathbf{x})| - |g_2(\mathbf{x}(\lambda))|\Big| \leq \Big|\delta - \frac{|g_2(\mathbf{x})|}{L}\Big|$
3. $|g_2(\mathbf{x}) - g_2(\mathbf{x}(\lambda))| \leq |g_2(\mathbf{x})|$

In this moment we can state the following theorem.

Theorem 7.4. *Let g_k be L_{g_k}-Lipschitz continious. Then the following statements are true:*

1. For to make the dichotomy step correctly one can calculate $\boldsymbol{x}(\lambda)$ untill the following condition is approached:

$$L_{g_1}\|\boldsymbol{x} - \boldsymbol{x}(\lambda)\| \leq |g_1(\boldsymbol{x})|.$$

2. For to test stop condition correctly one can calculate $\boldsymbol{x}(\lambda)$ untill the following condition is approached:

$$\frac{L_{g_2}}{L}\|\boldsymbol{x} - \boldsymbol{x}(\lambda)\| \leq \Big|\delta - \frac{|g_2(\boldsymbol{x})|}{L}\Big|,$$

where δ is a distance between λ and solution of one dimensional task λ_*.

3. For to select rectangle according to strategy one can calculate $\boldsymbol{x}(\lambda)$ untill the following condition is approached:

$$L_{g_2}\|\boldsymbol{x} - \boldsymbol{x}(\lambda)\| \leq |g_2(\boldsymbol{x})|.$$

There are two interesting remarks.

Firstly, the calculating of $\mathbf{x}(\lambda)$ does not depend on required accuracy ϵ for function ϕ. This result is unique for our method and one can not see such effect in other inexact methods (see the next section).

Secondly, this strategies from written above theorem does not guarantee that the calculating of $\mathbf{x}(\lambda)$ will stop. For example, let's consider the first

condition. If $g_1(\mathbf{x}(\lambda)) = 0$ and $|g_1(\mathbf{x})|$ decreases faster than $L_{g_1}\|\mathbf{x} - \mathbf{x}(\lambda)\|$ with decreasing of $\|\mathbf{x}-\mathbf{x}(\lambda)\|$ than the stop condition will not be approached. This case is looked as extra specific and does not observed in our experiments but it can take place.

8 Complexity

Let's estimate complexity of one-dimensional problem.

For the strategy **ConstEst** we have that each one-dimensional problem needs exactly the following iterations of one-dimensional method:

$$\log_2 \frac{2La(\sqrt{2}+\sqrt{5})(1-\frac{\epsilon}{Ma\sqrt{2}})}{\epsilon} = O\left(\log\frac{1}{\epsilon}\right)$$

Now let's condider the strategy **CurGrad**. Let the modul of the perpendicular component at point-solution $|f'_\perp(\mathbf{x}_*)|$ is equal to $\tilde{\epsilon}$. A point from segment $\mathbf{x}$ is its approximation and δ is a distance between them.

If we use the dichotomy method and N is a number of current dichotomy iteration we have from L-Lipschitz continious gradient the following estimates for derivative at $\mathbf{x}$:

$$\tilde{\epsilon} - La2^{-N} \leq |f'_\perp(\mathbf{x})| \leq \tilde{\epsilon} + La2^{-N}$$

Accordint to this estimates for to approach a the first alternative from adaptive estimate 4 the following condition is sufficient:

$$a2^{-N} \leq \frac{\tilde{\epsilon} - La2^{-N}}{L}$$

Similarly for the second alternative:

$$a2^{-N} \leq \frac{\epsilon - \tilde{\epsilon} - La2^{-N}}{M+LR} \leq \frac{\epsilon-\tilde{\epsilon}}{M+LR} - La2^{-N}$$

Accordint to this we have that for to approach estimate 4 we need the following iterations number:

$$1 + \log_2 \min\left\{\frac{La}{\tilde{\epsilon}}, \frac{(M+LR)a}{\epsilon-\tilde{\epsilon}}\right\}$$

Of course, we assume in this estimate that $\tilde{\epsilon}$ and $\epsilon - \tilde{\epsilon}$ are positive. If it is not true we can trash a bad alternative.

The bad case of this iterations number's estimate occurs when

$$\tilde{\epsilon} = \frac{L}{M + L(R+1)}\epsilon.$$

In the bad case we have the following estimate for iterations number:

$$1 + \log_2 \frac{(M + L(R+1))a}{\epsilon} = O\left(\log \frac{1}{\epsilon}\right)$$

From this estimate we see that the our adaptive strategy needs the same iterations number in the bad case as the strategy **ConstGrad** needs always.

But for complexity estimation we have that for to solve one-dimensional problem we needs $O\left(\log \frac{1}{\epsilon}\right)$ iterations.

Theorem 8.1. *For to approach accuracy ϵ on function our method needs the following number of calculating function f and its derivatives of the first order:*

$$O\left(\log^2 \frac{1}{\epsilon}\right)$$

Proof. We have that for to solve one-dimensional problem one needs not more than

$$\log_2 \frac{Ca}{\epsilon}$$

iteration, where C is a constant determined by the used strategy and function's parameter and a is a size of current segment.

On each iteration of our method we solve to problems on segments of size $a_N = a2^{-N}$. Let $N_{\max 1} = \left\lfloor \log_2 \frac{Ca}{\epsilon} \right\rfloor$ is a maximal number of global method which the estimation for iteration number of one-dimensional problem is positive. And $N_{\max 2} = \left\lceil \log_2 \frac{La\sqrt{2}}{\epsilon} \right\rceil$ is estimation from 5.

$$N_{\max} := \min\{N_{\max 1}, N_{\max 2}\} = \log_2 \frac{1}{\epsilon} + O(1)$$

we have the following number of derivatives and function values calculation:

$$\sum_{k=0}^{N_{\max}} \log_2 \frac{Ca2^{-k}}{\epsilon} =$$

$$= \sum_{k=0}^{N_{\max}} \log_2 \frac{Ca}{\epsilon} - \sum_{k=0}^{N_{\max}} k =$$

$$= N_{\max} \log_2 \frac{Ca}{\epsilon} - \frac{1}{2} N_{\max}^2 + O\left(\log \frac{1}{\epsilon}\right) =$$

$$= \frac{1}{2} \log_2^2 \frac{1}{\epsilon} + O\left(\log \frac{1}{\epsilon}\right) =$$

$$= O\left(\log^2 \frac{1}{\epsilon}\right)$$

□

In the case of dual problems the complexity of function's parameters calculation depends on ϵ too. But accordinπ to 7.3 in this case the complexity will change by obvious way:

Theorem 8.2. *For to approach accuracy ϵ on function our method needs the following number of calculation of functions from primal problem and their derivatives:*

$$O(\log^3 \frac{1}{\epsilon})$$

9 Other Inexact Methods

Our optimization method can solve the task of minimization function f on square when the function and its gradient can not be calculated accurately but there are other optimization method for such tasks. In each section one describes some of them and below there is experimental comparison of them with our method.

The fist method is Primal Gradient Method (PGM) with (δ, L, μ) oracle. There are proves in the [7] that this method converges to the solution with accuracy δ:

$$\min_k f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{LR^2}{2} \exp\left(-k\frac{\mu}{L}\right) + \delta,$$

where $R = \|\mathbf{x}_0 - \mathbf{x}^*\|$ in our task. Moreover, in the [7] it is proved that for the function

$$f(\mathbf{x}) = \min_{\mathbf{u}} \left(\Psi(\mathbf{x}, \mathbf{u}) + \mathbf{u}^\top A \mathbf{x}\right)$$

there is following (δ, L, μ) oracle:

$$f_{\delta,L,\mu}(\mathbf{x}) = \Psi(\mathbf{x}, \mathbf{u}_\mathbf{x}) - \xi$$

$$g_{\delta,L,\mu}(\mathbf{x}) = A\mathbf{u}_{\mathbf{x}}$$

with parameters $\delta = 3\xi$, $L = \frac{2\lambda_{\max}(A^\top A)}{\mu(G)}$, $\mu = \frac{\lambda_{\min}(A^\top A)}{2L(G)}$ if $\mathbf{u}_{\mathbf{x}}$ is a solution approximation of $\mathbf{u}^*$ for current $\mathbf{x}$ with accuracy ξ on function.

The second method is Fast Gradient Method with (δ, L, μ) oracle. this method converges to the solution with accuracy δ:

$$\min_k f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \min\left(\frac{4LR^2}{k^2}, LR^2 \exp\left(-\frac{k}{2}\sqrt{\frac{\mu}{L}}\right)\right) + C_k\delta,$$

where $C_k = \min\left(\frac{k}{3} + \frac{12}{5}, 1 + \sqrt{\frac{L}{\mu}}\right)$. The method's description and proves for it is in the [7] too. This method has significantly better convergence rate for ill-conditioned problems.

The third having to be discussed method is inexact ellipsoid method. The ellipsoid method with ϵ-subgradient instead usual subgradient converges to a solution with accuracy ϵ:

$$\min_k f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \max_{\mathbf{x}\in Q} |f(\mathbf{x})| \exp\left(-\frac{k}{8}\right) + \delta$$

It is proved in [8]. Moreover, in [9] there is proved that for the function

$$f(\mathbf{x}) = \min_{\mathbf{u}} \Psi(\mathbf{x}, \mathbf{u})$$

the following statement is met:

$$\Psi(\mathbf{x}, \mathbf{u}_{\mathbf{x},\epsilon}) \in \partial_\epsilon f(\mathbf{x}),$$

if $\mathbf{u}_{\mathbf{x},\epsilon}$ is such point that $\Psi(\mathbf{x}, \mathbf{u}_{\mathbf{x},\epsilon}) - \min_{\mathbf{u}} \Psi(\mathbf{x}, \mathbf{u}) \leq \epsilon$.

Note that our method converge by the following way:

$$\min_k f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq Cexp(-k\ln 2),$$

and it is the best theoretical convergence rate from all methods in this section.

10 Tests

In this section we show estimate on number iterations of practice. Also there is comparison work time of our new method with work time of other optimization methods such as inexact gradient methods and inexact ellipsoids method[1]. All code was made in Anaconda 5.3.1 Python 3.6 (see cite [4])

[1] You can find all code in the repository [10]

10.1 Comparison With Other Methods

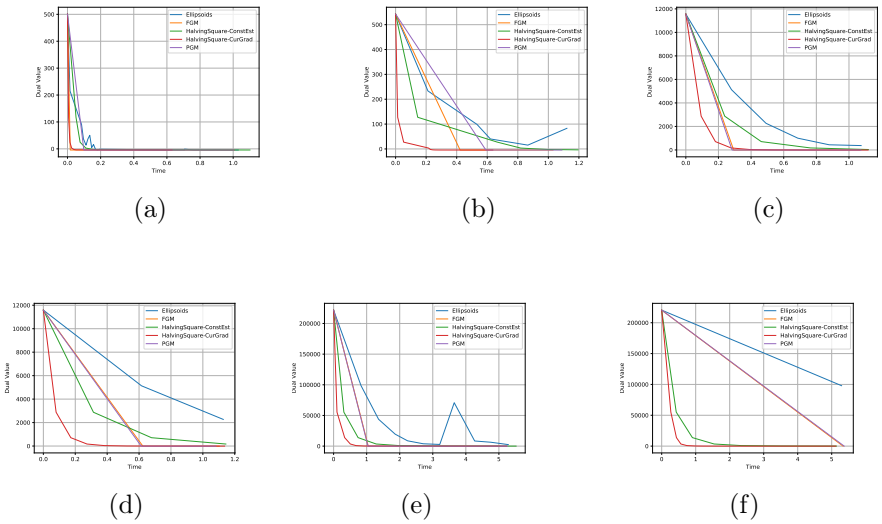

Figure 1: Comparison of different on inexact methods for task with different dimension N and for different required accuracy ϵ: (a) $N = 100, \epsilon = 10^{-3}$; (b) $N = 100, \epsilon = 10^{-10}$; (c) $N = 1000, \epsilon = 10^{-3}$; (d) $N = 1000, \epsilon = 10^{-10}$; (e) $N = 10000, \epsilon = 10^{-3}$; (f) $N = 10000, \epsilon = 10^{-10}$.

Let's compare our method with inexact ellipsoid method and gradient methods (PGM and FGM) with (δ, L, μ) oracle (see previous subsection 9) on a dual task.

For to find $\mathbf{x}(\lambda)$ we will use gradient descent. There are theoretical result that can help to manange distance $\|\mathbf{x}_k - \mathbf{x}^*\|$ from optimal point to its current approximation. In particular, there are following results:

- If f is a convex function with L-Lipschitz continious gradient then gradient descent with step $\alpha_k = \frac{1}{L}$ converges with speed

$$\|f(\mathbf{x}_k) - f(\mathbf{x}^*)\| \leq \frac{\|\mathbf{x}_0 - \mathbf{x}^*\|}{k+4}$$

- If f is a μ-strong convex function with L-Lipschitz continious gradient then gradient descent with step $\alpha_k = \frac{1}{L+\mu}$ converges with speed

$$\|f(\mathbf{x}_k) - f(\mathbf{x}^*)\| \leq \left(\frac{M-1}{M+1}\right)^k L\|\mathbf{x}_0 - \mathbf{x}^*\|,$$

where $M = \frac{L}{\mu}$.

The proves for this statements one can find in many books of optimization, for example, in the book [9].

We will use functions where μ is small enough. Therefore, our method for calculating $\mathbf{x}(\lambda)$ will converge to solution according to the first estimate.

For all inexact method we will calculate $\mathbf{x}(\lambda)$ with such accuracy as the method will converge to the solution with same for all methods accuracy ϵ. For PGM, FGM and ellipsoids methods we will calculate $\mathbf{x}(\lambda)$ with accuracy $\frac{\epsilon}{2}$ on function. For our method with the both strategies we will calculate $\mathbf{x}(\lambda)$ untill the conditions from the 13 is approached.

We consider the following prime task:

$$f(\mathbf{x}) = \ln\left(1 + \sum_{k=1}^{n} e^{\alpha x_k}\right) + \beta\|\mathbf{x}\|_2^2 \to \min_{\mathbf{x}\in\mathbb{R}^N} \tag{17}$$

$$g_k(\mathbf{x}) = \langle \mathbf{b}_k, \mathbf{x}\rangle + c_k \leq 0, k = \overline{1, m} \tag{18}$$

$$\tag{19}$$

It is task of minimization the LogSumExp-function with $l2$-regularization. The regularization parameter β determines strong convexity of our task and in the tests one takes $beta = 0.1$. The N is diminsionality of primal task and is determined for different tests below. The parameter α is equal to 1. The parameters c_k are equal to 1 too. The vectors $\{b_k\}_{k=1}^m$ are generated randomly for the each test. The m is equal to dimensionality of dual task and in the current case is equal to 2.

The LogSumExp-problem is L-Lipschitz continious function with M-Lipschitz continious gradient where $L = 1$ and $M = \alpha$. Therefore:

$$L_f = \alpha + 2\beta R, M_f = \alpha^2 + 2\beta,$$

$$\mu_f = 2\beta,$$

where $R = \|\mathbf{x}_0 - \mathbf{x}^*\|$ is the size of initial approximation. The functions g_k are L_k-Lipschitz continious where $L_k = \|b_k\|$.

We introduce the following notation:

$$\phi(\lambda_1, \lambda_2) = -\min_{\mathbf{x}\in\mathbb{R}^N} \left(f(\mathbf{x}) + \lambda_1 g_1(\mathbf{x}) + \lambda_2 g_2(\mathbf{x})\right) \tag{20}$$

In such notations the dual task for the task 17 looks like:

$$\phi(\lambda_1, \lambda_2) \to \min_{\lambda_1, \lambda_2} \tag{21}$$

$$\text{s.t}\, \lambda_1, \lambda_2 \geq 0 \tag{22}$$

Obviously, $\min_{\mathbf{x}} f(\mathbf{x}) \geq 0$. Therefore, according to 11 we can add following conditions on the dual variables:

$$|\lambda_k| \leq \lambda_{\max} = \frac{f(\overline{\mathbf{x}})}{\gamma}, k = 1, 2$$

And we have following task:

$$\phi(\lambda_1, \lambda_2) \to \min_{0 \leq \lambda_k \leq \lambda_{max}}$$

Calculating of function and derivatave value for such task was discussed in the section 7.

We can see on 1 the following results. Firstly, the halving square method with provided in this work strategy **CurGrad** are the fastest method in the most tests tests. This method can be slower than other inexact methods if dimensional of primal task is small or ϵ is big. In particular, this strategy is faster than strategy with constant estimate provided in [2]. It proves that provided by Nesterov method with strategy through gradient is the best method for to solve two dimensional dual task of minimization. Secondly, the gain of this strategy in comparison with other method is increase when the required ϵ decrease. This fact demonstrated important advantange of this strategy: it does not depend on required accuracy strongly. So, this method with constant estimate has strong dependity on it because there is this accuracy in the constant estimate, PGM and inexact ellipsoid method require that the $x(\lambda)$ is found with accuracy depended on ϵ. But halving square with ellipsoid method has not such dependety.

11 Conclusion

We discussed and proved that this method converges to the solution for smooth convex functon. Moreover, in the [2] there is conterexample when

the problem is non-smooth and we can not to converge to the solution with the accuracy on function better than a constant.

After it we discussed different strategy for one-dimensional task. Two strategies were considered. The both suggest to use stop conditions that are met when the current approximation is "very near" to the segment's solution. The first compares the distance between them with derivative value in accurate segment's solution but the second compares it with derivative value in approximation. In the experiment the first has a little better result but it can not be used for real task. The second strategy using derivative value in current approximation is significantly better then constant estimate and does not depend on required accuracy.

But all this strategy are good when the derivative value is high enough. But when the segment is near to the global solution this value will be small. Therefore one needs to make a lot of iterations on segment. For to avoid it we consider additional stop condition for global task on square when in current approximation the derivative value is near to zero.

The most steps of methods assumped that derivative can be calculated accurately. But the main method purpose is to solve dual problems and for it one can not usually calculate it so. That's why we consider different modifications of this method for to solve such problems. The important moment is we don't add some dependence on initial required accuracy in the modificated method.

Finally, we compared our method with new strategy for dual problem to prime LogSumExp problem with two linear constaints with our method with strategy using the constant estimate, primal gradient method and fast gradient method with (δ, L, μ) -oracle and with inexact ellipsoids methods. The Halving Square Method is the fastest of them for enough high dimension (more 100) and for enough high required solution ($1e-3$ and more).

References

[1] Gasnikov A. Universal gradient descent // MIPT — 2018, 240 p.

[2] Pasechnyk D.A., Stonyakin F.S. One method for minimization a convex Lipchitz continuous function of two variables on a fixed square // arXiv.org e-Print archive. 2018. – URL: https://arxiv.org/pdf/1812.10300.pdf

[3] Nesterov U.E. Methods of convex optimization // M.MCNMO — 2010, 262 p.

[4] Anaconda[site]. At available: https://www.anaconda.com

[5] Danskin, J.M.: The theory of Max-Min, with applications. J. SIAM Appl. Math.14(4) (1966)

[6] Fedor S. Stonyakin, Mohammad S. Alkousa, Alexander A. Titov,and Victoria V. Piskunova1 On Some Methods for Strongly Convex Optimization Problems with One Functional Constraint // ...

[7] Olivier Devolder Exactness, Inexactness and Stochasticityin First-Order Methods for Large-ScaleConvex Optimization // UCL — 2013,

[8] Need Reference To Book with Inexact Ellipsoids

[9] B.T. Polyak. The Introduction to Optimization // Moscow, Science - 1983

[10] Repository with code: https://github.com/ASEDOS999/Optimization-Halving-The-Square