КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет комп'ютерних наук та кібернетики

Кафедра обчислювальної математики

Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 113 Прикладна математика
на тему:

АДАПТИВНІ ТА ШВИДКІ АЛГОРИТМИ ОПТИМІЗАЦІЇ

Виконав студент 4-го курсу

Василенко Руслан Владиславович _____

(підпис)

Науковий керівник:

доктор фіз.-мат. наук
професор кафедри обчислювальної математики

Семенов Володимир Вікторович _____

(підпис)

Засвідчую, що в цій роботі
немає запозичень з праць інших авторів
без відповідних посилань.
Студент _____
(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри обчислювальної математики
«____» _____ 201_ р., протокол № ___
Завідувач кафедри Ляшко С. І.

_____
(підпис)

Київ – 2021

ЗМІСТ

РЕФЕРАТ 2
ВСТУП 3
1.UNIXGRAND (UNIVERSAL EXTRA GRADIENT) 5
1.1.ЗАГАЛЬНІ ОЗНАЧЕННЯ 6
1.2.АЛГОРИТМ 7
1.3.РІЗНІ ПОСТАНОВКИ 10
2.ACCELEGRAD 122
2.1.ADAGRAD 133
2.2.ОФЛАЙН ПОСТАНОВКА 133
3.ДОВЕДЕННЯ ТЕОРЕМ 177
4.ЧИСЕЛЬНІ ДОСЛІДЖЕННЯ 179
5.ПРИКЛАДИ КОДУ 24
ВИСНОВКИ 25
ПОСИЛАННЯ 26

РЕФЕРАТ

Робота має 30 сторінок, чотири алгоритми, вісім теорем, два рисунки та дві леми.

Об’єктом дослідження роботи є адаптивні алгоритми оптимізації направленні на пошук мінімуму функції.

Мета роботи полягає в розробці та дослідженні алгоритмів опуклої оптимізації, що поєднують в собі гарні риси швидких градієнтних та адаптивних алгоритмів. Ідея полягає в заміні константи Ліпшиця градієнта функції в кроці алгоритмів на адаптивний множник з AdaGrad.

Методом дослідження є аналіз інших алгоритмів для побудови розглянутих у цій роботі та їх програмування на мові Python 3 з метою отримання графічних результатів роботи.

Дані алгоритми є сучасними і використовують підхід, що дозволяє їх використання для розв’язання великого кола проблем, в основному класифікації.

Алгоритми наведені в даній роботі можна використовувати в області медицини, бо вони являються доволі швидкими та мають високу точність для розв’язання проблем класифікації.

ВСТУП

На даний час існує дуже багато різних алгоритмів оптимізації, але більша кількість популярних алгоритмів застосовується лише для вирішення однієї проблеми, тобто для випадків гладких та негладких функцій будуть використовуватися різні алгоритми, які потребують великої кількості апріорних знать, таких як дисперсія та опуклість функції [1-8].

Алгоритми представлені в даній роботі відходять від такого підходу та зосереджують увагу на речах, які раніше не розглядалися, для розуміння того як правильно вибрати напрямок руху, використовуючи інформації про градієнт, а не тільки використовувати його як основу, бо як ми знаємо напрямок найбільшого спадання функції в більшості випадків приводить нас до локальних екстремумів.

Ще один з аспектів які не розглядалися раніше для прискорення знаходження екстремумів – інформація про гладкість функції та розглядання ліпшицевих функцій. Така інформація використовується в даній роботі разом з переходом до дивергенції Брегмана замість звичайної евклідової норми, що використовується в більшості алгоритмів, що побудовані на основі методів градієнтного та субградієнтного спусків.

Також буде розглянутий підхід в якому інформацію яку ми будемо отримувати з роботи базових алгоритмів оптимізації дасть нам змогу побудувати описаний в даній роботі алгоритм та показати його перевагу.

В межах розглянутих в роботі алгоритмів будуть наведені налаштування, коли ми не можемо отримати точне значення градієнта в точці, але лише деяку його оцінку, використовуючи умовне математичне сподівання, та розглянемо збіжність такого алгоритму. Доведення таких випадків в даній роботі не буду наводити через те, що підхід до доведення майже не буде відрізнятися від детермінованого випадку.

Отже, використовуючи всі ці підходи ми маємо описані в даній роботі алгоритми з доведеннями оцінки їх збіжності та інформації про те як ці підходи надають змогу вибрати правильний крок для уникнення випадків «перескакування» глобального мінімуму функції.

Після того як алгоритми будуть розглянуті ми перейдемо до їх порівняння на прикладі задачі класифікацій.

1.UNIXGRAND (UNIVERSAL EXTRA GRADIENT)

В межах алгоритмів, що побудовані стохастичної оптимізації на обмеженому множині, оптимальна швидкість збіжності як для випадків випуклих так і невипуклих функцій визначається як $O\left(\frac{GD}{\sqrt{T}}\right)$, $O\left(\frac{LD^2}{T^2}+\frac{\sigma D}{\sqrt{T}}\right)$, де T – загальна кількість градієнтів, L – константа опуклості, σ^2 – дисперсія наближення градієнта, D – діаметр множини допустимих розв'язків, G – границя градієнтних наближень. Такі оцінки потребують додаткових обрахунків для більш точних результатів.

Оптимальна оцінка для більшості випадків невипуклих функцій може бути досягнута за використання таких алгоритмів як стохастичний спуск, або AdaGrad. В той час як для випадків опуклої функції оптимальна оцінка отримується за допомогою прискорених методів.

Проблема полягає в тому, що для використання вище наведених алгоритмів ми потребуємо деяких початкових знать про константу L та відхилення σ. Запропонований в даній роботі алгоритм, що базується на алгоритмі Mirror-Prox [1] та досягає оптимальних оцінок як у випадку гладкої так і негладкої функції, а також не потребує вище наведених початкових параметрів.

Далі буде наведено означення, що будуть потрібні для побудови алгоритму.

1.1.ЗАГАЛЬНІ ОЗНАЧЕННЯ

Означення 1. Функція $f: \mathbb{R}^n \to \mathbb{R}$ є $\beta -$ опуклою, якщо:

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{\beta}{2\|x - y\|^2}, \forall x, y \in \mathbb{R}^n$$

Означення 2. Функція $f: K \to \mathbb{R}$ є L-гладкою над K, якщо вона має градієнт Ліпшица по L, тобто:

$$\forall x, y \in K, \|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\|$$

Означення 3. Функція f з елементами $x_1, \ldots x_n$ та вагами a_i є випуклою, якщо виконується нерівність Йенсена:

$$f\left(\frac{\sum a_i x_i}{\sum a_i}\right) \leq \frac{\sum a_i f(x_i)}{\sum a_i}$$

Означення 4. Нехай функція $\mathcal{R}: K \to \mathbb{R}$ є 1-опуклою та диференційованою, тоді дивергенцію Брегмана від $\mathcal{R}$ називається функція

$$\mathcal{D}_{\mathcal{R}}(x, y) = \mathcal{R}(x) - \mathcal{R}(y) - \langle \nabla \mathcal{R}(y), x - y \rangle$$

Важлива особливість дивергенції Брегмана

$$\mathcal{D}_{\mathcal{R}}(x, y) \geq \frac{1}{2}\|x - y\|^2, \forall x, y \in K$$

1.2.АЛГОРИТМ

UniXGrad [7] – алгоритм, що використовується для розв'язання проблеми пошуку мінімуму функції

$$\min_{x \in K} f(x)$$

де f: K → $\mathbb{R}$ опукла функція та K ⊂ $\mathbb{R}^d$ опукла множина.

$$\mathbb{E}\left[\tilde{\nabla} f(x) | x\right] = \nabla f(x) \quad (1)$$

$$\left\|\tilde{\nabla} f(x)\right\|_* \leq G, \forall x \in K$$

Як вказувалось вище даний алгоритм побудований на основі Mirror-Prox [1], тому розглянемо його та відповідний Optimistic Mirror Descent алгоритм.

Ціль даного алгоритму – оптимізація випуклої функції f в випуклій множині K. Даний алгоритм виконує крок від y_{t-1} до x_t, базуючись на інформації про градієнт в точці y_{t-1}. Далі ми повертаємося до y_{t-1} і робимо новий крок, але в цей раз на базі інформації про градієнт в точці x_t. Кожен крок будується на основі дивергенції Брегмана.

Алгоритм 1.

Вхідні дані: кількість ітерацій **T**, $y_0 \in K$, шкидкість навчання $\{\eta_t\}_{t \in [T]}$

1. Для t = 1,…., T виконується
2. $x_t = \underset{x \in K}{\operatorname{argmin}} \langle x, M_t \rangle + \frac{1}{\eta_t} * D_{\mathcal{R}}(x, y_{t-1})$
3. $y_t = \underset{y \in K}{\operatorname{argmin}} \langle y, g_t \rangle + \frac{1}{\eta_t} * D_{\mathcal{R}}(y, y_{t-1})$
4. Кінець циклу

Тепер розберемо різницю між Mirror-Prox та UniXGrad відносно вибору g_t та M_t та функції $\mathcal{R}$. Optimistic Mirror Descent має $g_t = \nabla f(x_t)$ та обраховує $M_t = \nabla f(x_{t-1})$ базуючись на інформації про градієнт на попередніх кроках. Даний вектор є доступним на кожній ітерації циклу і алгоритм стає оптимізованим, коли $M_t \approx g_t$. Mirror-Prox стає Extra-Gradient, коли $\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$ базуючись на нормі в Евклідовому просторі.

Даний алгоритм швидко розв'язує дану проблему з оцінкою $\mathcal{O}(\frac{1}{T})$, але у випадку гладкої випуклої оптимізації не задовольняє оцінці $\mathcal{O}(\frac{1}{T^2})$ для випадку мінімізації.

Ми введемо зміни до даного алгоритму, а саме вибір g_t, M_t, швидкості навчання та вагів.

Усереднення. Точки в яких ми будемо шукати значення g_t, M_t визначаються наступним чином, враховуючи ваги $\alpha_t = t$:

$$\overline{x_t} = \frac{\alpha_t x_t + \sum_{i=1}^{t-1} \alpha_i x_i}{\sum_{i=1}^{t} \alpha_i} \quad (2)$$

$$\widetilde{z_t} = \frac{\alpha_t y_{t-1} + \sum_{i=1}^{t-1} \alpha_i x_i}{\sum_{i=1}^{t} \alpha_i}$$

Отже для UniXGrad ми маємо $g_t = \nabla f(\overline{x_t})$ та $M_t = \nabla f(\widetilde{z_t})$.

Швидкість навчання. Визначимо швидкість навчання як:

$$\eta_t = \frac{2D}{\sqrt{1 + \sum_{i=1}^{t-1} a_i^2 \|g_i - M_i\|_*^2}}, \quad (3)$$

де $D^2 = \sup_{x,y \in K} \mathcal{D}_{\mathcal{R}}(x, y)$ – брегманівський діаметр множини K.

Вибір вагів. Важливо, щоб $a_t = \Theta(t)$ для досягнення оптимальної швидкості, тому ми беремо початкове значення рівне t.

Враховуючи вище наведені зміни маємо алгоритм:

Алгоритм 2.

Вхідні дані: кількість ітерацій **T**, , $y_0 \in K$,швидкість навчання $\{\eta_t\}_{t\in[T]}$,ваги α_t та діаметр D

1. Для *t = 1,….*, *T* виконується
2. $x_t = \underset{x\in K}{argmin}\langle x, M_t\rangle + \frac{1}{\eta_t} * D_{\mathcal{R}}(x, y_{t-1})$

$$M_t = \nabla f(\widetilde{z_t})$$

3. $y_t = \underset{y\in K}{argmin}\langle y, g_t\rangle + \frac{1}{\eta_t} * D_{\mathcal{R}}(y, y_{t-1})$

$$g_t = \nabla f(\overline{x_t})$$

4. Кінець циклу
5. Повертаємо $\overline{x_T}$

Далі буде наведено теореми, що доводять збіжність даного алгоритму. Не поглиблюючись в аналіз його буде спрощено до аналізу збіжності в умовах обмеження значень вагів. Згідно з цього буде наведено стратегію переведення цих вагів в швидкість збіжності.

Лема 1 ([7]). Нехай задано зважене середнє $\overline{x_t}$ з рівності (2). Також визначимо $R_T(x_*) = \sum_{t=1}^{T} \alpha_t\langle x_t - x_*, g_t\rangle$, що позначає значення вагів після T ітерацій, $\alpha_t = t, g_t = \nabla f(\overline{x_t})$ тоді

$$f(\overline{x_T}) - f(x_*) \leq \frac{2R_T(x_*)}{T^2}.$$

1.3.РІЗНІ ПОСТАНОВКИ

Негладкі постановки

- Детермінована постановка

  Теорема 1 ([7]). Нехай $f: K \to \mathbb{R}$ власна, опукла та є ліпшицевою функцію з константою G, що визначена на компактній та випуклій множині *K*. Також $x^* \in \min\limits_{x \in K} f(x)$. Тоді Алгоритм 2 гарантує виконання нерівності:

$$f(\overline{x_T}) - \min_{x \in K} f(x) \leq \frac{7D\sqrt{1 + \sum_{t=1}^{T} a_t^2 \|g_t - M_t\|_*^2} - D}{T^2} \leq \frac{6D}{T^2} + \frac{14GD}{\sqrt{T}}$$

- Стохастична постановка

  Теорема 2 ([7]). Нехай $f: K \to \mathbb{R}$ не гладка та опукла функція. $\{x_t\}_{t=1,\dots,T}$ – послідовність згенерована UniXGrad така що $g_t = \widetilde{\nabla} f(\overline{x_t})$ та $M_t = \widetilde{\nabla} f(\widetilde{z_t})$ з $\alpha_t = t$ та швидкістю навчання визначеною в (3), тоді

$$\mathbb{E}[f(\overline{x_T})] - \min_{x \in K} f(x) \leq \frac{6D}{T^2} + \frac{14GD}{\sqrt{T}}$$

Гладкі постановки

- Детермінована постановка

  Теорема 3 ([7]). Нехай $f: K \to \mathbb{R}$ власна, опукла та *L* – гладка функція, що визначена на компактній та випуклій множині *K*. Також $x^* \in \min\limits_{x \in K} f(x)$. Тоді Алгоритм 2 гарантує виконання нерівності:

$$f(\overline{x_T}) - \min_{x \in K} f(x) \leq \frac{20\sqrt{7}D^2L}{T^2}$$

- Стохастична постановка

Теорема 4 ([7]). Нехай $f: K \to \mathbb{R}$ L - гладка та опукла функція. $\{x_t\}_{t=1,\dots,T}$ – послідовність згенерована UniXGrad така що $g_t = \widetilde{\nabla} f(\overline{x_t})$ та $M_t = \widetilde{\nabla} f(\widetilde{z_t})$ з $\alpha_t = t$ та швидкістю навчання визначеною в (3), тоді

$$\mathbb{E}[f(\overline{x_T})] - \min_{x \in K} f(x) \leq \frac{112\sqrt{14}D^2L}{T^2} + \frac{14\sqrt{2}\sigma D}{\sqrt{T}}$$

2.ACCELEGRAD

Accelegrad розлоблений Нестеровим розглядає ту саму задачу, що й UniXGrad, а саме: дана опукла функція $f: \mathbb{R}^d \to \mathbb{R}$, ціль знайти

$$\min_{x \in \mathbb{R}^d} f(x),$$

також даний метод базується лише на інформації відомій про градієнт і використовується як у випадку гладкої так і негладкої функцій *f*.

Як і у випадку UniXGrad ми не маємо ніякої інформації, щодо параметра гладкості функції, але ми визначаємо границю відстані від деякої точки x_0 до глобального мінімуму функції f шляхом використання діаметру заданої опуклої множини $K \subset \mathbb{R}^d$ якій належить точка x_0. Даний діаметр визначимо як

$$D \coloneqq \max_{x,y \in K} \|x - y\|.$$

Також зазначимо, що ми можемо брати точки, що не входять в множину *K* і функція *f* є функією Ліпшица по *G*.

Ми не завжди матимемо доступ до градієнта, тому в цих випадках ми будемо використовувати деяку шумну оцінку градієнта.

Оскільки даний алгоритм побудований на базі AdaGrad [2], то коротко розглянемо його основні моменти.

2.1.ADAGRAD

Алгоритм 3. Маємо кількість ітерацій T, $x_1 \in K$

1. Для $t = 1, \ldots, T$
2. $g_t = \nabla f(x_i)$
3. $\eta_t = D(2\sum_{i=1}^{t}\|g_T\|^2)^{-1/2}$
4. $x_{t+1} = \Pi_K(x_t - \eta_t g_t)$
5. Завершення циклу
6. Вивід $\overline{x_T} = \frac{1}{T}\sum_{t=1}^{T} x_t$

Теорема 5 ([2, 4]). Нехай K опукла множина з діаметром D та f опукла функція. Тоді алгоритм 3 гарантує похибку:

$$f(\overline{x_T}) - \min_{x \in K} f(x) \leq \sqrt{2D^2 \sum_{t=1}^{T} \|g_t\|^2 / T}$$

Зауважимо, що під $\Pi_K(x)$ – проекція на опуклу множину K $\forall x \in \mathbb{R}^d, \Pi_K(x) = arg\min_{y \in K}\|y - x\|^2$.

2.2.ОФЛАЙН ПОСТАНОВКА

Офлайн налаштування означає, що ми маємо доступ до градієнта. Алгоритм наведений вище гарантує оцінку $O\left(\frac{1}{T^2}\right)$ у випадку гладкої функції та $O(\sqrt{\frac{\log T}{T}})$ в загальному випадку опуклої функції. AcceleGrad буде лінійно пов'язувати послідовності $\{z_t\}_t, \{y_t\}_t$ в послідовність $\{x_{t+1}\}_t$. Тобто, за допомогою градієнта $g_t = \nabla f(x_{t+1})$ ці послідовності будуть оновлюватися з однаковою швидкістю навчання η_t, але з різними опорними точками та значеннями градієнта. Мається на увазі, що ми робимо крок від точки x_{t+1} до

точки y_{t+1}, далі з точки z_t робимо крок до точки z_{t+1} на основі проекція зваженого градієнта. В кінці алгоритм виводить зважене середнє послідовності $\{y_{t+1}\}_t$.

Алгоритм 4. Accelegrad [4, 8]

Маємо кількість ітерацій T, $x_0 \in K$, діаметр D, ваги $\{\alpha_t\}_{t\in[T]}$, швидкість навчання $\{\eta_t\}_{t\in[T]}$ та визначимо, що $y_0 = x_0 = z_0$

1. Для $t = 1, \ldots, T$
2. $\tau_t = 1/\alpha_t$
3. $x_t = \tau_t z_t + (1-\tau_t)y_t$,
   де $g_t \coloneqq \nabla f(x_{t+1})$
4. $z_{t+1} = \Pi_K(z_t - \alpha_t \eta_t g_t)$
5. $y_{t+1} = x_{t+1} - \eta_t g_t$
6. Кінець циклу
7. Вивід $\overline{y_T} \propto \sum_{t=0}^{T-1} \alpha_t y_{t+1}$

Для даного алгоритму нижче наведені незалежні від константи гладкості параметри:

$$\eta_t = \frac{2D}{\sqrt{G^2 + \sum_{\tau=0}^{t} \alpha_\tau^2 \|g_\tau\|^2}}, \alpha_t = \begin{cases} 1, 0 \le t \le 2 \\ \dfrac{1}{4(t+1)}, t \ge 3 \end{cases} \quad (4), (5)$$

Стосовно поведінки алгоритму у гладкому випадку відомо наступне:

- Детермінована постановка:

  Теорема 6. Нехай *f* опукла та β-гладка функція, що визначена на опуклій множині *K* з діаметром *D* та також існує мінімум цієї функції в *K*, тоді вище наведений алгоритм гарантує таку оцінку:

$$f(\overline{y_T}) - \min_{x \in \mathbb{R}^d} f(x) \leq \frac{DG + \beta D^2 \log(\beta D/G)}{T^2}$$

Уточнення: Константа Ліпшица потрібна лише у випадку негладкої функції, тому, якщо ми знаємо, що функція є гладкою, то ми можемо змінити швидкість навчання $\eta_t = \frac{2D}{\sqrt{\sum_{\tau=0}^{t} \alpha_\tau^2 \|g_\tau\|^2}}$, за допомогою якої отримуємо оцінку $O(\frac{\beta D^2 \log(\frac{\beta D}{\|g_0\|})}{T^2})$.

- Стохастична постановка:

  Нехай *f* опукла та β-гладка функція, що визначена на опуклій множині *K* з діаметром *D* та також існує мінімум цієї функції в *K*, тоді ми використовуємо алгоритм Алгоритм 3, але з використанням шумних наближень градієнта (1) і маємо таку оцінку:

$$\mathbb{E}[f(\overline{x_T})] - \min_{x \in \mathbb{R}^d} f(x) \leq \frac{\beta D^2}{T} + \frac{\sigma D}{\sqrt{T}}$$

Розглянемо негладкий варіант:

- Детермінована постановка:

  Теорема 7. Нехай *f* опукла ліпшицева функція з константою *G*, що визначена на опуклій множині *K* з діаметром *D* та також існує мінімум цієї функції в *K*, тоді вище наведений алгоритм гарантує таку оцінку:

$$f(\overline{y_T}) - \min_{x \in \mathbb{R}^d} f(x) \leq GD\sqrt{logT}/\sqrt{T}$$

- Стохастична постановка:

Теорема 8.Нехай f опукла ліпшицева функція з константою G, що визначена на опуклій множині K з діаметром D та також існує мінімум цієї функції в K, тоді ми використовуємо алгоритм Алгоритм 4, але з використанням шумних наближень градієнта (1) і маємо таку оцінку:

$$\mathbb{E}[f(\overline{x_T})] - \min_{x \in \mathbb{R}^d} f(x) \leq GD\sqrt{logT}/\sqrt{T}$$

3.ДОВЕДЕННЯ ТЕОРЕМ

В цьому розділі будуть наведені необхіді для доведення теорем леми та нерівності. Доведення наведених в роботі теорем можна переглянути в роботах [4, 7, 8].

Лема 2. $\{\alpha_i\}_{i=1,..,n}$ – невід'ємна послідовність:

$$\sqrt{\sum_{i=1}^{n} a_i} \leq \sum_{i=1}^{n} \frac{a_i}{\sum_{j=1}^{i} a_j} \leq 2\sqrt{\sum_{i=1}^{n} a_i}$$

Визначення. Подвійною нормою функції f з визначенною в ній нормою $\|\cdot\|$ називають таке невід'ємне число

$$\| f \|_* = sup\{|f(x)| : \| x \| \leq 1 \text{ та } x \in X\}$$

Лема 3. Нехай b_i невід'ємна послідовність, а α_t ваги визначенні для Алгоритма 4, тоді

$$\sum_{t=0}^{T-1} \frac{\alpha_t b_t}{(1 + \sum_{\tau=0}^{t} \alpha_\tau^2 b_\tau^2)^{1/2}} \leq 5\sqrt{\log T}\sqrt{T}$$

Лема 4. $u, v, z \in \mathbb{R}^d$ тоді для $\mathcal{R}_v(u) = 1/2\|u - v\|^2$

$$-\nabla\mathcal{R}_v(u) \cdot (u - z) = \frac{1}{2} \| v - z \|^2 - \frac{1}{2} \| u - z \|^2 - \frac{1}{2} \| u - v \|^2$$

Лема 5.

$$\begin{aligned} &\alpha_t g_t \cdot (z_t - z) \\ &\qquad \leq \left(\alpha_t g_t \cdot (z_t - z_{t+1}) - \frac{1}{2\eta_t}\|z_t - z_{t+1}\|^2\right) \\ &\qquad + \frac{1}{2\eta_t}\left(\|z_t - z\|^2 - \|z_{t+1} - z\|^2\right) \end{aligned}$$

Лема 6. Для невід'ємної послідовності

$$\sum_{i=1}^{n} \frac{a_i}{1+\sum_{j=1}^{i} a_j} \leq 1 + log\left(1 + \sum_{i=1}^{n} a_i\right)$$

Нерівність Гельдера.

$$\boldsymbol{u^T v} \leq \|\boldsymbol{u}\|\|\boldsymbol{v}\|_*$$

Нерівність Коші-Буняковського

$$\boldsymbol{u^T v} \leq \|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2$$

4. ЧИСЕЛЬНІ ДОСЛІДЖЕННЯ

В цьому розділі ми порівняємо ці два алгоритми. Тестування буде відбуватися на основі SVM (Support Vector Machine) задачі на базі двох датасетів, які будемо брати з LIBSVM.

Задача виглядає наступним чином:

$$y \in \{-1,1\}^n, X \in \mathbb{R}^{n \times p} \text{ з рядками } x_1, x_2, \dots, \boldsymbol{x}_n$$

$$\min_{\boldsymbol{w},\xi} \frac{1}{2} \| \boldsymbol{w} \|_2^2 + \lambda \sum_{i=1}^{n} \xi_i$$

$$\text{де } \xi_i \geq, y_i(\boldsymbol{x}_i^\top \boldsymbol{w}) \geq 1 - \xi_i, i = 1, \dots, n$$

Перепишемо нерівність:

$$\xi \geq max\{0,1 - y_i(\boldsymbol{x}_i^\top \boldsymbol{w})\}$$

Підставляючи в нашу задачу, маємо:

$$\min_{\boldsymbol{w}} \frac{1}{2} \| \boldsymbol{w} \|_2^2 + \lambda \sum_{i=1}^{n} max\{0,1 - y_i(\boldsymbol{x}_i^\top \boldsymbol{w})\}$$

Але в нашому випадку ми будемо розглядати квадрат:

$$\min_{\boldsymbol{w}} \frac{1}{2} \| \boldsymbol{w} \|_2^2 + \lambda \sum_{i=1}^{n} (max\{0,1 - y_i(\boldsymbol{x}_i^\top \boldsymbol{w})\})^2$$

Датасет представляє собою міні батчі з п’яти елементів, де кожен елемент має порядковий номер та ймовірний діагноз раку легенів, де ваги будуть ініціалізуватися випадковим чином, тобто задача в загальному розумінні полягає в класифікацій діагнозу.

Таблиця датасетів має велику кількість стовбців основними з яких є:

діагноз, радіус, гладкість, площа, текстура, симетричність зони ураження та зони за нею. Даний датасет створений Ф. Волбергом та Н. Стрітом з Університету Вісконсіна.

Алгоритмічно визначені параметри мають наступний вигляд:

- UniXGrad
  - Діаметр D = 10;
  - Константа гладкості L = 0,001.
- Accelegrad
  - Діаметр D = 10;
  - Константа гладкості L = 0,1.

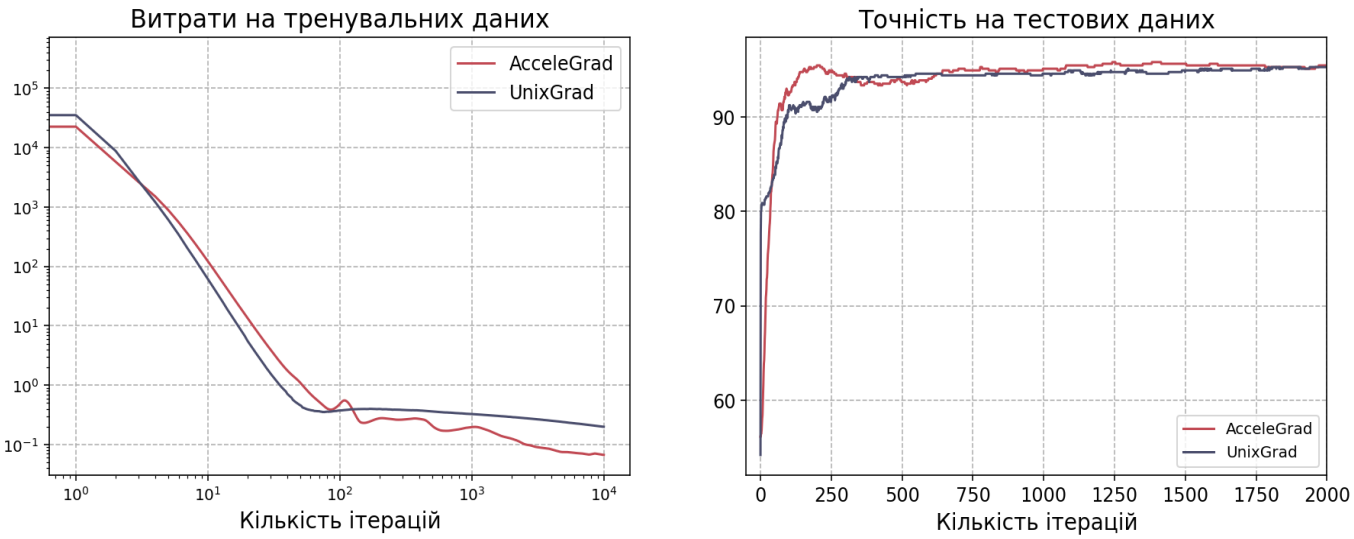

Рис. 1. Порівняння алгоритмів

З цих результатів ми бачимо явне порівняння цих двох алгоритмів. Повертаючись до Accelegrad ми можемо побачити деякий феномен зображений на Рис. 2. Зі збільшенням розмірності датасетів швидкість збіжності зростає. Це відбувається тому що з маленькою розмірністю *b*

наближення градієнта є більш шумним. Якщо визначити кількість ітерацій як $N = bT$, то $T = N/b$, тобто якщо застосувати це до наближення $O(\frac{1}{\sqrt{T}})$ для стохастичного випадку та $O(\frac{1}{T^2})$ для детермінованого випадку, то ми отримуємо такі наближення відповідно $O(\frac{\sqrt{b}}{\sqrt{N}})$ та $O(b^2/N^2)$.

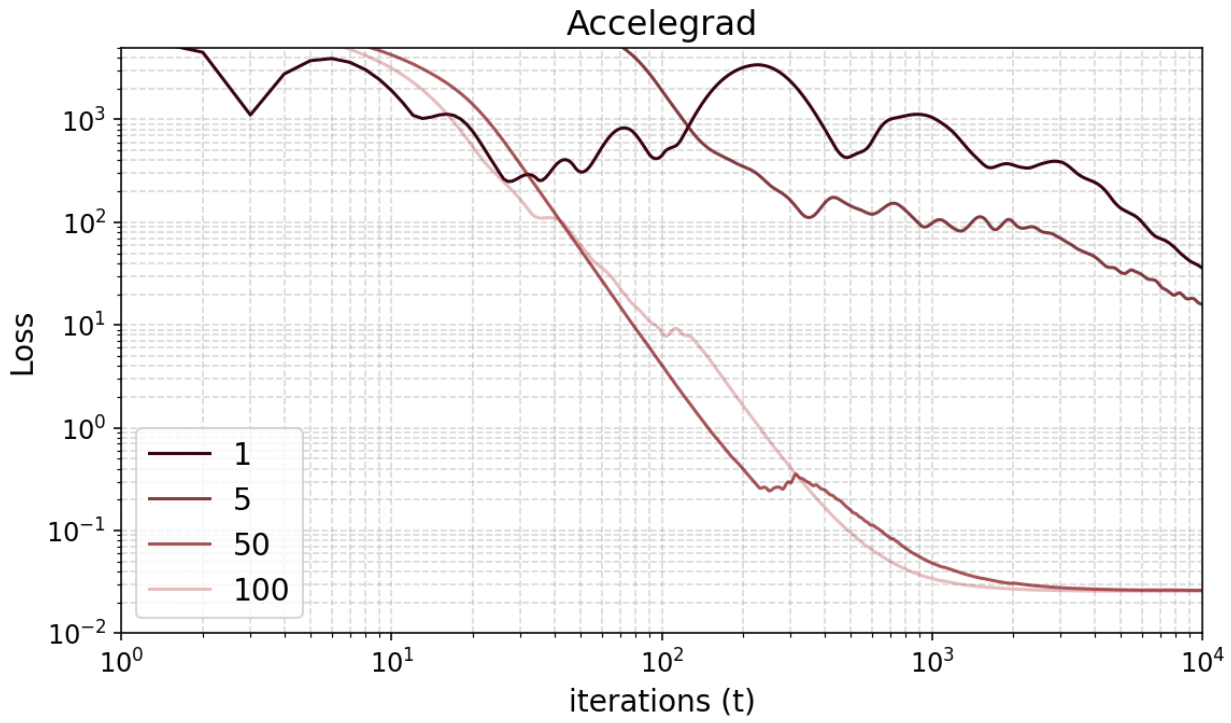

Рис. 2.Accelegrad на різній розмірності датасетів

Для логістичної регресії ми маємо наступну точність на тестових даних

Рис. 3

Для звичайної лінійної регресії

Рис. 4

Для L1 – регуляризатора

Рис. 5

4.ПРИКЛАДИ КОДУ

Основний цикл Accelegrad

```
for t in range(max_iter+1):

    # обирання батчу
    batch, label = random_batch_selection(train_batches, train_labels)

    alphat = np.maximum( 0.25*(t+1), 1 )
    taut = 1/alphat

    # Правило оновлення
    xt = taut * zt + (1-taut)*yt

    # Сабградієнт
    gt = compute_grad(xt, lamb, batch, label)

    ng = np.linalg.norm(gt)

    S1 = S1 + (alphat*ng)**2

    etat = D/np.sqrt( 2*(G**2 + S1) )
    zt = gradient_projection(zt - alphat*etat*gt,D)

    yt = xt - etat*gt

    # Усереднення
    S2 = S2 + alphat
    weight = alphat/S2
    ybar =  (1 - weight)*ybar + weight*yt

    loss = squared_hinge_loss(ybar, lamb, x_train, y_train)
    loss_hist.append(loss)

    # Спроба визначити наступне значення
    prediction = predict(x_test, ybar)
    accuracy_hist.append( get_accuracy( prediction, y_test) )

return loss_hist, accuracy_hist, ybar
```

ВИСНОВКИ

Як ми бачимо то дані алгоритми показують гарні результати для задачі класифікації і можуть використовуватися для такого роду проблем, що можуть виникати у медицині, або ж для швидкої класифікацій та збору статистики.

Алгоритми представлені в даній роботі є простими для розуміння в плані їх основи на більш простих алгоритмах таких як Mirror Descent та Sub-gradient Descent. Останній алгоритм дуже часто обирає неправильний напрямок руху до екстремуму функції або ж взагалі «перескакує» його, через що вибір правильного кроку потребує індивідуального підходу до кожної проблеми, тому що він не має явної залежності від константи гладкості. Враховуючи це робота проведена для встановлення такої залежності дає змогу краще оцінити збіжність алгоритму та вибрати залежний від константи Ліпшица крок.

Також перехід до дивергенції Брегмана надав змогу швидше знаходити правильний напрямок руху. Тому на такій базі можна будувати швидкі адаптивні алгоритми оптимізації для знаходження глобальних екстремумів різного виду опуклих функцій.

ПОСИЛАННЯ

1. Nemirovski A. Prox-method with rate of convergence o (1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems, SIAM Journal on Optimization, vol. 15, no. 1, pp. 229 - 251, 2004
2. Duchi J., Hazan E., Singer Y. Adaptive subgradient methods for online learning and stochastic optimization, Journal of Machine Learning Research, vol. 12, pp. 2121-2159,07 2011.
3. Zhu Z. A., Orecchia L. A novel, simple interpretation of Nesterov's accelerated method as a combination of gradient and mirror descent, CoRR, vol. abs $/1407.1537, 2014$.
4. Levy K. Y., Yurtsever A., Cevher V. Online adaptive methods, universality, and acceleration, 2018.
5. Shalev-Shwartz S. Online learning and online convex optimization, Foundations and Trends $\hat{A}(R)$ in Machine Learning, vol. 4, no. 2, pp. 107-194, 2012.
6. Cutkosky A. Anytime online-to-batch conversions, optimism, and acceleration, 2019.
7. Kavis A., Levy K. Y., Bach F., Cevher V., Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization, 2019.
8. Кравець О.П. Прискорені адаптивні методи опуклої оптимізації. Кваліфікаційна робота на здобуття ступеня бакалавра. ФКНК КНУ імені Тарса Шевченка. 2020. 39 с.