Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Кафедра Обчислювальної математики

КУРСОВА РОБОТА

за напрямом 6.040301 Прикладна математика
на тему:

Розподілена оптимізація і статистичне навчання за допомогою методу множників, що змінюють напрямок

Виконав студент 3-го курсу
Скибицький Нікіта Максимович _____

Науковий керівний
доктор фіз.-мат. наук, професор
Клюшин Дмитро Анатолійович _____

Засвідчую, що в цій курсовій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ — 2019

Зміст

Анотація

Багато актуальних задач статистики та машинного навчання можуть бути сформульовані у термінах опуклої оптимізації. Сьогодні розміри та складність даних зростають з експоненційною швидкістю, і разом із цим стає все більш важливим вміти розв'язувати задачі із великою кількістю ознак у об'єктів та/або великою кількістю самих об'єктів. Як наслідок, як децентралізований збір та збереження даних, так і розподілені методи розв'язування задач є необхідними, або принаймні дуже бажаними. У цьому огляді ми намагаємося показати, що метод множників які змінюють напрямок гарно підходить для розподіленої опуклої оптимізації, і зокрема до задач великого масштабу, що виникають у статистиці, машинному навчанні, та суміжних областях. Метод був розроблений у 1970-их, з коренями у 1950-их, і є еквівалентним, або тісно пов'язаним з багатьма іншими алгоритмами, такими як двоїстий розклад, метод множників (Лагранжа), розбиття Дугласа-Рашфорда, метод часткових обернених Спінгарна, метод почергових проекцій Дейкстри, ітеративними алгоритмами Брегмана для задач з нормою ℓ_1, проксимальними (eng. *proximal*) методами, та іншими. Після короткого огляду теоретичних результатів та історії алгоритму, ми обговоримо його застосування до широкого кола актуальних задач статистики та машинного навчання, включаючи ласо (eng. *lasso*), розріджену логістичну регресію, вибір базису (eng. *basis pursuit*), оцінку матриці коваріації (eng. *covariance selection*), машини опорних векторів (eng. *support vector machine*), та багато інших.

1 Вступ

Зараз у всіх прикладних областях розповсюдженим підходом до розв'язування задач є застосування аналізу даних, зокрема використання алгоритмів статистики та машинного навчання, зачасту на великих наборах даних. У промисловості така "мода" здобула назву "Big Data", і вона вже маж суттєвий вплив на такі різноманітні області як штучний інтелект, мережеві застосунки, обчислювальна біологія, медицина, фінанси, маркетинг, журналістика, аналіз мереж, та логістика.

Незважаючи на те, що ці задачі виникають у різноманітних прикладних областях, вони поділяють кілька ключових характеристик. По-перше, дані зачасту надзвичайно великі, і можуть складатися з сотень мільйонів або навіть мільярдів тренувальних прикладів. По-друге, дані часто мають великі розмірність, оскільки зараз стало можливим вимірювати і зберігати дуже деталізовану інформацію про кожний об'єкт. Як наслідок, стає життєво необхідною розробка алгоритмів, як водночас достатньо складні для опису складної структури сучасних даних, і достатньо легко застосовуються до паралельної, або цілком розподіленої обробки гігантських даних. Насправді, деякі дослідники [1] припускають, що навіть задачі дуже складної структури можуть легко піддатися відносно простим моделям, якщо останні були натреновані на гігантських даних.

Багато таких задач можуть бути сформульовані у термінах опуклої оптимізації. Враховуючи велику кількість роботи, проведеної людством над методами розкладу і децентралізованими алгоритмами для задач оптимізації, є цілком природнім спроби застосування паралельних алгоритмів оптимізації для розв'язування статистичних задач великого масштабу. Серед переваг такого загального підходу є те, що один алгоритм може достатньо гарно підходити для розв'язування багатьох задач.

Цей огляд присвячений методу множників які змінюють напрямок (ADMM, eng. *alternating direction method of multipliers*), простий але потужний алгоритм який гарно пристосований до розподіленої опуклої оптимізації, і зокрема до задач прикладної статистики та машинного навчання. Він має вигляд процедури *розподілення-координації*, у якій розв'язки малих локальних підзадач координуються для знаходження розв'язку великої глобальної задачі. ADMM можна розглядати як спробу поєднання переваг двоїстого розкладу та доповненого (eng. *augmented*) методу множників Лагранжа для оптимізації з обмеженнями, двох більш ранніх підходів, які ми розглянемо у § 2. Він виявляється еквівалентним до або тісно пов'язаним з багатьма іншими алгоритмами, такими як розбиття Дугласа-Рашфорда з чисельних методів, метод частинних обернених Спінгарна, метод змінних проекцій Дейкстри, ітеративні алгоритми Бергмана для задач з нормою ℓ_1 у обробці сигналів, проксимальні методи, та багатьма іншими. Той факт, що цей алгоритм пере-відкривався у різних областях протягом десятиліть підкреслює інтуїтивні причини його застосування.

Варто зауважити, що сам алгоритм не є новим, і що ми не представляємо жодних нових теоретичних результатів. Він був вперше розглянутий Габаєм, Мерсьєром, Гловінські та Маррокко у середині 1970-их, причому схожі ідеї з'явилися ще у середині 1950-их. Алгоритм детально досліджувався протягом 1980-их, і до середини 1990-их були отримані майже всі теоретичні результати які представлені тут. Той факт, що ADMM був розроблений настільки задовго до появи розподілених обчислювальних систем великого масштабу і відповідних

оптимізаційних задач, відіграє певну роль у тому, що зараз цей алгоритм не настільки широко відомий, як нам здається він має бути відомим.

Основні внески цього огляду можуть бути резюмовані наступним чином:

1. Ми проводимо простий але вичерпний огляд наявних у літературі результатів таким чином, що він підкреслює важливість і поєднує найважливіші деталі реалізації на практиці.
2. На великій кількості прикладів ми показуємо, що алгоритм гарно підходить до широкого кола сучасних розподілених задач великого масштабу. Ми виводимо метод розкладу широкого класу статистичних задач як за тренувальними об'єктами, так і за їхніми ознаками, чого взагалі кажучи не просто досягнути.
3. Ми підкреслюємо практичну важливість реалізації більше ніж усі попередні роботи, що мали радше теоретичний характер.

Протягом усього огляду фокус знаходиться радше на практичних застосуваннях, аніж на теорії, і головною метод є надати читачеві свого роду "мішком інструментів", які можна буде застосувати у багатьох ситуаціях для виведення та реалізації розподіленого алгоритму оптимізації який потім можна буде успішно застосовувати на практиці. Хоча у цьому огляді фокус і знаходиться на паралельних алгоритмах, ADMM можна застосовувати і послідовно, причому для деяких задач він навіть у такій формі буде спроможним конкурувати з найкращими розробленими на сьогодні алгоритмами.

Хоча ми і підкреслюємо тільки ті застосування, які можна просто і вичерпно пояснити, але алгоритм також буде природнім вибором для більш складних задач у таких областях як графічні моделі. На додачу, не зважаючи на те що наш фокус знаходиться на задачах статистичного навчання, ADMM також може бути застосованим у інших випадках, наприклад для інженерного дизайну, аналізу часових рядів, мережевих потоків, або навіть для складання розкладів.

План

Ми починаємо у § 2 з короткого рев'ю двоїстого розкладу і методу множників Лагранжа, двох важливих "попередників" ADMM. Цей параграф включається для послідовності викладу і може бути пропущеним досвідченим читачем без втрати розуміння. У § 3 ми презентуємо ADMM, включаючи основну теорему про збіжність та її доведення, кілька корисних на практиці варіацій основної версії алгоритму, а також огляд основної літератури.

У § 4, ми описуємо кілька загальних шаблонів що часто виникають на практиці, зокрема випадки у яких один з кроків ADMM може бути виконаний особливо ефективно. Ці загальні шаблони будуть зустрічатися у більшості наших прикладів. У § 5 ми розглядаємо використання ADMM для загальної опуклої оптимізації, такої як мінімізація функціоналу з обмеженнями, лінійне та квадратичне програмування. У § 6 ми обговорюємо широке коло задач з нормою ℓ_1. Виявляється, що ADMM приводить до тих методів розв'язування цих задач, які пов'язані з (state-of-the-art) алгоритмами. Цей параграф також пояснює, чому ADMM є особливо гарно пристосованим до задач машинного навчання.

2 Попередники ADMM

У цьому параграфі ми стисло пройдемося по двом алгоритмам оптимізації які є попередниками методу множників які змінюють напрямок. Цей матеріал не використовується у подальшому, але складає підґрунтя до наступних розділів і вмотивовує спосіб їх викладення.

2.1 Метод двоїстого сходження

Розглянемо задачу опуклої оптимізації з обмеженням типу рівність:

$$f(x) \xrightarrow[Ax=b]{} \min, \tag{2.1}$$

де змінна $x \in \mathbb{R}^n$, (стала) матриця $A \in \mathbb{R}^{m \times n}$, (сталий) вектор $b \in \mathbb{R}^m$, і функція $f : \mathbb{R}^n \to \mathbb{R}$ є опуклою.

Функцією Лагранжа задачі (2.1) є

$$L(x, y) = f(x) + y^T \cdot (Ax - b), \tag{2.2}$$

а двоїстою функцією

$$g(y) = \inf_x L(x, y) = -f^* \left(-A^T y\right) - b^T \cdot y, \tag{2.3}$$

де y є двоїстою змінною, або множником Лагранжа, а f^* позначає опукле спряження функції f; див. [2, §3.3] або [3, §12] для визначення. Двоїста задача має вигляд

$$g(y) \to \max, \tag{2.4}$$

де змінна $y \in \mathbb{R}^m$. У припущенні строгої двоїстості, оптимальні значення прямої і двоїстої задач однакові. 0

Ми можемо відновити оптимальну точку x^* прямої задачі з двоїстої оптимальної точки y^* наступним чином:

$$x^* = \operatorname*{argmin}_x L(x, y^*), \tag{2.5}$$

якщо існує тільки одне[1] значення x яке мінімізує $L(x, y^*)$.

Визначення 2.1. У подальшому ми будемо використовувати запис $\operatorname{argmin}_x F(x)$ на позначення *довільного* значення x що мінімізує $F(x)$, навіть якщо воно не єдине.

У методі двоїстого сходження, ми розв'язуємо двоїсту задачу використовуючи градієнтне сходження.

> **Зауваження 2.2** — У припущення диференційовності g, градієнт $\nabla g(y)$ може бути знайденим наступним чином. Спершу ми знаходимо $x^+ = \operatorname{argmin}_x L(x, y)$; тоді маємо $\nabla g(y) = Ax^+ - b$, тобто нев'язка умови рівності.

[1] Ця остання умова часто виконується, зокрема її виконання забезпечує строга опуклість f.

Таким чином, метод двоїстого сходження складається з ітеративних оновлень:

$$x^{k+1} := \operatorname*{argmin}_x L\left(x, y^k\right), \tag{2.6}$$

$$y^{k+1} := y^k + \alpha^k \cdot \left(Ax^{k+1} - b\right), \tag{2.7}$$

де $\alpha^k > 0$ — розмір кроку, а верхній індекс позначає номер ітерації. Перший крок (2.6) є кроком мінімізації по змінній x, а другий крок (2.7) є оновленням двоїстої змінної.

Визначення 2.3. Двоїсту змінну y можна інтерпретувати як вектор цін нев'язок по компонентам, тому крок її оновлення ще називаються *оновленням цін*, або *підбором цін*.

Цей алгоритм називається двоїстим сходження бо при правильному виборі α^k двоїста функція зростає на кожному кроці, тобто $g(y^{k+1}) > g(y^k)$.

Метод двоїстого сходження може бути застосованим навіть у деяких випадках[2] коли g не є диференційовною. Тоді нев'язка $Ax^k - b$ буде не градієнтом g, але мінус *субградієнтом* $-g$. Цей випадок потребує іншого вибору α^k і, взагалі кажучи, не гарантує монотонну збіжність, тобто зачасту $g(y^{k+1}) \not> g(y^k)$.

Якщо ж обирати α^k правильним чином, і виконуються ще кілька інших припущень, то x^k збігається до оптимальної точки, а y^k збігається до оптимальної точки двоїстої задачі. Однак, ці припущення часто не виконуються у багатьох застосування, тому двоїсте сходження не може бути застосованим у таких випадках.

Приклад 2.4

Якщо f є відмінною від нуля афінною функцією довільної компоненти x то крок (2.6) виконати неможливо, оскільки L є необмеженою знизу по x для більшості y.

2.2 Метод двоїстої декомпозиції

Однією з головних переваг метод двоїстого сходження є те, що у деяких випадках він призводить до децентралізованого алгоритму.

Припущення 2.5. Припустимо, для прикладу, що цільова функція f є *розділимою* (відносно розбиття змінної на під-вектори), тобто що

$$f(x) = \sum_{i=1}^{N} f_i(x_i), \tag{2.8}$$

де $x = (x_1, \ldots, x_N)$ і змінні $x_i \in \mathbb{R}^{n_i}$ є під-векторами x.

Розбиваючи A відповідним чином:

$$A = [A_1 \cdots A_N], \tag{2.9}$$

отримаємо $Ax = \sum_{i=1}^{N} A_i x_i$, а тому функція Лагранжа може бути записана у вигляді

[2] У цьому випадку алгоритм зазвичай називають двоїстим субградієнтним методом [4].

$$L(x, y) = \sum_{i=1}^{N} L_i(x_i, y) = \sum_{i=1}^{N} \left(f_i(x_i) + y^T A_i x_i - (1/N) y^T b\right), \tag{2.10}$$

тобто також є розділимою по x. Це означає, що крок (2.6) мінімізації по x розбивається на N незалежних задач які можуть бути розв'язані паралельно. А саме, алгоритм набуває вигляд

$$x_i^{k+1} := \underset{x_i}{\operatorname{argmin}}\, L_i\left(x_i, y^k\right), \tag{2.11}$$

$$y^{k+1} := y^k + \alpha^k \cdot \left(Ax^{k+1} - b\right), \tag{2.12}$$

Крок (2.11) мінімізації по x виконуються незалежно, паралельно для всіх $i = 1, \ldots, N$.

Визначення 2.6. У цьому випадку ми називаємо метод двоїстого сходження методом *двоїстої декомпозиції*.

У загальному випадку кожна ітерація методу двоїстої декомпозиції вимагає виконання операцій *трансляції* та *збору*. У кроці (2.12) оновлення двоїсто змінної для обчислення нев'язки збираються "внески" $a_i x_i^{k+1}$ всіх під-векторів. Після оновлення двоїстої змінної її нове значення транслюється до N процесів для виконання N незалежних кроків (2.11) мінімізації по x_i.

Двоїста декомпозиція є давньою "ідеологією" в оптимізації, і її використання відстежується принаймні до ранніх 1960-их. Схожі ідеї зустрічаються у відомих роботах Данціга і Вольфа [5] та [6] щодо великомасштабних задач лінійного програмування, а також у (seminal) книзі Данціга [7]. Схоже, що загальна ідея двоїстої декомпозиції належить Еверету [8], але досліджується у і багатьох більш ранніх роботах [9–12]. Використання недиференційовної оптимізації, таке як субградієнтний метод, для розв'язання двоїстої задачу обговорюється Шором у [4]. Гарні матеріали стосовно двоїстих методів і декомпозиції включаються також книгу Берцекаса [13, глава 6] і огляд Недіча і Оздаглара [14] розподіленої оптимізації, який обговорює двоїсту декомпозицію і задачі (consensus). Багато статей також досліджують варіації стандартної двоїстої декомпозиції, як-то [15].

Взагалі кажучи, децентралізована оптимізація була активною галуззю для досліджень з початку 1980-их. Наприклад, Цицикліс та його співавтори працювали над багатьма децентралізованими задачами (detection and consensus) що включали мінімізацію гладкої функції f відомої багатьом агентам [16–18]. Деякі гарні матеріали щодо паралельної оптимізації включають роботи [18] Берцекаса і Цицикліса, та [19] Цензора і Зеніоса. Нещодавно також проводилася робота над задачами, де у кожного агента є своя власна опукла (але можливо не диференційовна) цільова функція [20]. У [21] можна знайти свіже обговорення розподілених методів для (graph-structured) задач оптимізації.

2.3 Доповнена функція Лагранжа і метод множників

Методи що використовують доповнені функції Лагранжа були розроблені щоб привнести стійкість у метод двоїстого сходження і гарантувати збіжність без таких обмежень як строга опуклість чи скінченність f.

Визначення 2.7. *Доповненою функцією Лагранжа* задачі (2.1) називається

$$L_\rho(x, y) = f(x) + y^\intercal \cdot (Ax - b) + (\rho/2) \cdot \|Ax - b\|_2^2, \tag{2.13}$$

де $\rho > 0$ називається[3] *штрафним параметром.*

Доповнену функцію Лагранжа можна розглядати як стандартну функцію Лагранжа задачі

$$f(x) + (\rho/2) \cdot \|Ax - b\|_2^2 \xrightarrow[Ax=b]{} \min. \tag{2.14}$$

Ця задача, очевидно, еквівалентні початковій задачі (2.1), оскільки для довільного допустимого x до цільової функції додається доданок що дорівнює нулеві. Відповідною доповненою двоїстою функцією є $g_\rho = \inf_x L_\rho(x, y)$.

Перевагою додавання штрафного доданку є те, що g_ρ стає диференційовною за більш слабких умов на оригінальну задачу. Градієнт доповненої двоїстої функції знаходиться так само як для звичайної функції Лагранжа.

Застосування двоїстого сходження до модифікованої задачі приводить до алгоритму

$$x^{k+1} \coloneqq \operatorname*{argmin}_x L_\rho\left(x, y^k\right), \tag{2.15}$$

$$y^{k+1} \coloneqq y^k + \rho \cdot \left(Ax^{k+1} - b\right), \tag{2.16}$$

також відомому як *метод множників* для задачі (2.1).

Це все той же метод двоїстого сходження, хіба що у кроці мінімізації по x використана доповнена функція Лагранжа, а розмір кроку α^k взятий штрафним параметром ρ.

Зауваження 2.8 — Метод множників збігається за набагато більш загальних умов ніж метод двоїстого сходження, включаючи випадки коли f набуває значення $+\infty$ або не є строго опуклою.

Твердження 2.9

Вибір розміру кроку у (2.16) рівним ρ легко поясними.

Припущення 2.10. Для простоти припустимо, що f диференційовна (хоча цього і не вимагається для роботи алгоритму).

Доведення. Тоді умовами оптимальності для задачі (2.1) є пряма і двоїста допустимість, тобто

$$Ax^* - b = 0, \quad \nabla f(x^*) + A^\intercal y^* = 0, \tag{2.17}$$

відповідно.

[3] Зауважимо, що L_0 є стандартною функцією Лагранжа цієї задачі.

За визначенням, x^{k+1} мінімізує $L_\rho(x, y^k)$, тому

$$0 = \nabla_x L_\rho \left(x^{k+1}, y^k\right) = \tag{2.18}$$

$$= \nabla_x f\left(x^{k+1}\right) + A^\intercal \cdot \left(y^k + \rho \cdot \left(Ax^{k+1} - b\right)\right) = \tag{2.19}$$

$$= \nabla_x f\left(x^{k+1}\right) + A^\intercal \cdot y^{k+1}. \tag{2.20}$$

Як бачимо, при використанні ρ у якості розміру кроку у оновленні двоїстої змінної, ітерації $\left(x^{k+1}, y^{k+1}\right)$ є двоїсто допустимою. □

Тому, коли при ітераціях методу множників нев'язка $Ax^{k+1} - b$ прямує до нуля, то цього достатньо для оптимальності.

Значно покращена збіжність методу множників приходить не безкоштовно. А саме, коли f є розділимою, доповнена функція Лагранжа вже не є розділимою, тому крок (2.15) мінімізації по x не може бути виконаний паралельно[4] для кожного x_i.

Доповнені функції Лагранжа і метод множників для оптимізації з обмеженнями були вперше запропоновані у пізніх 1960-их Хестенесом у [22, 23] і Пауеллом [24]. Багато з перших чисельних експериментів над методом множників належать Міле і ко. [25–27]. Більшість ранніх праць зібрані у монографії Берцекаса [28], який також обговорює схожість з більш давніми методами що використовували функції Лагранжа і штрафні функції [29–31], а також багато узагальнень.

[4] Це означає, що базовий варіант методу множників не може бути використаний для декомпозиції. Ми розглянемо цю проблему далі.

3 Метод множників що змінюють напрямок

3.1 Алгоритм

ADMM є алгоритмом, що покликаний поєднати можливість декомпозиції у методі двоїстого сходження з покращеною збіжністю методу множників. Алгоритм застосовується до задач у вигляді

$$f(x) + g(z) \xrightarrow[Ax+Bz=c]{} \min. \tag{3.1}$$

де змінні $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, (сталі) матриці $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{p\times m}$, (сталий) вектор $c \in \mathbb{R}^p$. Поки що ми припустимо, що f і g опуклі, а більш конкретні припущення будуть розглянуті в §3.2. Єдиною відмінністю від загальної задачі (2.1) з лінійним обмеженням типу рівність є те, що змінна x була розділена на дві частини, x і z, причому цільова функція є розділимою відносно такого поділу. Позначатимемо оптимальне значення задачі (3.1) так:

$$p^* = \inf\{f(x) + g(z) | Ax + Bz = c\}. \tag{3.2}$$

Як і у методі множників, розглянемо розширену функцію Лагранжа:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T \cdot (Ax + Bz - c) + (\rho/2) \cdot \|Ax + Bz - c\|_2^2. \tag{3.3}$$

ADMM складаєть з ітерацій

$$x^{k+1} := \operatorname*{argmin}_x L_\rho \left(x, z^k, y^k\right), \tag{3.4}$$

$$z^{k+1} := \operatorname*{argmin}_z L_\rho \left(x^{k+1}, z, y^k\right), \tag{3.5}$$

$$y^{k+1} := y^k + \rho \cdot \left(Ax^{k+1} + Bz^{k+1} - c\right), \tag{3.6}$$

де $\rho > 0$. Алгоритм дуже схожий на двоїсте сходження і на метод множників: він складається з кроку (3.4) мінімізації по x, кроку (3.5) мінімізації по z, і кроку (3.6) оновлення двоїстої змінної. Як і у методі множників, при оновленні двоїстої змінної у якості розміру кроку використовується штрафний параметр ρ.

Метод множників для задачі (3.1) мав форму

$$\left(x^{k+1}, z^{k+1}\right) := \operatorname*{argmin}_{x,z} L_\rho \left(x, z, y^k\right), \tag{3.7}$$

$$y^{k+1} := y^k + \rho \cdot \left(Ax^{k+1} + Bz^{k+1} - c\right). \tag{3.8}$$

Тут розширена функція Лагранжа мінімізувалася по x і по z водночас. Натомість у ADMM змінні x та z оновлюються по черзі (eng. *in the alternating fashion*), звідки і походить назва алгоритму. ADMM можна розглядати як звичайний метод множників де замість одночасної мінімізації використовується один прохід Гауса-Зейделя по x і z [32, §10.1]. Виявляється, що розділення мінімізації по x і z на два кроки це саме те, що дозволяє застосувати декомпозицію у випадку коли f або g розділимі.

Зауваження 3.1 — "Стан" алгоритму складається тільки з z^k і y^k, тобто (z^{k+1}, y^{k+1}) є функцією від (z^k, y^k), а змінна x^k є радше проміжним результатом обчислень.

Зрозуміло, що якщо поміняти місцями (пере-назвати) x і z, f і g, A і B у задачі (3.1), то ми отримаємо той же алгоритм з інвертованим порядком кроків (3.4) і (3.5). Тут ролі x і z майже симетричні, єдине що при інвертованому порядку вийде нібито крок оновлення двоїстої змінної y знаходиться між кроками оновлення x та z.

3.1.1 Масштабована форма

ADMM можна записати трохи інакше, у вигляді який зачасту є більш зручним. Для цього поєднаємо лінійний і квадратичний доданок у розширеній функції Лагранжа і масштабуємо двоїсту змінну. Якщо позначити нев'язку $r = Ax + Bz - c$, то будемо мати

$$y^T \cdot r + (\rho/2) \cdot \|r\|_2^2 = (\rho/2) \cdot \|r + (1/\rho) \cdot y\|_2^2 - (1/2\rho) \cdot \|y\|_2^2 = \tag{3.9}$$

$$= (\rho/2) \cdot \|r + u\|_2^2 - (\rho/2) \cdot \|u\|_2^2, \tag{3.10}$$

де $u = (1/\rho) \cdot y$ називається *масштабованою двоїстою змінною*. За допомогою масштабованої двоїстої змінної ADMM можна записати у вигляді

$$x^{k+1} := \operatorname*{argmin}_x \left(f(x) + (\rho/2) \cdot \left\| Ax + Bz^k - c + u^k \right\|_2^2 \right), \tag{3.11}$$

$$z^{k+1} := \operatorname*{argmin}_z \left(g(z) + (\rho/2) \cdot \left\| Ax^{k+1} + Bz - c + u^k \right\|_2^2 \right), \tag{3.12}$$

$$u^{k+1} := u^k + Ax^{k+1} + Bz^{k+1} - c. \tag{3.13}$$

Якщо позначити нев'язку на ітерації k як $r^k = Ax^k + Bz^k - c$, то побачимо, що

$$u^k = u^0 + \sum_{j=1}^{k} r^j, \tag{3.14}$$

тобто є сумою нев'язок.

Визначення 3.2. Першу форму ADMM яка складається з ітерацій (3.4)–(3.6) будемо називати *немасштабованою* формою, а другу форму яка складається з ітерацій (3.11)–(3.13) — *масштабованою*, оскільки вона використовує масштабовану двоїсту змінну.

Ці форми, очевидно, еквівалентні, але формули у масштабованій формі задачі коротші ніж в немасштабованій, тому ми будемо використовувати саме масштабовану форму надалі. Немасштабована форма застосовуватиметься тільки коли нам знадобиться інтерпретація зав'язана на немасштабовану двоїсту змінну.

3.2 Збіжність

Є багато результатів стосовно збіжності ADMM; тут ми обмежимося базовим, але все ще дуже загальним результатом, який застосовний до усіх прикладів які ми розглянемо. Зробимо одне припущення щодо функцій f і g і ще одне припущення щодо задачі (3.1).

Припущення 3.3. Функції $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ і $g : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ замкнуті, правильні, та опуклі.

Це припущення можна компактно записати в термінах над-графіків: функція f задовольняє припущення тоді і тільки тоді, коли її над-графік

$$\operatorname{epi} f = \{(x, t) \in \mathbb{R}^n \times \mathbb{R} | f(x) \leq t\} \tag{3.15}$$

є замкнутою непорожньою опуклою множиною.

Зауваження 3.4 — Це припущення означає, що підзадачі які виникають у кроках (3.4) і (3.5) *можна розв'язати*, тобто існують x і z (не обов'язково єдині без подальших припущень щодо A і B) які мінімізують розширену функцію Лагранжа.

Зауваження 3.5 — Важливо, що це припущення не вимагає від f і g диференційовності чи скінченності.

Приклад 3.6

Зокрема, f може бути індикатором непорожньої опуклої множини $\mathcal{C}$, тобто $f(x) = 0$ для $x \in \mathcal{C}$ і $f(x) = +\infty$ інакше. У цьому випадку крок (3.4) мінімізації по x буде задачею квадратичного програмування над $\mathcal{C}$.

Припущення 3.7. У не розширеної функції Лагранжа L_0 є сідлова точка.

Тобто, існують (x^*, z^*, y^*), не обов'язково єдині, такі, що

$$L_0(x^*, z^*, y) \leq L_0(x^*, z^*, y^*) \leq L_0(x, z, y^*) \tag{3.16}$$

виконується для усіх x, z, y.

З першого припущення випливає, що $L_0(x^*, z^*, y^*)$ є скінченним для довільної сідлової точки (x^*, z^*, y^*). З цього у свою чергу випливає, що (x^*, z^*) є розв'язком (3.1), тобто що $Ax^* + Bz^* = c$ і $f(x^*), g(z^*) < \infty$. Звідси також випливає, що y^* є двоїстою оптимальною точкою, і що оптимальні значення прямої і двоїстої задач однакові, тобто що виконується строга двоїстість.

Зауваження 3.8 — Ми не робили жодних більш конкретних припущень щодо A, B чи c. Зокрема, ані A ані B не зобов'язані бути невиродженими.

3.2.1 Збіжність

Теорема 3.9

При виконанні припущень 1 і 2, ітерації ADMM задовольняють наступне:

- *Збіжність нев'язки.* $r^k \to 0$ при $k \to \infty$, тобто ітерації збігаються до допустимої множини.
- *Збіжність цільової функції.* $f\left(x^k\right) + g\left(z^k\right) \to p^*$ при $k \to \infty$, тобто цільова функція збігається до оптимального значення.
- *Збіжність двоїстої змінної.* $y^k \to y$ при $k \to \infty$, де y^* є двоїстою оптимальною точкою.

Доведення перших двох пунктів подані у додатку.

Зауваження 3.10 — x^k і z^k не зобов'язані збігатися до оптимальних значень, хоча це можна забезпечити шляхом подальших припущень.

3.2.2 Збіжність на практиці

Прості приклади показують, що ADMM може дуже повільно збігатися до високої точності. Щоправда, здебільшого ADMM дуже швидко сходиться (за кілька десятків ітерацій) до пристойної точності, а це саме те, що потрібно у більшості застосувань. Така поведінка робить ADMM схожим на алгоритми на кшталт двоїстого градієнтного методу у тому сенсі, що вони швидко видають пристойний результат який можна застосовувати на практиці. Щоправда повільна збіжність ADMM відрізняє його від методу Ньютона (або, що задач з обмеженнями, методів внутрішньої точки), де навіть висока точність може бути досягнута за розумний час. у деяких випадках можливо поєднати ADMM з одним із таких методів синтезу високоточного розв'язку з менш точного розв'язку [33], але здебільшого ADMM застосовується у галузях де пристойної точності цілком достатньо. На щастя, саме так воно і є для задач великого масштабу які ми розглядаємо. Ба більше, у задачах статистики або машинного навчання, надточне знаходження параметрів здебільшого тягне за собою або незначне покращення якості прогнозів (головної метрики якості), або взагалі жодного покращення.

3.3 Умови оптимальності і критерій зупинки

Лема 3.11

Необхідними і достатніми умовами оптимальності є пряма допустимість

$$Ax^* + Bz^* - c = 0, \tag{3.17}$$

і двоїста допустимість,

$$0 \in \partial f\left(x^*\right) + A^T \cdot y^*, \tag{3.18}$$

$$0 \in \partial g\left(z^*\right) + B^T \cdot y^*. \tag{3.19}$$

Зауваження 3.12 — Тут ∂ позначає оператор субдиференціювання, див. [3, 34, 35] для визначення.

Зауваження 3.13 — Коли f і g диференційовні, субдиференціали ∂f і ∂g можна замінити на градієнти ∇f і ∇g, а $\in$ — на $=$.

Доведення. Оскільки z^{k+1} мінімізує $L_\rho\left(x^{k+1}, z, y^k\right)$ за визначенням, то ми маємо

$$0 \in \partial g\left(z^{k+1}\right) + B^\intercal \cdot y^k + \rho \cdot B^\intercal \cdot \left(Ax^{k+1} + Bz^{k+1} - c\right) = \tag{3.20}$$

$$= \partial g\left(z^{k+1}\right) + B^\intercal \cdot y^k + \rho \cdot B^\intercal \cdot r^{k+1} = \tag{3.21}$$

$$= \partial g\left(z^{k+1}\right) + B^\intercal \cdot y^{k+1}. \tag{3.22}$$

Це означає, що z^{k+1} і y^{k+1} завжди задовольняють (3.19), тобто для оптимальності залишається забезпечити (3.17) і (3.18). Цей феномен аналогічний до того, що ітерації методу множників завжди є двоїсто допустимими, див. § 2.

Оскільки x^{k+1} мінімізує $L_\rho\left(x, z^k, y^k\right)$ за визначенням, то ми маємо

$$0 \in \partial f\left(x^{k+1}\right) + A^\intercal \cdot y^k + \rho \cdot A^\intercal \cdot \left(Ax^{k+1} + Bz^k - c\right) = \tag{3.23}$$

$$= \partial f\left(x^{k+1}\right) + A^\intercal \cdot \left(y^k + \rho \cdot r^{k+1} + \rho \cdot B \cdot \left(z^k - z^{k+1}\right)\right) = \tag{3.24}$$

$$= \partial f\left(x^{k+1}\right) + A^\intercal \cdot y^{k+1} + \rho \cdot A^\intercal \cdot B \cdot \left(z^k - z^{k+1}\right), \tag{3.25}$$

або, що те саме,

$$\rho \cdot A^\intercal \cdot B \cdot \left(z^{k+1} - z^k\right) \in \partial f\left(x^{k+1}\right) + A^\intercal \cdot y^{k+1}. \tag{3.26}$$

Це у свою чергу означає, що

$$s^{k+1} = \rho \cdot A^\intercal \cdot B \cdot \left(z^{k+1} - z^k\right) \tag{3.27}$$

можна розглядати як нев'язку до умови двоїстої допустимості (3.18). □

Визначення 3.14. Будемо називати s^{k+1} *двоїстою нев'язкою* на ітерації $k+1$, а $r^{k+1} = Ax^{k+1} + Bz^{k+1} - c$ — *прямою нев'язкою'* на ітерації $k+1$.

Резюмуючи, умови оптимальності для ADMM складаються з трьох умов (3.17)-(3.19). Остання умова (3.19) завжди виконується для трійки $\left(x^{k+1}, z^{k+1}, y^{k+1}\right)$; нев'язки інших двох, (3.17) та (3.18) є прямою та двоїстою нев'язками r^{k+1} та s^{k+1} відповідно. Ці дві нев'язки збігаються[5] до нуля при ітераціях ADMM.

[5] Насправді, доведення у додатку показує, що $B \cdot \left(z^{k+1} - z^k\right)$ збігається до нуля, звідки випливає що s^k збігається до нуля.

3.3.1 Критерій зупинки

Можна показати, що нев'язки умов оптимальності пов'язані з оцінкою на нестачу до оптимальності цільової функції, тобто на $f\left(x^k\right) + g\left(z^k\right) - p^*$. Як показано у додатку, виконується наступна нерівність:

$$f\left(x^k\right) + g\left(z^k\right) - p^* \leq -\left(y^k\right)^T \cdot r^k + \left(x^k - x^*\right)^T \cdot s^k. \tag{3.28}$$

Звідси випливає, що коли нев'язки r^k і s^k достатньо малі, то нестача до оптимальності також мала.

Щоправда, ми не можемо користуватися безпосередньо цією нерівністю як критерієм зупинки, адже ми не знаємо x^*. Втім, якщо ми тим чи іншим чином знаємо, що $\left\|x^k - x^*\right\|_2 \leq d$, то можна записати

$$f\left(x^k\right) + g\left(z^k\right) - p^* \leq -\left(y^k\right)^T \cdot r^k + d \cdot \left\|s^k\right\|_2 \leq \left\|y^k\right\|_2 \cdot \left\|r^k\right\|_2 + d \cdot \left\|s^k\right\|_2. \tag{3.29}$$

Праву (або середню) частину цієї нерівності можна використовувати як оцінку на нестачу до оптимальності.

Звідси випливає наступний логічний критерій зупинки: пряма і двоїста нев'язки мають бути малі, а саме

$$\left\|r^k\right\|_2 \leq \varepsilon^{\text{pri}} \quad \text{та} \quad \left\|s^k\right\|_2 \leq \varepsilon^{\text{dual}}, \tag{3.30}$$

де $\varepsilon^{\text{pri}} > 0$ і $\varepsilon^{\text{dual}} > 0$ є так званими толерантностям до умов прямої і двоїстої (не) допустимості, (3.17) і (3.18) відповідно. Ці толерантності можна вибрати за допомогою абсолютних або відносних критеріїв, наприклад таких:

$$\varepsilon^{\text{pri}} = \sqrt{p} \cdot \varepsilon^{\text{abs}} + \varepsilon^{\text{rel}} \cdot \max\left\{\left\|Ax^k\right\|_2, \left\|Bz^k\right\|_2, \|c\|_2\right\}, \tag{3.31}$$

$$\varepsilon^{\text{dual}} = \sqrt{n} \cdot \varepsilon^{\text{abs}} + \varepsilon^{\text{rel}} \cdot \left\|A^T \cdot y^k\right\|_2, \tag{3.32}$$

де $\varepsilon^{\text{abs}} > 0$ є абсолютною[6] толерантністю, а $\varepsilon^{\text{rel}} > 0$ — відносною. Розумним вибором відносної толерантності є $\varepsilon^{\text{rel}} = 10^{-3}$, або 10^{-4}, залежно від задачі. Вибір абсолютної толерантності сильно залежить від масштабу компонент змінних що розглядаються.

3.4 Узагальнення і варіації

Багато варіацій класичного алгоритму ADMM були досліджені протягом десятиліть. Тут ми коротко оглянемо деякі з цих варіацій згруповані за ідеями. Деяких з цих варіацій мають кращу збіжність на практиці ніж класичний ADMM. Більшість узагальнень дуже детально досліджені і для них доведені аналогічні результати стосовно збіжності.

[6] Множники $\sqrt{p}$ і $\sqrt{n}$ додаються бо ℓ_2-норми беруть в різних просторах, $\mathbb{R}^p$ і $\mathbb{R}^n$ відповідно.

3.4.1 Змінний штрафний параметр

Стандартним узагальненням є можливість змінювати штрафний параметр ρ^k на кожній ітерацій, аби покращити збіжність на практиці, а також зменшити вплив початкового значення параметру на перебіг алгоритму. У контексті методу множників цій підхід проаналізований в [36], де показується, що над-лінійна збіжність може бути досягнута якщо $\rho^k \to \infty$.

Взагалі кажучи, складно довести збіжність ADMM якщо ρ може змінюватися на кожній ітерації, але вищезгадані теоретичні результати, очевидно, залишаються в силі, якщо припустити що ρ стає сталим після певної скінченної кількості ітерацій.

Простою схемою яка часто працює добре (див., наприклад [37, 38]) є:

$$\rho^{k+1} := \begin{cases} \tau^{\text{incr}} \cdot \rho^k, & \left\|r^k\right\|_2 \geq \mu \cdot \left\|s^k\right\|_2, \\ \rho^k / \tau^{\text{decr}}, & \left\|s^k\right\|_2 \geq \mu \cdot \left\|r^k\right\|_2, \\ \rho^k, & \text{інакше}, \end{cases} \tag{3.33}$$

де $\mu > 1$, $\tau^{\text{incr}} > 1$, $\tau^{\text{decr}} > 1$ — параметри. Типовим вибором може бути $\mu = 10$, $\tau^{\text{incr}} = \tau^{\text{decr}} = 2$. За такою зміною штрафного параметру ховається ідея "намагатися втримати пряму і двоїсту нев'язку близькими одна до одної по мірі того як обидві збігаються до нуля".

Справді, ітераційні рівняння ADMM показують, що великі значення ρ збільшують штраф за пряму нев'язку, тому призводять до малих прямих нев'язок. Натомість визначення s^{k+1} показує, що малі значення ρ проводять до малих двоїстих нев'язок (але більших прямих нев'язок).

Схема (3.33) збільшує ρ у τ^{incr} разів коли пряма нев'язка стає великою у порівняння з двоїстою, і зменшує у τ^{decr} разів коли відбувається зворотне. Цю схему можна модифікувати із використанням ε^{pri} та $\varepsilon^{\text{dual}}$:

$$\rho^{k+1} := \begin{cases} \tau^{\text{incr}} \cdot \rho^k, & \varepsilon^{\text{pri}} \geq \mu \cdot \varepsilon^{\text{dual}}, \\ \rho^k / \tau^{\text{decr}}, & \varepsilon^{\text{dual}} \geq \mu \cdot \varepsilon^{\text{pri}}, \\ \rho^k, & \text{інакше}, \end{cases} \tag{3.34}$$

Зауваження 3.15 — Якщо ми використовуємо змінний штрафний параметр то у масштабованій формі ADMM потрібно відповідним чином змінювати масштабовану двоїсту змінну $u^k = (1/\rho)y^k$.

Приклад 3.16

Якщо ρ зменшилося вдвічі, то u^k потрібно збільшити удвічі перед тим як переходити до наступної ітерації.

3.4.2 Штрафні доданки загального вигляду

Іншою ідеєю є заміна квадратичного доданку $(\rho/2) \cdot \|r\|_2^2$ на $(1/2) \cdot r^T P r$, де P — симетрична додатно визначена матриця. Коли P є сталою то це узагальнення ADMM можна розглядати

як класичний ADMM застосований до модифікованої початкової задачі у якій умову $r = 0$ замінили на $Fr = 0$, де $F^TF = P$.

3.4.3 Над-релаксація

У кроках оновлення змінних z та y, величину Ax^{k+1} можна замінити величиною

$$\alpha^k Ax^{k+1} - (1 - \alpha^k) \cdot (Bz^k - c), \tag{3.35}$$

де $\alpha^k \in (0, 2)$ є *параметром релаксації*.

Визначення 3.17. Коли $\alpha^k > 1$ ця техніка називається *над-релаксацією*, а коли $\alpha^k < 1$ то недо-релаксація.

Ця схема проаналізована в [39], і експерименти в [33, 40] дають підстави вважати, що над-релаксація з $\alpha^k \in [1.5, 1.8]$ може покращувати збіжність.

3.4.4 Неточна мінімізація

ADMM буде збігатися навіть якщо кроки мінімізації по x і по z виконуються не точно а наближено, за умови що нестачі до оптимальності по всім ітераціям будуть задовольняти певним умовам, наприклад якщо ряд з них буде збіжним.

Цей результат встановлений Екштейном і Берцекасом в [39], на основі попередніх результатів Гольштейна і Третьякова [41]. Ця техніка важлива у ситуаціях коли кроки мінімізації по x і по z виконуються за допомогою ітеративних алгоритмів; вона дозволяє розв'язувати такі задачі мінімізації з низькою точністю на перших ітераціях і поступово її підвищувати.

3.4.5 Порядок кроків

Декілька варіацій ADMM передбачають зміну порядку кроків оновлення змінних x, y і z, або виконання одного з кроків декілька разів за ітерацію.

Приклад 3.18

Див. [42]: якщо ми розділимо змінні x і z на k блоків, кожен з яких будемо оновлювати по черзі, перед тим як оновити змінну y, то отриманий алгоритм можна інтерпретувати як декілька проходів Гауса-Зейделя замість одного.

Приклад 3.19

Якщо ж змінні x та z оновлюються багато разів по черзі перед оновленням y то отримаємо алгоритм, дуже схожий на метод множників (див. [18, §3.4.4] для пояснення чому).

Приклад 3.20

Іншим варіантом є проведення додаткового кроку оновлення змінної y між кроками для x і z, з половинною довжиною крока [18].

3.4.6 Пов'язані алгоритми

Є ще багато інших алгоритмів які відрізняються від ADMM, але які використовують схожі ідеї. Наприклад, Фукушіма [43] застосовує ADMM до постановки двоїсто задачі як до прямої, що призводить до "двоїстого ADMM", який, як показано в [44], є еквівалентним до "прямого методу Дугласа-Рашфорда" у постановці [45, §3.5.6].

Є також алгоритми які поєднують ADMM з проксимальним методом множників [46], радше ніж зі звичайним методом множників; див. також [47, 48]. Інші відповідні публікації включають [40, 49–54].

3.5 Примітки та посилання

Спочатку ADMM був запропонований в середині 1970-х років Гловінські, Маррокко [55] і Габаєм і Мерсьєром [56]. Є ще ряд інших важливих робіт з аналізу властивостей алгоритму, в тому числі [43, 44, 47, 57–61]. Зокрема, збіжність ADMM була досліджена багатьма авторами, включаючи Габая [58], Екштейна і Берцекаса [39].

ADMM також застосовувався до ряду статистичних проблем, такі, як розріджена регресія з обмеженнями [62], розріджене відновлення сигналу [63], відновлення зображень і знешумлення [64–66], мну з регуляризацією нормою сліду [67], оцінка матриці коваріації з розрідженою оберненою [68], селектор Данціга [69] і машина опорних векторів [70]. Приклади обробки сигналів див. [54, 71–74] і посилання в них.

Багато робіт, що аналізують ADMM, роблять це з точки зору *максимальних монотонних операторів* [36, 39, 46, 75, 76]. Широке різноманіття задач можна задати як знаходження нуля максимального монотонного оператор.

Приклад 3.21

Якщо f замкнений, власний і опуклий, то субдиференціальний оператор $\partial \mathsf{f}$ є максимальним монотонним, і знаходження нуля $\partial \mathsf{f}$ — те саме що проста мінімізація f.

Зауваження 3.22 — Така мінімізація може неявно містити обмеження, якщо f дозволено приймати значення $+\infty$.

Метод проксимальної точки Рокфеллера [36] є загальним методом знаходження нуля максимального монотонного оператора, і широкий спектр алгоритмів був розроблений для цієї задачі, включаючи проксимальну мінімізацію (див. §4.1), метод множників і ADMM. Для більш детального огляду старої літератури, див. [45, §2].

Показано, що метод множників є особливим випадком алгоритму проксимальної точки Рокафеллара [46]. Габай [58] показав, що ADMM — це особливий випадок методу, що називається *розбиттям Дугласа-Рашфорда* для монотонних операторів [77, 78], Екштейн та Берцекас [39] показали, що розбиття Дугласа-Ращфорда є особливим[7] випадком алгоритму проксимальної точки.

[7] Натомість варіант ADMM, який виконує додатковий крок оновлення змінної y між кроками оновлення x і z є еквівалентним до *розбиття Пісмана-Рашфорда* [78, 79] цього, як показали Гловінські і Ле Таллек [60].

Використовуючи той же фреймворк, Екштейн і Берцкас [39] також показали зв'язок з деякими іншими алгоритмами, такі як метод часткових обернених Спінгарна [80]. Лоуренс і Спінгарн [81] розробили альтернативний фреймворк який показує, що розщеплення Дугласа-Рашфорда, а отже і ADMM, є особливим випадком алгоритму проксимальної точки; Екштейн і Ферріс [33] пропонують більш свіжий опис, який пояснює цей підхід.

Головне значення цих результатів полягає в тому, що вони дозволяють застосовувати потужну теорію збіжності методу проксимальної точки до ADMM та інших методів, і показують, що багато з цих алгоритмів по суті ідентичні.

Зауваження 3.23 — Наші докази збіжності базового алгоритму ADMM, наведені в додатку A, самостійні і не покладаються на цю абстрактну техніку.

Дослідження методів розбиття операторів та їхній зв'язок із алгоритмами декомпозиції продовжується до сьогодні [82, 83].

Значну кількість останніх досліджень розглядають заміну квадратичного штрафного доданку у стандартному методі множників на більш загальний доданок, наприклад, такий, що отримується з *дивергенції Брегмана* [84, 85]; див. [86] для визначення.

На жаль, ці узагальнення, здається, не переносяться простим чином від не декомпозиційних методів з розширеною функцією Лагранжа на ADMM: На сьогодні не існує доведення збіжності ADMM з не квадратичними штрафними доданками.

4 Загальні патерни

Особливості структури в f, g, A, і B зачасту можна використати аби виконувати кроки оновлення змінних x та z ефективніше. Тут ми розглянемо три загальні випадки які часто зустрічатимуться нам в подальшому: квадратичні цільові доданки, розділимі цільові функції та розділимі обмеження, а також гладкі цільові доданки. Наше обговорення буде описане у термінах кроку оновлення змінної x, але аналогічно переноситься на крок оновлення змінної z. Записуватимемо крок оновлення змінної x як

$$x^+ = \operatorname{argmin}_x \left(f(x) + (\rho/2) \cdot \|Ax - v\|_2^2 \right), \tag{4.1}$$

де $v = -Bz + c - u$ — відомий сталий (на кроці оновлення змінної x) вектор.

4.1 Проксимальний оператор

По-перше, розглянемо найпростіший випадок коли $A = I$, який насправді доволі часто зустрічається у прикладах. Тоді оновлення x набуває вигляду:

$$x^+ = \operatorname{argmin}_x \left(f(x) + (\rho/2) \cdot \|x - v\|_2^2 \right). \tag{4.2}$$

Визначення 4.1. Як функція v права частина цієї рівності позначається $\operatorname{prox}_{f,\rho}(v)$ і називається *проксимальним оператором* функції f зі штрафним доданком ρ [87].

Визначення 4.2. У варіаційному аналізі,

$$\tilde{f}(v) = \inf_x \left(f(x) + (\rho/2) \cdot \|x - v\|_2^2 \right) \tag{4.3}$$

відома як *обгортка Мореа* або *регуляризація Мореа-Йосіди* функції f і є тісно пов'язаною з теорією алгоритму проксимальної точки [76].

Визначення 4.3. Мінімізацію по x у проксимальному операторі зазвичай називають *проксимальною мінімізацією*.

Хоча ці спостереження самі по собі не покращують ефективність ADMM, але вони пов'язують цей крок алгоритму з іншими відомими ідеям.

Коли функція f достатньо проста, то проксимальний оператор (тобто оновлення x) може бути обчисленим аналітично, див. [72] для великої кількості прикладів.

Приклад 4.4

Зокрема, коли f є індикатором замкненої непорожньої опуклої множини $\mathcal{C}$, то крок оновлення змінної x набуває вигляду

$$x^+ = \operatorname{argmin}_x \left(f(x) + (\rho/2) \cdot \|x - v\|_2^2 \right) = \Pi_{\mathcal{C}}(v), \tag{4.4}$$

де $\Pi_{\mathcal{C}}$ позначає проекцію (в Евклідовій нормі) на $\mathcal{C}$.

Зауваження 4.5 — Попереднє твердження справджується не залежно від вибору ρ, тобто автоматично гарно поєднується з варіаціями у яких ρ змінне, як у §3.4.1.

Приклад 4.6

Зокрема, якщо f — індикаторна функція невід'ємного ортанту $\mathbb{R}^n_+$, то ми маємо $x^+ = (v)_+$, тобто вектор який отримується взяттям невід'ємної частини кожної компоненти вектору v.

4.2 Квадратичні штрафні доданки

Припустимо, що f — задана (опукла) квадратична функція

$$f(x) = (1/2) \cdot x^T P x + q^T x + r, \tag{4.5}$$

де $P \in \mathbb{S}^n_+$, тобто симетрична додатно-визначена матриця $n \times n$.

Зауваження 4.7 — Це охоплює випадки коли f — лінійна або константа, за рахунок використання або $P = 0$, або $q = 0$, або і того і того.

Тоді, у припущенні існування оберненої у $P + \rho A^T A$, x^+ є афінною функцією змінної v, а саме

$$x^+ - (P + \rho A^T A)^{-1} \cdot (\rho A^T v - q)\,. \tag{4.6}$$

Іншими словами, оновлення x зводиться до розв'язування СЛАР з додатно-визначеною матрицею коефіцієнтів $P + \rho A^T A$ і правою частиною $\rho A^T v - q$. Як було показано раніше, розумне використання лінійної алгебри може експлуатувати цей факт і суттєво покращити роботу. Для загального бекграунду з чисельної лінійної алгебри див. [88] або [32]. Див. [2, додаток C] для короткого огляду аналітичних методів.

4.2.1 Аналітичні методи

Поки що розглянемо використання аналітичних методів для розв'язування вищезгаданої системи лінійних алгебраїчних рівнянь, ітеративні методи будуть оглянуті в §4.3. Аналітичні методи розв'язування СЛАР $Fx = g$ базуються на розкладі (*факторизації*) матриці F у добуток $F_1 \cdot F_2 \cdot \ldots \cdot F_k$ простіших матриць, а потім обчислення $x = F^{-1} b$ шляхом розв'язання низки задач вигляду $F_i z_i = z_{i-1}$, де $z_1 = F_1^{-1} g$ і $x = z_k$.

Визначення 4.8. Процес розв'язування простіших задач називається *зворотнім ходом*.

Обчислювальна складність факторизації і зворотнього ходу залежить від розрідженості та інших властивостей матриці F. Як наслідок, сумарна складність розв'язування системи $Fx = g$ є сумою складностей факторизації F та оберненого ходу.

У нашому випадку матриця коефіцієнтів $F = P + \rho A^T A$ і права частина $g = \rho A^T v - q$, де $P \in \mathbb{S}^n_+$ і $A \in \mathbb{R}^{p \times n}$. Припустимо, що ми не експлуатуємо структуру A чи P, тобто використовуємо загальні методи що спрацюють для довільної матриці. Тоді створення матриці

$F = P + \rho A^\intercal A$ вимагає $O\left(pn^2\right)$ флопс. Факторизація Холецького матриці F здійснюється за $O\left(n^3\right)$ операцій, а зворотній хід є $O(n^2)$. (Вартість створення g нікчемно мала у порівнянні з вище перерахованими).

Резюмуючи, коли p приблизно n або більше, то сумарна складність $O\left(pn^2\right)$. (Коли p менше ніж n на порядок то лема про обернену матрицю описана нижче дозволяє провести крок увесь оновлення за $O\left(p^2n\right)$ флопс).

4.2.2 Експлуатація розрідженості

Коли A і P такі, що F розріджена (тобто має достатньо нульових елементів для експлуатації їх наявності), то можна застосувати ефективніші процедури факторизації і зворотного ходу.

Приклад 4.9

Якщо і P і A діагональні матриці $n \times n$ то як факторизація так і зворотній хід можуть бути виконані за $O(n)$.

Приклад 4.10

Якщо і P і A три-діагональні, або п'яти-діагональні, або l-діагональні (eng. *banded*), то і F також, причому якщо F — k-діагональна, то факторизацію можна провести а $O\left(nk^2\right)$, а зворотній хід — за $O(nk)$. Сумарна вартість кроку оновлення змінної x у цьому випадку складає $O\left(nk^2\right)$.

Зауваження 4.11 — Схожий підхід працює якщо $P + \rho A^\intercal A$ має більш загальний патерн розрідженості, тоді використовують факторизацію Холецького з перестановкою.

4.2.3 Кешування факторизації

Припустимо, що тепер нам необхідно розв'язати багато систем вигляду $Fx^{(i)} = g^{(i)}$, де $i = 1, \ldots, N$, з однаковою матрицею коефіцієнтів, але різними правими частинами. Така ситуація виникає коли штрафний параметр ρ не змінюється в ADMM (принаймні протягом кількох ітерацій). Зрозуміло, що у цьому випадку факторизацію можна виконати лише один раз, а потім просто виконувати кілька послідовних зворотних ходів для кожної правої частини.

Якщо обчислювальна вартість факторизації була t, а обчислювальна складність зворотного ходу s, то сумарна вартість N ітерацій стає $t + N \cdot s$ замість $N \cdot (t + s)$ яку ми б отримали якби проводили факторизацію F на кожному кроці. Тому поки ρ не змінюється, ми можемо розкладати на множники $P + \rho A^\intercal A$ один раз і потім використовувати цю факторизацію у послідовних кроках оновлення змінної x.

Приклад 4.12

Якщо ми навіть не експлуатували розрідженість і використовували стандартну факторизацію Холецького, то такий підхід дозволяє виконувати кроки оновлення змінної x асимптотично (при кількості цих кроків що прямує до нескінченності) ефективніше ніж при наївній реалізації у n разів!

Зауваження 4.13 — Коли розрідженість експлуатується, то відношення $t : s$ зазвичай менше за n але все ще відчутне, тому тут також є виграш у ефективності. Щоправда, у цьому випадку виграш від відсутності необхідності перераховувати факторизацію $P + \rho A^T A$ (тобто незмінності ρ) менший, що змушує зважувати його супроти виграшу від зміни ρ, яка як ми бачили в §3.4.1 може суттєво покращувати швидкість збіжності.

Резюмуючи, ефективна реалізація завжди має запам'ятовувати факторизацію $P + \rho A^T A$ і пере-обчислювати її тільки при зміні ρ, а яким саме буде виграш у ефективності — залежить від конкретної задачі, але він точно є.

4.2.4 Лема про обернення матриці

Лема 4.14 (про обернення матриці)

Рівність нижче виконується якщо всі обернені існують

$$(P + \rho A^T A)^{-1} = P^{-1} - \rho P^{-1} A^T \cdot \left(I + \rho A P^{-1} A^T\right)^{-1} \cdot A P^{-1}. \tag{4.7}$$

Це означає, що СЛАР з матрицею коефіцієнтів P можна ефективно розв'язати і p на порядок менше за n (або принаймні не більше), то і крок оновлення змінної x можна виконати ефективно. Для цього використовується той же трюк: кешується факторизація $I + \rho A P^{-1} A^T$ і використовуються дешевші зворотні ходи.

Приклад 4.15

Якщо P діагональна то наївна реалізація потребує $O\left(n^3\right)$ флопс на першій ітерації і по $O\left(n^2\right)$ на кожній наступній. Якщо тепер $p \leq n$, то збереження множників правої частини рівності з леми замість множників $P + \rho A^T A$ дозволить провести факторизацію за $O\left(np^2\right)$ флопс, у $(n/p)^2$ разів ефективніше, а потім ще й проводити зворотні ходи за $O(np)$ флопс, що у n/p разів ефективніше.

Зауваження 4.16 — Використання леми про обернення матриці для обчислення x^+ також робить дешевшою зміну ρ від ітерації до ітерації: коли P діагональна то ми можемо обчислити $AP^{-1}T$ один раз і потім створювати $I + \rho^k A P^{-1} A^T$ на ітерації k за $O\left(p^3\right)$ флопс. Іншими словами, оновлення ρ коштує додаткових $O(np)$ флопс, тому коли p^2 менше або дорівнює n (за порядком), то немає додаткової складності (за порядком) у тому аби оновлювати ρ хоч на кожній ітерації.

4.2.5 Квадратична функція на афінній множині

Ті ж зауваження справедливі для трохи складнішого випадку опуклої квадратичної функції що розглядається на афінній множині:

$$f(x) = (1/2) \cdot x^T P x + q^T x + r, \quad \operatorname{dom} f = \{x | Fx = g\}. \tag{4.8}$$

У цьому випадку x^+ все ще є афінною функцією змінної ν, а крок оновлення змінної x потребує розв'язання системи ККТ (Каруша-Куна-Такера):

$$\begin{pmatrix} P + \rho I & F^T \\ F & 0 \end{pmatrix} \cdot \begin{pmatrix} x^{k+1} \\ \nu \end{pmatrix} + \begin{pmatrix} q - \rho \cdot (z^k - u^k) \\ -g \end{pmatrix} = 0. \tag{4.9}$$

Бачимо, що всі попередні зауваження справедливі і тут: факторизації все ще можна кешувати, а структура матриць все ще може бути експлуатованою для покращення ефективності факторизації і зворотнього ходу.

4.3 Гладкі цільові доданки

4.3.1 Ітеративні методи

Коли f гладка, то для виконання мінімізації по x можуть бути застосовані загальні ітеративні методи. Найцікавішим є клас методів які вимагають тільки вміння рахувати $\nabla f(x)$ для заданого x, множити вектор на матрицю A, і множини вектор на матрицю A^T. Справа в тому, що саме такі методи гарно підходять для задач великого масштабу.

Приклад 4.17

До таких алгоритмів належать стандартний градієнтний метод, (нелінійний) спряжений градієнтний метод, а також алгоритм Бройдена-Флетчера-Гольдфарба-Шенно з обмеженою пам'яттю, більш відомий як L-BFGS (див. [89, 90]).

Також див. [91] для подальших подробиць.

Збіжність цих методів залежить від обумовленості функції що мінімізується.

Зауваження 4.18 — Наявність квадратичного штрафного доданку $(\rho/2) \cdot \|Ax - \nu\|_2^2$ на практиці покращує обумовленість задачі, а тому і пришвидшує збіжність ітеративних методів.

Зауваження 4.19 — Одним з методів зміни параметру ρ від ітерації до ітерації полягає у тому, щоб збільшувати його поки відповідний ітеративний алгоритм не почне збігатися достатньо швидко.

4.3.2 Рання зупинка

Стандартною технікою що пришвидшує алгоритм є зупиняти ітеративний метод що виконує оновлення змінної x є не доводити його до кінця, тобто перед тим як градієнт від $f(x) + (\rho/2) \cdot$

$\|Ax-v\|_2^2$ стає дуже малим. Ця техніка виправдовується результатами щодо збіжності ADMM з неточною мінімізацією на кроках оновлення змінних x та z.

У цій техніці, на перших кроках оновлення можна зупинятися при досягненні хоча б якоїсь низької точності яка має підвищуватися ближче по мірі кроків щоб гарантувати збіжність. Зрозуміло, що такий підхід призводить до потенційно більшої кількості кроків, зате самі кроки стають набагато дешевшими з обчислювальної точки зору бо потребують значно менше ітерацій ітеративного методу.

4.3.3 Теплий старт

Іншим стандартним трюком є ініціалізація ітеративного методу що використовується у кроці оновлення змінної x не у випадковій чи нульовій точці, а на результаті на якому відбувалася зупинка ітеративного методу на попередньому кроці.

Ця техніка називається *теплим стартом*, бо ми ніби граємо у "тепло-холодно" і починаємо одразу з "тепло", бо зачасту результат попереднього кроку же дає дуже гарне наближення для розв'язку поточного кроку, що дозволяє ще більше зменшити кількість ітерацій ітеративного методу на один крок оновлення змінної x. Особливо відчутна перевага на останніх кроках, коли ADMM вже майже збігся і x^{k+1} не може сильно відрізнятися від x^k.

4.3.4 Квадратичні цільові доданки

Навіть якщо f квадратична, може виявитися що краще застосовувати ітеративний а не аналітичний алгоритм для оновлення змінної x. У таких випадках можна використовувати стандартний (можливо передумовлений) метод спряженого градієнту.

Цей підхід э розумним якщо аналітичні методи не працюють, наприклад коли вони потребують надто багато пам'яті, або коли A густа, але відомий швидкий метод множення векторів на A або на A^T.

Приклад 4.20

Одним з частих таких випадків є матриці A що представляють дискретне перетворення Фур'є, див. [32].

4.4 Декомпозиція

4.4.1 Блочна "роздільність"

Нехай $x = \begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix}^T$ — певне розбиття вектор-змінної x на під-вектори таке, що f "роздільна" відносно цього розбиття, тобто

$$f(x) = f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n), \tag{4.10}$$

де $x_i \in \mathbb{R}^{n_i}$ і $\sum_{i=1}^N n_i = N$. Якщо квадратичний доданок $\|Ax\|_2^2$ також роздільний відносно цього розбиття, тобто якщо $A^T \cdot A$ блочно-діагональна матриця відносно цього розбиття, то і доповнена функція Лагранжа L_ρ також роздільна.

Це все разом означає, що крок оновлення змінної x може бути обчисленим паралельно, де кожний під-вектор x_i бере участь в своїй власній окремій від інших мінімізації.

4.4.2 По-компонентна роздільність

У деяких випадках декомпозиція можлива аж до рівня окремих компонент, тобто

$$f(x) = f_1(x_1) + \ldots + f_n(x_n), \tag{4.11}$$

де $f_i : \mathbb{R} \to \mathbb{R}$, а $A^\intercal \cdot A$ — діагональна матриця. Тоді крок мінімізації за змінною x може бути виконаним як n мінімізацій за *скалярними* змінними, які інколи можуть бути виконаними аналітично, але у будь-якому разі відбуваються надзвичайно ефективно. Така ситуація називається *по-компонентною роздільністю*.

4.4.3 М'яке порогування

Приклад 4.21

На практиці часто зустрічається функція $f(x) = \lambda \cdot \|x\|_1$, де $\lambda > 0$ і $A = I$. У цьому випадку (скалярні) оновлення змінних x_i набувають вигляду

$$x_i^+ \coloneqq \operatorname{argmin}_{x_i} \left(\lambda \cdot |x_i| + (\rho/2) \cdot (x_i - v_i)^2 \right). \tag{4.12}$$

Не зважаючи на недиференційовність першого доданку ми все ще запросто можемо знайти розв'язок цієї задачі у замкненій формі використовуючи субдиференціальне числення (див. [3, §23] для бекграунду). А саме, розв'язок набуває вигляду

$$x_i^+ \coloneqq S_{\lambda/\rho}(v_i), \tag{4.13}$$

де S — *оператор м'якого порогування*, визначений наступним чином:

$$S_\kappa(a) = \begin{cases} a - \kappa, & a > \kappa, \\ 0, & |a| \le \kappa, \\ a + \kappa, & a < -\kappa, \end{cases} \tag{4.14}$$

або, що те саме,

$$S_\kappa(a) = (a - \kappa)_+ - (-a - \kappa)_+. \tag{4.15}$$

Ось ще одна формула яка явно показує, що оператор м'якого порогування є оператором *стиску* у тому розумінні, що він рухає точки до нуля, тобто зменшує $\|\cdot\|_1$-норму:

$$S_\kappa(a) = (1 - \kappa/|a|)_+ \cdot a, \tag{4.16}$$

для $a \neq 0$. Оновлення компонент що виражаються у такому вигляді називатимемо покомпонентним м'яким порогуванням. На мові §4.1, оператор м'якого порогування є проксимальним оператором для норми ℓ_1.

5 Опукла оптимізація з обмеженнями

Загальна задача опуклої оптимізації з обмеженнями має вигляд

$$f(x) \xrightarrow[x \in \mathcal{C}]{} \min, \tag{5.1}$$

де змінна $x \in \mathbb{R}^n$, а f і $\mathcal{C}$ — опуклі. Ця проблема може бути записана в ADMM форми (3.1) наступним чином:

$$f(x) + g(z) \xrightarrow[x-z=0]{} \min, \tag{5.2}$$

де g — індикаторна функція множини $\mathcal{C}$.

Доповнена функція Лагранжа (з масштабованою двоїстою змінною) для цієї задачі має вигляд

$$L_\rho(x, z, u) = f(x) + g(z) + (\rho/2) \cdot \|x - z + u\|_2^2, \tag{5.3}$$

тому масштабована форма ADMM цієї задач наступна:

$$x^{k+1} := \operatorname{argmin}_x \left(f(x) + (\rho/2) \cdot \left\| x - z^k + u^k \right\|_2^2 \right), \tag{5.4}$$

$$z^{k+1} := \Pi_{\mathcal{C}} \left(x^{k+1} + u^k \right), \tag{5.5}$$

$$u^{k+1} := u^k + x^{k+1} - z^{k+1}. \tag{5.6}$$

Крок оновлення змінної x включає мінімізацію суми f і опуклої квадратичної функції, тобто обчислення проксимального оператору пов'язаного з f. Крок оновлення змінної z є евклідовою проекцією на $\mathcal{C}$.

Зауваження 5.1 — Цільова функція f не зобов'язана бути гладкою.

Приклад 5.2

Можна вводити додаткові обмеження (окрім того що $x \in \mathcal{C}$) шляхом покладання $f(x) = +\infty$ скрізь де ці обмеження не виконуються.

Як і у всіх інших задачах де обмеження має вигляд $x - z = 0$, пряма і двоїста нев'язки набувають простих форм:

$$r^k = x^k - z^k, \quad s^k = -\rho \cdot \left(z^k - z^{k-1} \right). \tag{5.7}$$

У подальших розділах ми розглянемо більш конкретні випадки.

5.1 Опукла допустимість

5.1.1 Почергові проекції

Класичною задачею є знаходження спільної точки двох замкнених непорожніх опуклих множин. Класичним методом розробленим ще у 1930-их є метод почергових проецій фон Неймана [92–94]:

$$x^{k+1} := \Pi_{\mathcal{C}}\left(z^k\right), \tag{5.8}$$

$$z^{k+1} := \Pi_{\mathcal{D}}\left(x^{k+1}\right), \tag{5.9}$$

де $\Pi_{\mathcal{C}}$ і $\Pi_{\mathcal{D}}$ — оператори евклідового проектування на множини $\mathcal{C}$ і $\mathcal{D}$ відповідно.

У формі ADMM ця задача може бути записана так:

$$f(x) + g(z) \xrightarrow[x-z=0]{} \min, \tag{5.10}$$

де f і g — індикаторні функції множин $\mathcal{C}$ і $\mathcal{D}$ відповідно. Це дозволяє записати наступну масштабовану форму алгоритму:

$$x^{k+1} := \Pi_{\mathcal{C}}\left(z^k - u^k\right), \tag{5.11}$$

$$z^{k+1} := \Pi_{\mathcal{D}}\left(x^{k+1} + u^k\right), \tag{5.12}$$

$$u^{k+1} := u^k + x^{k+1} - z^{k+1}, \tag{5.13}$$

де обидва кроки оновлення змінних x та z містять проектування на опуклу множину, як і у класичному алгоритмі.

> **Зауваження 5.3 —** Насправді це один в один алгоритм почергових проекцій Дейкстри [95, 96], який куди більш ефективний ніж класичний метод фон Неймана за рахунок використання двоїстої змінної u.

Твердження 5.4

У цій задачі норма прямої нев'язки $\left\|x^k - z^k\right\|_2$ має наглядну інтерпретацію.

Доведення. Оскільки $x^k \in \mathcal{C}$, а $z^k \in \mathcal{D}$, то $\left\|x^k - z^k\right\|_2$ є оцінкою зверху на $\rho(\mathcal{C}, \mathcal{D})$, тобто на евклідову відстань між $\mathcal{C}$ та $\mathcal{D}$.

Якщо ми зупиняємося при $\left\|r^k\right\|_2 \leq \varepsilon^{\text{pri}}$, то це означає, що ми знайшли пару точок, одну в $\mathcal{C}$ а іншу в $\mathcal{D}$ на відстані не більше ε^{pri} одна від одної.

Тоді точка $(1/2) \cdot \left(x^k + z^k\right)$ знаходиться на відстані не більше $\varepsilon^{\text{pri}}/2$ як від $\mathcal{C}$ так і від $\mathcal{D}$. □

5.1.2 Паралельні проекції

Цей же метод застосовується до задачі знаходження спільної точки N замкнених опуклих множин $\mathcal{A}_1, \ldots, \mathcal{A}_N$ з $\mathbb{R}^n$ якщо розглянути задачу наступним чином: ми працюємо у просторі $\mathbb{R}^{nN}$, де

$$\mathcal{C} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_N, \quad \mathcal{D} = \left\{ (x_1, x_2, \ldots, x_N) \in \mathbb{R}^{nN} \middle| x_1 = x_2 = \ldots = x_N \right\}. \tag{5.14}$$

Твердження 5.5

Проекція на $\mathcal{C}$ виражається через проекції на $\mathcal{A}_1, \ldots, \mathcal{A}_N$.

А саме, якщо $x = (x_1, \ldots, x_N) \in \mathbb{R}^{nN}$, то:

$$\Pi_{\mathcal{C}}(x) = (\Pi_{\mathcal{A}_1}(x_1), \ldots, \Pi_{\mathcal{A}_N}(x_N)). \tag{5.15}$$

Твердження 5.6

У свою чергу, проекція на $\mathcal{D}$ має вигляд

$$\Pi_{\mathcal{D}} = (\bar{x}, \bar{x}, \ldots, \bar{x}), \tag{5.16}$$

де $\bar{x} = (1/N) \sum_{i=1}^{N} x_i$ — середнє точок $x_1, \ldots, x_N$.

Як наслідок, кожний крок ADMM складається з паралельного проектування точок на $\mathcal{A}_1, \ldots, \mathcal{A}_N$ і усереднення результатів:

$$x_i^{k+1} := \Pi_{\mathcal{A}_i}\left(z^k - u_i^k\right), \tag{5.17}$$

$$z^{k+1} := \frac{1}{N} \sum_{i=1}^{N} \left(x_i^{k+1} + u_i^k\right), \tag{5.18}$$

$$u_i^{k+1} := u_i^k + x_i^{k+1} - z^{k+1}. \tag{5.19}$$

Тут перший і третій кроки виконуються паралельно для $i = 1, \ldots, N$.

Зауваження 5.7 — Опис вище використовує трохи спрощений запис опускаючи індекси i в z_i оскільки всі z_i однакові.

Отриманий алгоритм може трактуватися як особливий випадок оптимізації з обмеженнями описаний у §4.4, де індикаторна функція множини $\mathcal{A}_1 \cap \ldots \cap \mathcal{A}_N$ розбивається у суму індикаторних функцій усіх $\mathcal{A}_i$.

Зауваження 5.8 — Беручи середнє по i в останньому рівнянні ми отримуємо

$$\bar{u}^{k+1} = \bar{u}^k + \bar{x}^{k+1} = z^k, \tag{5.20}$$

що у поєднанні з $z^{k+1} = \bar{x}^{k+1} + \bar{u}^k$ з другого рівняння дає $\bar{u}^{k+1} = 0$.

Тобто після першого кроку середнє u_i дорівнює нулю.

Це означає, що $z^{k+1} = \bar{x}^{k+1}$.

Використовуючи ці спрощення ми приходимо до наступного простого алгоритму:

$$x_i^{k+1} := \Pi_{\mathcal{A}_i}\left(\bar{x}^k - u_i^k\right), \tag{5.21}$$

$$u_i^{k+1} := u_i^k + \left(x_i^{k+1} - \bar{x}^{k+1}\right). \tag{5.22}$$

Ця форма показує, що u_i^k є сумою відхилень $x_i^k - \bar{x}^k$ за всі попередні ітерації (у припущенні що $u^0 = 0$). Перший крок цього алгоритму є паралельними проекціями на $\mathcal{A}_i$, а другий включає усереднення отриманих проекцій.

Існує дуже багато літератури стосовно алгоритмів послідовних проекцій та їхніх застосувань. Див. [97] для загального огляду, [98] для застосувань в обробці зображень, [19, §5] для обговорення у контексті роз-паралеленої оптимізації.

5.2 Лінійне і квадратичне програмування

Стандартний вигляд задачі квадратичного програмування наступний:

$$(1/2)x^T P x + q^T x \xrightarrow[\substack{Ax=b\\ x\geq 0}]{} \min, \tag{5.23}$$

де змінна $x \in \mathbb{R}^n$, а $P \in S_+^n$.

Зауваження 5.9 — Коли $P = 0$ ця задача перетворюється на стандартну задачу лінійного програмування.

Представимо цю задачу у ADMM-вигляді:

$$f(x) + g(z) \xrightarrow[x-z=0]{} \min, \tag{5.24}$$

де

$$f(x) = (1/2)x^T P x + q^T x, \tag{5.25}$$

з

$$\operatorname{dom} f = \{x \mid Ax = b\} \tag{5.26}$$

— початкова цільова функція з початковою допустимою областю, а g — індикаторна функція невід'ємного ортанту $\mathbb{R}_+^n$.

Тоді масштабований вигляд ADMM складається з ітерацій

$$x^{k+1} := \operatorname*{argmin}_x \left(f(x) + (\rho/2)\left\|x - z^k + u^k\right\|\right) \tag{5.27}$$

$$z^{k+1} := \left(x^{k+1} + u^k\right)_+ \tag{5.28}$$

$$u^{k+1} := u^k + x^{k+1} - z^{k+1}. \tag{5.29}$$

Як описано в §4.2.5, крок оновлення змінної x є задачею МНК з обмеженням типу рівність, і умови оптимальності цієї задачі мають вигляд

$$\begin{pmatrix} P + \rho I & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x^{k+1} \\ b \end{pmatrix} + \begin{pmatrix} q - \rho\left(z^k - u^k\right) \\ -b \end{pmatrix} = 0. \tag{5.30}$$

Зауваження 5.10 — Усі зауваження щодо ефективних обчислень зроблені в §4.2, такі як кешування факторизації для здешевлення подальших ітерацій, можуть бути застосовані і тут.

Приклад 5.11

Якщо P діагональна (можливо нульова), то перший крок оновлення змінної x можна виконати за $O(np^2)$ флопс, де p — кількість обмежень типу рівність у початкові задачі квадратичного програмування, а наступні кроки — за $O(np)$ флопс.

5.2.1 Лінійне і квадратичне програмування на конусі

У загальному випадку, можна вводити довільне конічне обмеження $x \in \mathcal{K}$ замість $x \geq 0$, тоді стандартна задача квадратичного програмування (5.23) стає задачею квадратичного програмування на конусі.

Єдиною зміною в ADMM буде крок оновлення змінної z, який тепер потребуватиме проектування на $\mathcal{K}$.

Приклад 5.12

Можна розв'язувати напіввизначену задачу з $x \in S_+^n$ шляхом проектування $x^{k+1} + u^k$ на S_+^n з використанням спектрального розкладу.

6 Задачі з ℓ_1-нормою

Задачі що розглядаються у цьому розділі допоможуть продемонструвати, чому ADMM є природнім вибором для задач машинного навчання і статистики. Причина у тому, що на відміну від двоїстого сходження чи методу множників, ADMM явно націлюються на задачі які розділяються на дві різні частини, f і g, з якими можна розбиратися незалежно.

Задачі такого вигляду домінують у машинному навчанні, адже значна кількість задач навчання передбачає мінімізацію функції втрат разом із штрафнии доданком, або із сторонніми обмеженнями. У інших випадках ці обмеження виникають за рахунок переведення початкової задачі у якусь загальноприйнятий вигляд.

Цей розділ містить багато простих але важливих задач з ℓ_1 нормами. На сьогоднішній день зацікавленість у цих задачах особливо поширена серед спеціалістів по задачам статистики, машинного навчання, і обробки сигналів.

Застосування ADMM зачасту призводить до state-of-the-art методів, або до близьких до таких. Ми покажемо, що ADMM дозволяє природнім чином виділити негладкий ℓ_1 доданок з гладкої функції втрат, що є вигідним з обчислювальної точки зору. У цьому розділі ми сфокусуємося на не розподілених версіях цих задач для простоти; задачі розподіленого підбору моделей будуть розглянуті далі.

Ідеї ℓ_1-регуляризації вже десятки років, вона простежується з роботи Губера [99] щодо стійкої статистики, і статті Клербо [100] з геофізики. Існує обширна література з цього приводу, але кількома важливими сучасними статтями є [101] щодо варіаційного знешумлення, [102] щодо м'якого порогування, [103] щодо ласо, [104] щодо виділення базису, [105–107] щодо скомпресованого відчуття і [108] щодо структурного навчання розріджених графічних моделей.

Поширення моделей з ℓ_1-регуляризацією викликало значний обсяг досліджень алгоритмів розв'язування таких задач. Недавній огляд Янга та ін. [109] порівнює швидкодію багатьох різних алгоритмів, включаючи метод проекції градієнту [110, 111], гомотопічні методи [112], ітеративне звуження порогування [113], проксимальні градієнтні методи [114–117], доповнені методи Лагранжа [118], і методи внутрішньої точки [119].

Є також інші підходи, як-то ітеративні алгоритми Брегмана [120] і ітеративні алгоритми порогування [121], які реалізуються у фреймворку обміну повідомленнями.

6.1 Найменше абсолютне відхилення

Спрощеним варіантом задачі найменших квадратів є *найменше абсолютне відхилення*, у якому ми мінімізуємо $\|Ax-b\|_1$ замість $\|Ax-b\|_2$. Найменші абсолютні відхилення надають більш стійку модель ніж МНК у випадках коли дані містять великі аномалії, і часто використовуються у статистиці та економетриці[8].

У формі ADMM цю задачу можна записати як

$$\|z\|_1 \xrightarrow[Ax-z=b]{} \min, \tag{6.1}$$

[8] Див., наприклад, [122, §10.6], [123, §9.6], і [2, §6.1.2].

тобто $f = 0$ і $g = \|\cdot\|_1$.

Використання особливого вигляду f і g у припущенні невиродженості $A^\intercal A$ дозволяє записати ітерації у наступному вигляді:

$$x^{k+1} \coloneqq (A^\intercal A)^{-1} A^\intercal \left(b + z^k - u^k\right) \tag{6.2}$$

$$z^{k+1} \coloneqq S_{1/\rho}\left(Ax^{k+1} - b + u^k\right) \tag{6.3}$$

$$u^{k+1} \coloneqq u^k + Ax^{k+1} - z^{k+1} - b, \tag{6.4}$$

де оператор м'якого порогування застосовується по-координатно.

Зауваження 6.1 — Як і в §4.2, матрицю $A^\intercal A$ можна факторизувати всього лише один раз, а закешовані множники потім використовуються для здешевлення подальших кроків оновлення змінної x.

Крок оновлення змінної x це МНК із матрицею коефіцієнтів A і правою частиною $b+z^k-u^k$. Таким чином ADMM можна розглядати як метод розв'язування задачі мінімальних абсолютних відхилень шляхом ітеративного розв'язування асоційованих задач найменших квадратів зі змінною правою частиною. Змінена права частина оновлюється оператором м'якого порогування.

Зауваження 6.2 — З кешуванням факторизації обчислювальна складність розв'язування подальших задач найменших квадратів набагато менше ніж першої, що зачасту дозволяє розв'язувати задачу найменших абсолютних відхилень ледь не так само швидко як і задачу найменших квадратів.

6.1.1 Задача Губера

Задачею, яка знаходиться між найменшими квадратами і найменшими абсолютними відхиленнями є задача з *функцією Губера*:

$$g^{\text{hub}}(Ax - b) \to \min, \tag{6.5}$$

де *штрафна функція Губера* є квадратичною для малих значень аргументу і переходить у абсолютне значення для більших значень аргументів.

Визначення 6.3. Для скаляру a вона задається наступним чином:

$$g^{\text{hub}}(a) = \begin{cases} a^2/2, & |a| \le 1, \\ |a| - 1/2, & |a| > 1. \end{cases} \tag{6.6}$$

Зауваження 6.4 — На векторний аргумент функція Губера узагальнюється як сума значень функції Губера на компонентах.

Зауваження 6.5 — Для простоти ми розглядаємо стандартизовану функцію Губера яка змінює поведінку з квадратичної на лінійну у точках з $|a| = 1$, хоча взагалі кажучи можна розглядати $|a| = \text{const}$.

Цю задачу можна розглядати у такій же ADMM формі як і вище, з єдиною різницею у кроці оновлення змінної z. Тепер він використовуватиме проксимальний оператор функції Губера а не ℓ_1-норми:

$$z^{k+1} := \frac{\rho}{1+\rho}\left(Ax^{k+1} - b + u^k\right) + \frac{1}{1+\rho} \cdot S_{1+1/\rho}\left(Ax^{k+1} - b + u^k\right). \tag{6.7}$$

Зауваження 6.6 — Коли розв'язок МНК $x^{ls} = (A^\intercal A)^{-1} b$ задовольняє умову $\left|x_i^{ls}\right| \le 1$ для всіх i, то він також є розв'язком задачі Губера, і ADMM зупиняється за два кроки.

6.2 Вибір базису

Вибір базису це задача мінімізації з нормою ℓ_1 з обмеженням типу рівність у формі

$$\|x\|_1 \xrightarrow[Ax=b]{} \min, \tag{6.8}$$

де змінна $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $m < n$.

Вибір базису часто застосовується як евристика для пошуку розрідженого розв'язку недовизначеної СЛАР.

Зауваження 6.7 — Ця задача відіграє центральну роль у сучасній статистичній обробці сигналів, зокрема у теорії стиснутого сприйняття (див. [124] для свіжого огляду стану справ у цій галузі).

У ADMM-формі задача вибору базису може бути записана так:

$$f(x) + \|z\|_1 \xrightarrow[x-z=0]{} \min, \tag{6.9}$$

де f — індикаторна функція множини $\{x \in \mathbb{R}^n \mid Ax = b\}$.

Тоді ітерації ADMM запишуться у вигляді

$$x^{k+1} := \Pi\left(z^k - u^k\right) \tag{6.10}$$

$$z^{k+1} := S_{1/\rho}\left(x^{k+1} + u^k\right) \tag{6.11}$$

$$u^{k+1} := u^k + x^{k+1} - z^{k+1}, \tag{6.12}$$

де Π — оператор проектування на $\{x \in \mathbb{R}^n \mid Ax = b\}$.

Крок оновлення змінної x який вимагає розв'язування задачі мінімізації Евклідової норми з лінійним обмеженням можна явно записати наступним чином:

$$x^{k+1} := \left(I - A^\intercal(AA^\intercal)^{-1}A\right)\left(z^k - u^k\right) + A^\intercal(AA^\intercal)^{-1}b. \tag{6.13}$$

Зауваження 6.8 — Знову ж таки, зауваження з §4.2 щодо ефективної реалізації застосовні тут; кешування факторизації $AA^\intercal$ робить подальші проекції значно обчислювально дешевшими ніж першу.

Зауваження 6.9 — Можна інтерпретувати ADMM для задачі вибору базису як зведення однієї задачі мінімізації ℓ_1 норми до розв'язування послідовності задач мінімізації Евклідової норми. Для обговорення схожих алгоритмів для обробки зображень див. [125].

Зауваження 6.10 — Не так давно виник ще один клас алгоритмів, так звані ітеративні методи Брегмана, які подають надії для задач з нормою ℓ_1. Для задачі вибору базису і суміжних, ітеративна регуляризація Брегмана [120] виявляється еквівалентною методу множників, а розділений метод Брегмана [126] еквівалентний до ADMM (див. [127] для доведення).

6.3 Загальна задача мінімізації з ℓ_1-регуляризацією

Розглянемо загальну задачу

$$\ell(x) + \lambda\|x\|_1 \to \min, \tag{6.14}$$

де ℓ — якась опукла функція втрат.

У ADMM-формі цю задачу можна записати як

$$\ell(x) + g(z) \xrightarrow[x-z=0]{} \min, \tag{6.15}$$

де $g(z) = \lambda\|z\|_1$.

Тоді ітерації будуть

$$x^{k+1} := \operatorname*{argmin}_x \left(\ell(x) + (\rho/2) \left\| x - z^k + u^k \right\|_2^2 \right) \tag{6.16}$$

$$z^{k+1} := s_{\lambda/\rho}\left(x^{k+1} + u^k \right) \tag{6.17}$$

$$u^{k+1} := u^k + x^{k+1} - z^{k+1}. \tag{6.18}$$

Зауваження 6.11 — Крок оновлення змінної x вимагає обчислення проксимального оператору.

Приклад 6.12

Якщо ℓ гладка, то це можна зробити довільним стандартним методом, як-то метод Ньютона, або квазі-ньютонівським методом, як-то L-BFGS, або методом спряженого градієнту.

Приклад 6.13

Якщо ℓ квадратична, то крок мінімізації змінної x можна виконати розв'язавши СЛАР, як описано в §4.2.

Зауваження 6.14 — У загальному можна інтерпретувати ADMM для функції втрат з ℓ_1-регуляризацією як спрощення мінімізації цієї функції до послідовності мінімізацій задач з функціями втрат з ℓ_2.

Зауваження 6.15 — Дуже велика кількість моделей може бути представлена у подібному вигляді, включаючи узагальнені лінійні моделі [128] і узагальнені адитивні моделі [129].

Приклад 6.16

Узагальнені лінійні моделі включають лінійну регресію, логістичну регресію, softmax регресію, і навіть регресію Пуассона, адже вони дозволяють довільну експоненційну сім'ю функцій.

Зауваження 6.17 — Для загального опису моделей типу логістичної регресії з ℓ_1-регуляризацією див., наприклад, [122, §4.4.4].

Зауваження 6.18 — Для використання будь-якого іншого регуляризатора $g(z)$ замість $\|z\|_1$, ми просто замінюємо оператор м'якого порогування у кроці оновлення змінної z на проксимальний оператор функції g, як описано в §4.1.

6.4 Ласо

Важливим частинним випадком задачі (6.14) є лінійна регресія з ℓ_1-регуляризацією, яка також називається ласо [103]. Вона передбачає розв'язання задачі

$$(1/2)\,\|Ax - b\|_2^2 + \lambda\|x\|_1, \tag{6.19}$$

де $\lambda > 0$ — скалярний параметр регуляризації, який зазвичай обирається шляхом перехресної валідації.

У класичних застосуваннях зачасту є набагато більше факторів ніж тренувальних об'єктів, а ціллю є побудова якомога простішої моделі.

Зауваження 6.19 — Для загального огляду ласо див. [122, §3.4.2].

Ласо часто застосовується, зокрема для аналізу біологічних даних, де лише мала частина з дуже великої кількості потенційних факторів реально впливає на цікавий для дослідників результат.

Зауваження 6.20 — Для детального огляду подібних медичних досліджень див. [122, §18.4].

У ADMM-формі цю задачу можна записати наступним чином:

$$f(x) + g(z) \xrightarrow[x-z=0]{} \min, \tag{6.20}$$

де $f(x) = (1/2)\|Ax - b\|_2^2$, а $g(z) = \lambda\|z\|_1$.

Використовуючи зауваження з §4.2 і §4.4, можемо записати наступні ітерації:

$$x^{k+1} := (A^\intercal A + \rho I)^{-1}\left(A^\intercal b + \rho\left(z^k - u^k\right)\right) \tag{6.21}$$

$$z^{k+1} := S_{\lambda/\rho}\left(x^{k+1} + u^k\right) \tag{6.22}$$

$$u^{k+1} := u^k + x^{k+1} - z^{k+1}. \tag{6.23}$$

Зауваження 6.21 — Матриця $A^\intercal A + \rho I$ завжди невироджена, бо $\rho > 0$.

Зауваження 6.22 — Крок оновлення змінної x є по суті гребеневою регресією (тобто квадратично регуляризованим МНК), тому ADMM можна інтерпретувати як метод розв'язування задачі ласо шляхом розв'язування послідовності гребеневих регресій.

Зауваження 6.23 — При використанні прямих методів розв'язування задачі гребеневої регресії обчислювально вигідно кешувати факторизацію на першій ітерації, аби подальші можна було виконати з меншими обчислювальними складностями.

Зауваження 6.24 — Див. [130] для прикладу застосування в обробці зображень.

6.4.1 Узагальнене ласо

Задачу ласо можна узагальнити наступним чином:

$$(1/2)\,\|Ax - b\|_2^2 + \lambda\,\|Fx\|_1 \to \min, \tag{6.24}$$

де F — довільне лінійне перетворення.

Приклад 6.25

Важливим випадком є різницева матриця $F \in \mathbb{R}^{(n-1)\times n}$, тобто

$$F_{i,j} = \begin{cases} 1, & j = i+1, \\ -1, & j = i, \\ 0, & \text{інакше}, \end{cases} \tag{6.25}$$

і одинична матриця $A = I$, тоді отримуємо наступну задачу:

$$(1/2)\|x - b\|_2^2 + \lambda \sum_{i=1}^{n-1} |x_{i+1} - x_i|. \tag{6.26}$$

Визначення 6.26. Тут другий доданок є *сумарною варіацією* від x. Ця задача зазвичай називається *знешумленням сумарної варіації* [101] і має багато застосувань у обробці сигналів.

Визначення 6.27. Коли $A = I$, а F є другою різницевою матрицею, то узагальнена задача ласо має назву ℓ_1-*підбору тренду* [131].

У ADMM-формі узагальнену задачу ласо можна записати наступним чином:

$$(1/2)\,\|Ax - b\|_2^2 + \lambda\|z\|_1 \xrightarrow[Fx-z=0]{} \min, \tag{6.27}$$

що призводить до наступних ітерацій:

$$x^{k+1} := (A^TA + \rho F^TF)^{-1}\left(A^Tb + \rho F^T\left(z^k - u^k\right)\right) \tag{6.28}$$

$$z^{k+1} := S_{\lambda/\rho}\left(Fx^{k+1} + u^k\right) \tag{6.29}$$

$$u^{k+1} := u^k + Fx^{k+1} - z^{k+1}. \tag{6.30}$$

Приклад 6.28

Для випадку знешумлення сумарної варіації матриця $A^TA+\rho F^TF$ є тридіагональною, тому крок оновлення змінної x можна виконати за $O(n)$ флопс (див. [32, §4.3]).

Приклад 6.29

Для ℓ_1-підбору тренду ця матриця є п'ятидіагональною, і крок оновлення змінної x все ще може бути виконаним за $O(n)$ флопс.

6.4.2 Групове (блочне) ласо

Іншим прикладом є заміна регуляризатора $\|x\|_1$ на

$$\sum_{I=1}^{N} \|x_i\|_2, \tag{6.31}$$

де $x = \begin{pmatrix} x_1 & \cdots & x_N \end{pmatrix}$, де $x_i \in \mathbb{R}^{n_i}$. Зрозуміло, що якщо $n_i = 1$ і $N = n$, то це просто ласо.

Як бачимо, тут регуляризатор є розділимим відносно розбиття $x_1, \ldots, x_N$, але не цілком розділимим. Це розширення ℓ_1-резуляризації називається *групове ласо* [132], або, більш загально, *регуляризація сумою норм* [133].

ADMM для цієї задачі такий же як і вище, за виключенням того, що крок оновлення змінної z може бути заміненим наступним блочним м'яким порогуванням:

$$z_i^{k+1} = S_{\lambda/\rho}\left(x_i^{k+1} + u^k\right), \quad i = 1, \ldots, N, \tag{6.32}$$

де оператор м'якого порогування $S_\kappa : \mathbb{R}^m \to \mathbb{R}^m$ має вигляд

$$S_\kappa(a) = (1 - \kappa/\|a\|_2)_+ a, \tag{6.33}$$

де $S_\kappa(0) = 0$. Ця формула є звичайним скалярним оператором м'якого порогування коли a — скаляр, і узагальнює вигляд даний в §4.2.

Подібний підхід можна розвинути щоб працювати з недиз'юнктним розбиттям, що часто зустрічається у біоінформатиці та інших прикладних випадках [134, 135]. А саме, якщо є N потенційно перетинних груп $G_i \subseteq \{1, \ldots, n\}$, а цільова функція має вигляд

$$(1/2)\,\|Ax - b\|_2^2 + \lambda \sum_{i=1}^{N} \|x_{G_i}\|_2\,, \tag{6.34}$$

де x_{G_i} — підвектор, що містить компоненти з індексами з G_i. Оскільки групи можуть перетинатися, то цю задачу дуже складно розв'язати більшістю класичних методів, але виявляється що вона робиться "в лоб" за допомогою ADMM.

Справді, введемо N нових змінних $x_i \in \mathbb{R}^{|G_i|}$, і розглянемо задачу

$$(1/2)\,\|Az - b\|_2^2 + \lambda \sum_{i=1}^{N} \|x_i\|_2 \xrightarrow[\substack{x_i - z_i = 0,\\ i=1,\ldots,N}]{} \min, \tag{6.35}$$

де змінні x_i локальні, змінна z глобальна, а z_i — "уявлення" z про те, якими мають бути x_i, воно визначається лінійною функцією від z.

6.5 Оцінка матриці коваріації з розрідженою оберненою

Нехай задано вибірку значень з нормального розподілу в $\mathbb{R}^n$:

$$a_i \sim \mathcal{N}(0, \Sigma), \quad i = 1, \ldots, N. \tag{6.36}$$

Розглянемо задачу оцінки матриці коваріації Σ у припущенні що Σ^{-1} розріджена. Оскільки $\left(\Sigma^{-1}\right)_{i,j}$ дорівнює нулю тоді і тільки тоді, коли i-та і j-та компоненти випадкової величини умовно незалежні, то ця задача є еквівалентною задачі *структурного навчання* оцінки топології неорієнтованої графічної моделі представлення Гаусіану [136].

Визначення 6.30. Визначення природи розрідженості оберненої матриці коваріації Σ^{-1} також називається задачею *вибору коваріації*.

Зауваження 6.31 — Для дуже малих n теоретично можливо перебрати усі можливі патерни розрідженості в Σ^{-1}, оскільки для фіксованого патерну задача максимізації вірогідності Σ є задачею опуклої оптимізації.

Гарною евристикою яка поширюється на набагато більші значення n є мінімізація мінус логарифму вірогідності (по параметру $X = \Sigma^{-1}$) з ℓ_1-регуляризацією для "підтримки" розрідженішим варіантам [137]. Якщо S — емпірична матриця коваріації, тобто

$$S = \frac{1}{N} \sum_{i=1}^{N} a_i a_i^T, \tag{6.37}$$

то нашу задачу (мінімізації мінус логарифму вірогідності) можна записати у вигляді

$$\operatorname{tr}(SX) - \log\det X + \lambda \|X\|_1 \to \min, \tag{6.38}$$

де змінна $X \in S_+^n$, а $\|\cdot\|_1$ визначена поелементно, тобто як сума модулів усіх елементів матриці, а областю визначення $\log\det$ є S_{++}^n, множина симетричних додатновизначених $n \times n$ матриць. Це особливий випадок загальної ℓ_1-регуляризованої задачі з опуклою функцією втрат

$$\ell(X) = \operatorname{tr}(SX) - \log\det X. \tag{6.39}$$

Ідея вибору коваріації належить Демпстеру [138] і вперше була розглянути у розрідженому високорозмірному випадку Майншаусеном і Бюльманом [108]. Вищенаведена постановка задачі належить Банджері та ін. [137] Деякі недавні статті щодо цієї задачі включають *графічне ласо* Фрідмена та ін. [139], роботу Дачу та ін. [140], Лу [141], Юана [68] і Шайнберга та ін. [142], яка показу, що ADMM перевершує state-of-the-art алгоритми для цієї задачі.

Ітерації ADMM мають вигляд

$$X^{k+1} := \operatorname*{argmin}_X \left(\operatorname{tr}(SX) - \log\det X + (\rho/2) \left\| X - Z^K + U^k \right\|_F^2 \right) \tag{6.40}$$

$$Z^{k+1} := \operatorname*{argmin}_Z \left(\lambda \|Z\|_1 + (\rho/2) \left\| X^{k+1} - Z + U^k \right\|_F^2 \right) \tag{6.41}$$

$$U^{k+1} := U^k + X^{K+1} - Z^{k+1} \tag{6.42}$$

де $\|\cdot\|_F$ — норма Фробеніуса, тобто квадратний корінь з суми квадратів елементів.

Твердження 6.32

Цей алгоритм можна значно спростити.

Доведення. Крок оновлення змінної Z є поелементним м'яким порогуванням:

$$Z_{i,j}^{k+1} := S_{\lambda/\rho}\left(X_{i,j}^{K+1} + U_{i,j}^k \right), \tag{6.43}$$

а крок оновлення змінної X, як виявляється, має має аналітичний розв'язок.

Справді, критерієм оптимальності першого порядку є рівність градієнта нулеві, тобто

$$S - X^{-1} + \rho\left(X - z^k + u^k\right) = 0, \tag{6.44}$$

разом з неявним обмеженням $X \succ 0$. Це можна переписати як

$$\rho X - X^{-1} = \rho\left(z^k - u^k\right) - S. \tag{6.45}$$

Побудуємо тепер матрицю X яка задовольняє цій умові, а тому і мінімізує цільову функцію у кроці оновлення змінної X.

Спершу, візьмемо ортоспектральний розклад правої частини,

$$\rho\left(z^k - u^k\right) - S = Q\Lambda Q^T, \tag{6.46}$$

де $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, а $Q^TQ = QQ^T = I$. Множимо (6.45) на Q^T зліва і на Q справа, отримуємо

$$\rho\tilde{X} - \tilde{x}^{-1} = \Lambda, \tag{6.47}$$

де $\tilde{X} = Q^TXQ$.

Тепер можемо побудувати діагональний розв'язок цього рівняння, тобто знайти додатні числа $\tilde{X}_{i,i}$ що задовольняють рівнянню

$$\rho\tilde{X}_{i,i} - 1/\tilde{X}_{i,i} = \lambda_i. \tag{6.48}$$

А це квадратне рівняння, тому

$$\tilde{X}_{i,i} = \frac{\lambda_i + \sqrt{\lambda_i^2 + 4\rho}}{2\rho}, \tag{6.49}$$

яке завжди додатне адже $\rho > 0$.

Далі лишається обчислити $X = Q\tilde{X}Q^T$ і отримаємо розв'язок (6.45).

Зауваження 6.33 — Як наслідок, обчислювальна складність кроку оновлення змінної X приблизно дорівнює складності ортоспектрального розкладу симетричної матриці.

□

7 Висновки

Ми детально обговорили ADMM і продемонстрували його застосування до розподіленої опуклої оптимізації в цілому, і зокрема до багатьох проблем у статистичному машинному навчанні. Ми стверджуємо, що ADMM може слугувати універсальним інструментом для оптимізаційних задач, що виникають при аналізі та обробці сучасних масивних датасетів. ADMM слід розглядати як розроділений аналог градієнтного спуск і методу спряженого градієнту, які є стандартними інструментами для гладкої оптимізації на одному процесорі.

ADMM знаходиться на вищому рівні абстракції аніж класичні оптимізаційні алгоритми як метод Ньютона. У класичних алгоритмах базові операції є низькорівневими, здебільшого складаються з операцій лінійної алгебри і обчислень градієнтів і Гессіанів. У випадку з ADMM, базові операції включають розв'язування[9] простих задач опуклої оптимізації. Наприклад, застоування ADMM до дуже великої задачі підбору моделі, кожний крок оновлення зводиться до підбору регуляризованої моделі під менший датасет. Ці підзадачі можуть бути розв'язані використовуючи довільний класичний алгоритм який гарно підходить до малих або середніх задач. У цьому розумінні ADMM спирається на вже існуючі алгоритми для одного процесора, а тому може розглядатися як алгоритм координації який "стимулює" множину простіших алгоритмів "співпрацювати" для розв'язування набагато складнішої глобальної задачі ніж вони могли б поодинці. З іншого боку, ADMM можна розглядати як простий спосіб "самоналаштування" спеціалізованих алгоритмів для малих і середніх задачі для роботи над набагато більшими задачами, які неможливо було б розв'язати без ADMM.

Ми наголошуємо, що для довільної конкретної проблеми цілком може виявитися, що існує іншим, дуже спеціалізований алгоритм який впорається із цією задачею краще ніж ADMM, або навіть якийсь покращений варіант власне ADMM який значно покращить результати. Однак, простіший алгоритм який виводиться з базового ADMM завжди буде працювати принаймні непогано[10] у порівнянні зі спеціалізованими алгоритмами, і у більшості випадків результати його роботи вже будуть достатньо гарними для використання. У деяких випадках алгоритми засновані на ADMM насправді виявляються state-of-the-art алгоритмами навіть у послідовній реалізації. Окрім цього, ADMM має перевагу у простоті реалізації, і гарно накладається на реалізовані у сучасних мовах програмування моделі даних для розподіленого програмування.

ADMM був розроблений вже ціле покоління назад, з коренями які тягнуться у часи коли навіть не було Інтернету, розподілених і хмарних обчислень, гігантських високорозмірних датасетів, і асоційованих з ними величезних задач прикладної статистики. Незважаючи на це, він гарно підходить до сучасної організації обчислень, і є доволі загальним у розумінні кількості різних задач до яких ADMM може бути застосованим.

[9] У деяких випадках аналітичне.
[10] Навіть при послідовній реалізації.

A Доведення збіжності

Основний результат щодо збіжності, який згадується у §3.2 можна знайти у кількох джерелах, як-то [39, 58]. Багато з цих джерел надають також більш складні результати з загальнішими штрафними доданками і неточною мінімізацією. Для повноти наведемо тут доведення основного результату.

Теорема

Ми покажемо, що якщо f і g — замкнуті, власні, і опуклі, а Лагранжіан L_0 має сідлову точку, то ми маємо збіжність прямої нев'язки $r^k \to 0$, збіжність цільової функції $p^k \to p^\star$, де $p^k = f\left(x^k\right) + g\left(z^k\right)$. Ми також покажемо збіжність двоїстої нев'язки $s^k = \rho A^\intercal B\left(z^k - z^{k-1}\right)$ до нуля.

Доведення. Нехай $(x^\star, z^\star, y^\star)$ — сідлова точка L_0, визначимо тоді

$$V^k = (1/\rho)\left\|y^k - y^\star\right\|_2^2 + \rho\left\|B\left(z^k - z^\star\right)\right\|_2^2.$$

Покажемо, що V^k є *функцією Ляпунова* нашого алгоритму, тобто вона є невід'ємним числом яке спадає на кожній ітерації.

Зауваження A.1 — Значення V^k невідоме, оскільки залежить від невідомих значень $z^\star$ і $y^\star$.

Спершу опишемо головну ідею. Доведення покладається на три ключові нерівності, які ми доведемо нижче з використанням базових результатів опуклого аналізу і простої лінійної алгебри.

Лема (перша нерівність)

$$V^{k+1} \leq V^k - \rho\left\|r^{k+1}\right\|_2^2 - \rho\left\|B\left(z^{k+1} - z^k\right)\right\|_2^2. \tag{A.1}$$

Вона каже, що V^k зменшується на кожній ітерації на якесь значення, залежне від норм нев'язок, і від зміни z від ітерації до ітерації. Оскільки $V^k \leq V^0$, то звідси випливає обмеженість y^k і Bz^k.

Ітерування цієї нерівності дає

$$\rho\sum_{k=0}^{\infty}\left(\left\|r^{k+1}\right\|_2^2 + \left\|B\left(z^{k+1} - z^k\right)\right\|_2^2\right) \leq V^0,$$

звідки випливає, що $r^k \to 0$ і $B\left(z^{k+1} - z^k\right) \to 0$ при $k \to \infty$. Множення другого виразу на $\rho A^\intercal$ показує, що двоїста нев'язка $s^k = \rho A^\intercal B\left(z^{k+1} - z^k\right)$ збігається до нуля.

Зауваження A.2 — Це показує, що критерій зупинки (3.30), який вимагає одночасної малості прямої і двоїстої нев'язок, рано чи пізно таки виконається і алгоритм зупиниться.

Лема (друга ключова нерівність)

$$p^{k+1} - p^\star \le -\left(y^{k+1}\right)^\intercal r^{k+1} - \rho\left(B\left(z^{k+1} - z^k\right)\right)^\intercal \left(-r^{k+1} + B\left(z^{k+1} - z^\star\right)\right). \tag{A.2}$$

і

Лема (третя)

$$p^\star - p^{k+1} \le (y^\star)^\intercal r^{k+1}. \tag{A.3}$$

Права частина (A.2) прямує до нуля при $k \to \infty$ адже

$$B\left(z^{k+1} - z^\star\right)$$

обмежена, і як r^{k+1} так і $B\left(z^{k+1} - z^k\right)$ прямують до нуля. Права частина (A.2) прямує до нуля при $k \to \infty$ бо r^k прямує до нуля. Отже маємо $\lim_{k\to\infty} p^k = p^\star$, тобто цільова функція збігається. □

Перед доведенням усіх трьох ключових нерівностей виведемо нерівність (3.28) (з вищезгаданого критерію зупинки) з нерівності (A.2): просто помітимо, що $-r^{k+1} + B\left(z^{k+1} - z^k\right) = -A\left(x^{k+1} - x^\star\right)$; підстановка цього в (A.2) дає (3.28):

$$p^{k+1} - p^\star \le -\left(y^{k+1}\right)^\intercal r^{k+1} + \left(x^{k1} - x^\star\right)^\intercal s^k.$$

Доведення нерівності (A.3)

Доведення. Оскільки $(x^\star, z^\star, y^\star)$ — сідлова точка L_0, то маємо

$$L_0\left(x^\star, z^\star, y^\star\right) \le L_0\left(x^{k+1}, z^{k+1}, y^\star\right).$$

Використовуючи $Ax^\star + Bz^\star = c$, знаходимо, що ліва частина дорівнює $p^\star$. Враховуючи, що $p^{k+1} = f\left(x^{k+1}\right) + g\left(z^{k+1}\right)$, нерівність вище може бути записана у вигляді

$$p^\star \le p^{k+1} + (y^\star)^\intercal r^{k+1},$$

звідки безпосередньо випливає (A.3) □

Доведення нерівності (A.2)

Доведення. За визначенням, x^{k+1} мінімізує $L_\rho\left(x, z^k, y^k\right)$. Оскільки f замкнена, власна, і опукла, то вона субдиференційовна, а тому такою ж є L_ρ. Тоді (необхідними і достатніми) умовами оптимальності будуть[11]

$$0 \in \partial L_\rho\left(x^{k+1}, z^k, y^k\right) = \partial f\left(x^{k+1}\right) + A^\intercal y^k + \rho A^\intercal\left(Ax^{k+1} + Bz^k - c\right).$$

[11] Тут ми скористалися тим, що субдиференціао субдиференційовної і диференційовної функції з областю визначення $\mathbb{R}^n$ є сумою субдиференціалу і градієнту; див. [3, §23] для доведення.

Оскільки $y^{k+1} = y^k + \rho r^{k+1}$, то ми можемо підставити $y^k = y^{k+1} - \rho r^{k+1}$ і переставити доданки щоб отримати

$$0 \in \partial f\left(x^{k+1}\right) + A^T\left(y^{k+1} - \rho B\left(z^{k+1} - z^k\right)\right).$$

Це означає, що x^{k+1} мінімізує

$$f(x) + \left(y^{k+1} - \rho B\left(z^{k+1} - z^k\right)\right)^T Ax.$$

Аналоігчні міркування показують, що z^{k+1} мінімізує $g(z) + \left(y^{k+1}\right)^T Bz$. Звідси випливає, що

$$f\left(x^{k+1}\right) + \left(y^{k+1} - \rho B\left(z^{k+1} - z^k\right)\right)^T Ax^{k+1} \leq f\left(x^\star\right) + \left(y^{k+1} - \rho B\left(z^{k+1} - z^k\right)\right)^T Ax^\star,$$

а також що

$$g\left(z^{k+1}\right) + \left(y^{k+1}\right)^T Bz^{k+1} \leq g\left(z^\star\right) + \left(y^{k+1}\right)^T Bz^\star.$$

Додаючи дві нерівності вище, використовуючи рівність $Ax^\star + Bz^\star = c$, і переставляючи доданки, отримуємо (A.2) □

Доведення нерівності (A.1)

Доведення. Додаючи (A.2) і (A.3), перегруповуючи доданки, і домножаючи на 2 отримуємо

$$2\left(y^{k+1} - y^\star\right)^T r^{k+1} - 2\rho\left(B\left(z^{k+1} - z^k\right)\right)^T r^{k+1} + 2\rho\left(B\left(z^{k+1} - z^k\right)\right)^T\left(B\left(z^{k+1} - z^\star\right)\right) \leq 0. \quad \text{(A.4)}$$

Ми отримємо (A.1) з цієї нерівності після певних маніпуляцій.

Почнемо з переписування першого доданку. Робимо заміну $y^{k+1} = y^k + \rho r^{k+1}$, отримуємо

$$2\left(y^k - y^\star\right)^T r^{k+1} + \rho\left\|r^{k+1}\right\|_2^2 + \rho\left\|r^{k+1}\right\|_2^2,$$

а подальша заміну $r^{k+1} = (1/\rho)\left(y^{k+1} - y^k\right)$ у перших двох доданках дає

$$(2/\rho)\left(y^k - y^\star\right)^T\left(y^{k+1} - y^k\right) + (1/\rho)\left\|y^{k+1} - y^k\right\|_2^2 + \rho\left\|r^{k+1}\right\|_2^2.$$

Оскільки $y^{k+1} - y^k = \left(y^{k+1} - y^\star\right) - \left(y^k - y^\star\right)$, то можемо переписати останній вираз у вигляді

$$(1/\rho)\left(\left\|y^{k+1} - y^\star\right\|_2^2 - \left\|y^k - y^\star\right\|_2^2\right) + \rho\left\|r^{k+1}\right\|_2^2. \quad \text{(A.5)}$$

Тепер переписуємо решту доданків:

$$\rho\left\|r^{k+1}\right\|_2^2 - 2\rho\left(B\left(z^{k+1} - z^k\right)\right)^T r^{k+1} + 2\rho\left(B\left(z^{k+1} - z^\star\right)\right),$$

де $\rho\left\|r^{k+1}\right\|_2^2$ взяли з (A.5). Підставляючи

$$z^{k+1} - z^\star = \left(z^{k+1} - z^k\right) + \left(z^k - z^\star\right)$$

у останній доданок отримуємо

$$\rho \left\| r^{k+1} - B\left(z^{k+1} - z^k\right)\right\|_2^2 + \rho \left\| B\left(z^{k+1} - z^k\right)\right\|_2^2 + 2\rho \left(B\left(z^{k+1} - z^k\right)\right)^T \left(B\left(z^k - z^\star\right)\right),$$

а підставляючи

$$z^{k+1} - z^k = \left(z^{k+1} - z^\star\right) - \left(z^k - z^\star\right)$$

у останні два доданки, отримуємо

$$\rho \left\| r^{k+1} - B\left(z^{k+1} - z^k\right)\right\|_2^2 + \rho \left(\left\| B\left(z^{k+1} - z^\star\right)\right\|_2^2 - \left\| B\left(z^k - z^\star\right)\right\|_2^2 \right).$$

Разом з попереднім кроком це означає, що (A.4) можна записати у вигляді

$$V^k - V^{K+1} \geq \rho \left\| r^{k+1} - B\left(z^{k+1} - z^k\right)\right\|_2^2. \tag{A.6}$$

Щоб довести (A.1) залишилося всього лише показати, що середній доданок

$$-2\rho \left(r^{k+1}\right)^T \left(B\left(z^{k+1} - z^k\right)\right)$$

з розгорнутої правої частини (A.6) додатній.

Для того щоб показати це, загадаємо, що z^{k+1} мініммізує $g(z) + \left(y^{k+1}\right)^T Bz$, а z^k мінімізує $g(z) + \left(y^k\right)^T Bz$. Додаючи відповідні нерівності, а саме

$$g\left(z^{k+1}\right) + \left(y^{k+1}\right)^T Bz^{k+1} \leq g\left(z^k\right) + \left(y^{k+1}\right)^T Bz^k,$$

і

$$g\left(z^k\right) + \left(y^k\right)^T Bz^k \leq g\left(z^{k+1}\right) + \left(y^k\right)^T Bz^{k+1},$$

маємо

$$\left(y^{k+1} - y^k\right)^T \left(B\left(z^{k+1} - z^k\right)\right) \leq 0.$$

Підставляючи $y^{k+1} - y^k = \rho r^{k+1}$ отримуємо жаданий результат, адже $\rho > 0$. □

Література

[1] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE Intelligent Systems*, vol. 24, no. 2, 2009.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[3] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[4] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*. Springer-Verlag, 1985.

[5] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, pp. 101–111, 1960.

[6] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, pp. 238–252, 1962.

[7] G. B. Dantzig, *Linear Programming and Extensions*. RAND Corporation, 1963.

[8] H. Everett, "Generalized lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, no. 3, pp. 399–417, 1963.

[9] L. S. Lasdon, *Optimization Theory for Large Systems*. MacMillan, 1970.

[10] A. M. Geoffrion, "Generalized benders decomposition," *Journal of Optimization Theory and Applications*, vol. 10, no. 4, pp. 237–260, 1972.

[11] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, 1973.

[12] A. Bensoussan, J.-L. Lions, and R. Temam, *Sur les methodes de decomposition, de decentralisation et de coordination et applications*, pp. 133–257. Methodes Mathematiques de l'Informatique, 1976.

[13] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2 ed., 1999.

[14] A. Nedic and A. Ozdaglar, *Cooperative distributed multi-agent optimization in Convex Optimization in Signal Processing and Communications*. Cambridge University Press, 2010.

[15] I. Necoara and J. A. K. Suykens, "Application of a smoothing technique to decomposition in convex optimization," *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2674–2679, 2008.

[16] J. N. Tsitsiklis, *Problems in decentralized decision making and computation. PhD thesis*. Massachusetts Institute of Technology, 1984.

[17] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[18] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989.

[19] Y. Censor and S. A. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, 1997.

[20] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[21] J. C. Duchi, A. Agarwal, and M. J. Wainwright, *Distributed Dual Averaging in Networks*. Advances in Neural Information Processing Systems, 2010.

[22] M. R. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, vol. 4, pp. 302–320, 1969.

[23] M. R. Hestenes, *Multiplier and gradient methods, in Computing Methods in Optimization Problems.* Academic Press, 1969.

[24] M. J. D. Powell, *A method for nonlinear constraints in minimization problems.* Academic Press, 1969.

[25] A. Miele, E. E. Cragg, R. R. Iver, and A. V. Levy, "Use of the augmented penalty function in mathematical programming problems, part 1," *Journal of Optimization Theory and Applications*, vol. 8, pp. 115–130, 1971.

[26] A. Miele, E. E. Cragg, and A. V. Levy, "Use of the augmented penalty function in mathematical programming problems, part 2," *Journal of Optimization Theory and Applications*, vol. 8, pp. 131–153, 1971.

[27] A. Miele, P. E. Mosely, A. V. Levy, and G. M. Coggins, "On the method of multipliers for mathematical programming problems," *Journal of Optimization Theory and Applications*, vol. 10, pp. 1–33, 1972.

[28] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods.* Academic Press, 1982.

[29] K. J. Arrow and R. M. Solow, *Gradient methods for constrained maxima, with weakened assumptions.* Stanford University Press, 1958.

[30] K. J. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Linear and Nonlinear Programming.* Stanford University Press, 1958.

[31] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* Society for Industrial and Applied Mathematics, 1990.

[32] G. H. Golub and C. F. van Loan, *Matrix Computations.* Johns Hopkins University Press, 3 ed., 1996.

[33] J. Eckstein and M. C. Ferris, "Operator-splitting methods for monotone affine variational inequalities, with a parallel application to optimal control," *INFORMS Journal on Computing*, vol. 10, pp. 218–235, 1998.

[34] J. Borwein and A. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples.* Canadian Mathematical Society, 2000.

[35] J.-B. Hiriart-Urruty and C. Lemarechal, *Fundamentals of Convex Analysis.* Springer, 2001.

[36] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 14, p. 877, 1976.

[37] B. S. He, H. Yang, and S. L. Wang, "Alternating direction method with selfadaptive penalty parameters for monotone variational inequalities," *Journal of Optimization Theory and Applications*, vol. 106, no. 2, pp. 337–356, 2000.

[38] S. L. Wang and L. Z. Liao, "Decomposition method with a variable parameter for a class of monotone variational inequality problems," *Journal of Optimization Theory and Applications*, vol. 109, no. 2, pp. 415–429, 2001.

[39] J. Eckstein and D. P. Bertsekas, "On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.

[40] J. Eckstein, "Parallel alternating direction multiplier decomposition of convex programs," *Journal of Optimization Theory and Applications*, vol. 80, no. 1, pp. 39–62, 1994.

[41] E. G. Gol'shtein and N. V. Tret'yakov, "Modified lagrangians in convex programming and their generalizations," *Point-to-Set Maps and Mathematical Programming*, pp. 86–97, 1979.

[42] A. Ruszczynski, "An augmented lagrangian decomposition method for block diagonal linear programming problems," *Operations Research Letters*, vol. 8, no. 5, pp. 287–294, 1989.

[43] M. Fukushima, "Application of the alternating direction method of multipliers to separable convex programming problems," *Computational Optimization and Applications*, vol. 1, pp. 93–111, 1992.

[44] J. Eckstein and M. Fukushima, "Some reformulations and applications of the alternating direction method of multipliers," *Large Scale Optimization: State of the Art*, pp. 119–138, 1993.

[45] J. Eckstein, *Splitting methods for monotone operators with applications to parallel optimization*. PhD thesis, MIT, 1989.

[46] R. T. Rockafellar, "Augmented lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of Operations Research*, vol. 1, pp. 97–116, 1976.

[47] G. Chen and M. Teboulle, "A proximal-based decomposition method for convex minimization problems," *Mathematical Programming*, vol. 64, pp. 81–101, 1994.

[48] J. Eckstein, "Some saddle-function splitting methods for convex programming," *Optimization Methods and Software*, vol. 4, no. 1, pp. 75–83, 1994.

[49] J. Eckstein and D. P. Bertsekas, *An alternating direction method for linear programming*. Tech. Rep., MIT, 1990.

[50] R. T. Rockafellar and R. J.-B. Wets, "Scenarios and policy aggregation in optimization under uncertainty," *Mathematics of Operations Research*, vol. 16, no. 1, pp. 119–147, 1991.

[51] A. Ruszczynski, "On convergence of an augmented lagrangian decomposition method for sparse convex optimization," *Mathematics of Operations Research*, vol. 20, no. 3, pp. 634–656, 1995.

[52] P. Tseng, "Alternating projection-proximal methods for convex programming and variational inequalities," *SIAM Journal on Optimization*, vol. 7, pp. 951–965, 1997.

[53] P. Tseng, "A modified forward-backward splitting method for maximal monotone mappings," *SIAM Journal on Control and Optimization*, vol. 38, p. 431, 2000.

[54] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forwardbackward splitting," *Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1168–1200, 2006.

[55] R. Glowinski and A. Marrocco, "Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualite, d'une classe de problems de dirichlet no.lineares," *Revue Francaise d'Automatique, Informatique, et Recherche Operationelle*, vol. 9, pp. 41–76, 1975.

[56] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximations," *Computers and Mathematics with Applications*, vol. 2, pp. 17–40, 1976.

[57] M. Fortin and R. Glowinski, *On decomposition-coordination methods using an augmented Lagrangian in Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems*. North-Holland, 1983.

[58] D. Gabay, *Applications of the method of multipliers to variational inequalities in Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems*. North-Holland, 1983.

[59] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*. North-Holland, 1983.

[60] R. Glowinski and P. L. Tallec, *Augmented Lagrangian methods for the solution of variational problems*. Tech. Rep. 2965, University of Wisconsin-Madison, 1987.

[61] P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM Journal on Control and Optimization*, vol. 29, pp. 119–138, 1991.

[62] J. M. Bioucas-Dias and M. A. T. Figueiredo, *Alternating Direction Algorithms for Constrained Sparse Regression: Application to Hyperspectral Unmixing*. ??, 2010.

[63] M. J. Fadili and J. L. Starck, "Monotone operator splitting for optimization problems in sparse recovery," *IEEE ICIP*, 2009.

[64] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of poissonian images using alternating direction optimization," *IEEE Transactions on Image Processing*, vol. 19, pp. 3133–3145, 2010.

[65] G. Steidl and T. Teuber, "Removing multiplicative noise by douglas-rachford, splitting methods," *Journal of Mathematical Imaging and Vision*, vol. 36, no. 2, pp. 168–184, 2010.

[66] M. Ng, P. Weiss, and X. Yuang, "Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods," *ICM Research Report*, 2009.

[67] J. Yang and X. Yuan, "An inexact alternating direction method for trace norm regularized least squares problem," *??*, 2010.

[68] X. M. Yuan, "Alternating direction methods for sparse covariance selection," *??*, 2009.

[69] Z. Lu, T. K. Pong, and Y. Zhang, "An alternating direction method for finding dantzig selectors," *??*, 2010.

[70] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *Journal of Machine Learning Research*, vol. 11, pp. 1663–1707, 2010.

[71] P. L. Combettes and J. C. Pesquet, "A douglas-rachford splitting approach to nonsmooth convex variational signal recovery," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 564–574, 2007.

[72] P. L. Combettes and J. C. Pesquet, *Proximal Splitting Methods in Signal Processing*. ??, 2009.

[73] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc wsns with noisy links-part i: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, pp. 350–364, 2008.

[74] I. D. Schizas, G. Giannakis, S. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc wsns with noisy links-part ii: Distributed estimation and smoothing of random signals," *IEEE Transactions on Signal Processing*, vol. 56, pp. 1650–1666, 2008.

[75] H. Brezis, *Operateurs Maximaux Monotones et Semi-Groupes de Contractions dans les Espaces de Hilbert*. North-Holland, 1973.

[76] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Springer-Verlag, 1998.

[77] J. Douglas and H. H. Rachford, *On the numerical solution of heat conduction problems in two and three space variables*, vol. 82. Transactions of the American Mathematical Society, 1956.

[78] P. L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM Journal on Numerical Analysis*, vol. 16, pp. 964–979, 1979.

[79] D. W. Peaceman and H. H. Rachford, "The numerical solution of parabolic and elliptic differential equations," *Journal of the Society for Industrial and Applied Mathematics*, vol. 3, pp. 28–41, 1955.

[80] J. E. Spingarn, "Applications of the method of partial inverses to convex programming: decomposition," *Mathematical Programming*, vol. 32, pp. 199–223, 1985.

[81] J. Lawrence and J. E. Spingarn, "Proceedings of the london mathematical society," in *On fixed points of non-expansive piecewise isometric mappings*, vol. 3, p. 605, 1987.

[82] J. Eckstein and B. F. Svaiter, "A family of projective splitting methods for the sum of two maximal monotone operators," *Mathematical Programming*, vol. 111, no. 1-2, p. 173, 2008.

[83] J. Eckstein and B. F. Svaiter, "General projective splitting methods for sums of maximal monotone operators," *SIAM Journal on Control and Optimization*, vol. 48, pp. 787–811, 2009.

[84] Y. Censor and S. A. Zenios, "Proximal minimization algorithm with dfunctions," *Journal of Optimization Theory and Applications*, vol. 73, no. 3, pp. 451–464, 1992.

[85] J. Eckstein, *Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming*, pp. 202–226. Mathematics of Operations Research, 1993.

[86] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 3, pp. 200–217, 1967.

[87] J.-J. Moreau, "Reports of the paris academy of sciences," in *Fonctions convexes duales et points proximaux dans un espace Hilbertien*, vol. 255 of *A*, pp. 2897–2899, 1962.

[88] J. W. Demmel, *Applied Numerical Linear Algebra*. SIAM, 1997.

[89] D. C. Liu and J. Nocedal, "On the limited memory method for large scale optimization," *Mathematical Programming B*, vol. 45, no. 3, pp. 503–528, 1989.

[90] R. H. Byrd, P. Lu, and J. Nocedal, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific and Statistical Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[91] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer-Verlag, 1999.

[92] J. von Neumann, *Functional Operators*, vol. 2. Princeton University Press: Annals of Mathematics Studies, 1950.

[93] W. Cheney and A. A. Goldstein, "Proximity maps for convex sets," *Proceedings of the American Mathematical Society*, vol. 10, no. 3, pp. 448–450, 1959.

[94] L. M. Bregman, "Finding the common point of convex sets by the method of successive projections," *Proceedings of the USSR Academy of Sciences*, vol. 162, no. 3, pp. 487–490, 1965.

[95] R. L. Dykstra, "An algorithm for restricted least squares regression," *Journal of the American Statistical Association*, vol. 78, pp. 837–842, 1983.

[96] H. H. Bauschke and J. M. Borwein, "Dykstra's alternating projection algorithm for two sets," *Journal of Approximation Theory*, vol. 79, no. 3, pp. 418–443, 1994.

[97] H. H. Bauschke and J. M. Borwein, "On projection algorithms for solving convex feasibility problems," *SIAM Review*, vol. 38, no. 3, pp. 367–426, 1996.

[98] P. L. Combettes, "The convex feasibility problem in image recovery," *Advances in Imaging and Electron Physics*, vol. 95, pp. 155–270, 1996.

[99] P. J. Huber, "Robust estimation of a location parameter," *Annals of Mathematical Statistics*, vol. 35, pp. 73–101, 1964.

[100] J. F. Claerbout and F. Muir, "Robust modeling with erratic data," *Geophysics*, vol. 38, p. 826, 1973.

[101] L. Rudin, S. J. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, vol. 60, pp. 259–268. Physica D, 1992.

[102] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, pp. 613–627, 1995.

[103] R. Tibshirani, *Regression shrinkage and selection via the lasso*, vol. 58, pp. 267–288. Journal of the Royal Statistical Society, 1996.

[104] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, pp. 129–159, 2001.

[105] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[106] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, p. 489, 2006.

[107] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[108] N. Meinshausen and P. Buhlmann, "High-dimensional graphs and variable selection with the lasso," *Annals of Statistics*, vol. 34, no. 3, pp. 1436–1462, 2006.

[109] A. Y. Yang, A. Ganesh, Z. Zhou, S. S. Sastry, and Y. Ma, "A review of fast ℓ_1-minimization algorithms for robust face recognition," *??*, 2010.

[110] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.

[111] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale ℓ_1-regularized least squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, 2007.

[112] D. L. Donoho and Y. Tsaig, *Fast solution of ℓ_1-norm minimization problems when the solution may be sparse.* Tech. Rep., Stanford University, 2006.

[113] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, pp. 1413–1457, 2004.

[114] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $o(1/k^2)$," *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376, 1983.

[115] Y. Nesterov, *Gradient methods for minimizing composite objective function*, vol. 76, p. 2007. CORE Discussion Paper, Catholic University of Louvain, 2007.

[116] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[117] S. Becker, J. Bobin, and E. J. Candes, "Nesta: A fast and accurate firstorder method for sparse recovery," *??*, 2009.

[118] J. Yang and Y. Zhang, "Alternating direction algorithms for ℓ_1-problems in compresed sensing," *??*, 2009.

[119] K. Koh, S.-J. Kim, and S. Boyd, "An interior-point method for large-scale ℓ_1-regularized logistic regression," *Journal of Machine Learning Research*, vol. 1, no. 8, pp. 1519–1555, 2007.

[120] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing," *SIAM Journal on Imaging Sciences*, vol. 1, no. 1, pp. 143–168, 2008.

[121] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, p. 18914, 2009.

[122] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer, 2 ed., 2009.

[123] J. M. Wooldridge, *Introductory Econometrics: A Modern Approach.* South Western College Publications, 2009.

[124] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[125] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Transactions on Image Processing*, vol. 20, pp. 681–695, 2011.

[126] T. Goldstein and S. Osher, "The split bregman method for ℓ_1 regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[127] E. Esser, "Applications of lagrangian-based alternating direction methods and connections to split bregman," *CAM report*, vol. 9, p. 31, 2009.

[128] P. J. McCullagh and J. A. Nelder, *Generalized Linear Models*. Chapman & Hall, 1991.

[129] T. Hastie and R. Tibshirani, *Generalized Additive Models*. Chapman & Hall, 1990.

[130] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2345–2356, 2010.

[131] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "ℓ_1 trend filtering," *SIAM Review*, vol. 51, no. 2, pp. 339–360, 2009.

[132] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[133] H. Ohlsson, L. Ljung, and S. Boyd, "Segmentation of arx-models using sum-of-norms regularization," *Automatica*, vol. 46, pp. 1107–1111, 2010.

[134] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *Annals of Statistics*, vol. 37, no. 6, pp. 3468–3497, 2009.

[135] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach, "Network flow algorithms for structured sparsity," *Advances in Neural Information Processing Systems*, vol. 24, 2010.

[136] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[137] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data," *Journal of Machine Learning Research*, vol. 9, pp. 485–516, 2008.

[138] A. P. Dempster, "Covariance selection," *Biometrics*, vol. 28, no. 1, pp. 157–175, 1972.

[139] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, p. 432, 2008.

[140] J. C. Duchi, S. Gould, and D. Koller, "Projected subgradient methods for learning sparse gaussians," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2008.

[141] Z. Lu, "Smooth optimization approach for sparse covariance selection," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1807–1827, 2009.

[142] K. Scheinberg, S. Ma, and D. Goldfarb, "Sparse inverse covariance selection via alternating linearization methods," *Advances in Neural Information Processing Systems*, 2010.