УДК 519.854.3

В.М. КРИГІН
Міжнародний науково-навчальний центр інформаційних технологій та систем НАН України та МОН України, Київ, Україна,
e-mail: *valeriy.krygin@gmail.com*.

Р.О. ХОМЕНКО
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна,
e-mail: *ruslank3584@gmail.com*.

АЛГОРИТМ РОЗВ'ЯЗУВАННЯ СУПЕРМОДУЛЯРНИХ $(\max,+)$ ЗАДАЧ РОЗМІТКИ ІЗ САМОКОНТРОЛЕМ НА ОСНОВІ СУБГРАДІЄНТНОГО СПУСКУ[1]

Анотація. Розглянуто алгоритм, який для будь-якої поданої на вхід (max,+) задачі розмітки з цілочисельними вагами надасть на вихід одну з двох відповідей: або розв'язок у формі оптимальної розмітки, або фразу «задача не є супермодулярною», при цьому будь-яка відповідь гарантовано буде коректною. Самоконтроль у розпізнаванні образів полягає у тому, що не користувач приймає рішення, на яке питання треба відповісти, а сам алгоритм вирішує, що потрапляє у зону його компетентності. Іншою особливістю алгоритму є те, що він не потребує відомої впорядкованості міток для супермодулярних задач. Гарантію скінченної кількості кроків забезпечує використання субградієнтного спуску і цілочисельність ваг вершин та ребер.

Ключові слова: (max,+) задачі розмітки, супермодулярні задачі розмітки, самоконтроль у розпізнаванні образів, дискретна оптимізація, графові моделі, структурне розпізнавання образів.

Задачі розмітки відіграють важливу роль у структурному розпізнаванні зображень та мають багато застосувань [1–6]. Одним з важливих класів задач розмітки, які мають ефективний розв'язок, є клас супермодулярних $(\max,+)$ задач з відомою впорядкованістю [7] і з невідомою впорядкованістю [8] міток.

У цій роботі наведено алгоритм розв'язування супермодулярних $(\max,+)$ задач розмітки із самоконтролем. Це алгоритм, який на вхід приймає будь-яку $(\max,+)$ задачу розмітки з цілочисельними вагами, а на вихід надає або розв'язок, або відповідь «задача не є супермодулярною». Є відомі алгоритми із самоконтролем, які застосовуються для розв'язування задач лінійної класифікації, інваріантних відносно оператора напівґратки $(\vee,\wedge)$ задач розмітки [9], інваріантних відносно мажоритарного оператора (min, max) задач розмітки [10], та більш широкого класу задач розмітки, інваріантних відносно мажоритарного оператора [11], а також для супермодулярних (max,+) задач розмітки з цілочисельними вагами вершин та ребер [8]. Головна відмінність цієї роботи від [8] полягає у використанні субградієнтного спуску замість алгоритму дифузії. Вибір алгоритму обґрунтовано властивостями, описаними у [12, 13]. До того ж, наведений алгоритм для задачі, заданої множиною T об'єктів та множиною K міток, потребує не більше ніж $|T|\cdot\log_2|K|+1$ викликів процедури оптимізації на відміну від $|T|\cdot|K|$ викликів, зазначених у [8].

ПОСТАНОВКА $(\max,+)$ ЗАДАЧІ РОЗМІТКИ

Задано скінченну непорожню множину T об'єктів і скінченну непорожню множину K міток. На множині T об'єктів визначено структуру сусідства $\Gamma\subset T^2$, яка є асиметричною: $(t,t')\in\Gamma \Rightarrow (t',t)\notin\Gamma$. Далі замість запису (t,t') буде використовуватися запис tt'. Множину усіх сусідів об'єкта t позначимо

[1] Роботу виконано у межах теми «Створення інтелектуальних інформаційних технологій на базі методів і засобів образного мислення», державний реєстраційний номер 0114U002068.

$$N_t = \{t' : tt' \in \Gamma \cup \Gamma^{-1}\},$$

де $\Gamma^{-1} = \{t'\,t : tt' \in \Gamma\}$ — інверсія відношення Г. Функцію $k: T \to K$ називатимемо розміткою. Те, що об'єкту $t \in T$ розмітка k ставить у відповідність мітку $\ell \in K$, позначимо $k_t = \ell$.

Пару $(t, k) \in T \times K$ називатимемо вершиною, а пару $((t, k), (t', k'))$ вершин, де $k \in K$, $k' \in K$ і $tt' \in \Gamma$, називатимемо ребром. Позначимо цілочисельну функцію $q: T \times K \to \mathbb{Z}$ ваг вершин, цілочисельну функцію $g: \Gamma \times K^2 \to \mathbb{Z}$ ваг ребер та функцію $G: K^T \to \mathbb{Z}$

$$G(k) = \sum_{t \in T} q_t(k_t) + \sum_{tt' \in \Gamma} g_{tt'}(k_t, k_{t'}) .$$

Задача розмітки полягає у пошуку такої розмітки $k^*: T \to K$, за якої функція G має найбільше значення

$$k^* \in \arg\max_{k: T \to K} G(k) .$$

Задачу розмітки називають супермодулярною, якщо існує таке відношення $n: T \times K \to \{1, 2, \ldots, |K|\}$, що для будь-яких $k \in K$, $k' \in K$, $\ell \in K$, $\ell' \in K$ і $tt' \in \Gamma$ з умов $n_t(k) \ge n_t(\ell)$ і $n_{t'}(k') \ge n_{t'}(\ell')$ випливає

$$g_{tt'}(k, k') + g_{tt'}(\ell, \ell') \ge g_{tt'}(k, \ell') + g_{tt'}(\ell, k') .$$

Ребра $((t, k), (t', k'))$ і $((t, \ell), (t', \ell'))$ називають паралельними, а ребра $((t, k), (t', \ell'))$ і $((t, \ell), (t', k'))$ називають перехресними, тому наведену нерівність можна інтерпретувати так: сума ваг паралельних ребер є не гіршою за суму ваг перехресних ребер. На функцію q обмеження не накладають.

Якщо задача є супермодулярною і відображення n є відомим, задачу називають супермодулярною з відомою впорядкованістю. Коли відомо, що існує таке n, за якого задача є супермодулярною, проте саме відображення n є невідомим, задачу називають супермодулярною з невідомою впорядкованістю.

Існують алгоритми, які знаходять точний розв'язок будь-якої супермодулярної (max,+) задачі розмітки, якщо впорядкованість множини міток на кожному об'єкті є відомою [1, 2, 7, 14] або коли впорядкованість є невідомою, проте відомо, що задача є супермодулярною [8]. У цій роботі наведено алгоритм розв'язування супермодулярних (max,+) задач з цілими вагами вершин та ребер, для яких упорядкованість множини міток може бути невідомою. До того ж, якщо на вхід цьому алгоритму подати задачу, що не є супермодулярною, він може або розв'язати її, або відмовитися від розв'язування, проте його відповідь завжди буде правильною: або найкраща розмітка k^*, або відмова з результатом «задача не є супермодулярною».

Зауважимо, що обмеження на цілочисельність ваг g та q не є суттєвим на практиці, адже числа на обчислювальній техніці мають граничну точність, отже належать множині раціональних чисел. За таких умов будь-яку практичну (max,+) задачу розмітки з раціональними вагами можна перетворити на задачу з цілочисельними вагами, поділивши усі ваги на найбільший спільний дільник усіх значень g та q. Найбільшим спільним дільником набору раціональних чисел є найбільше число, яке ділить націло кожне число з цього набору.

ВЛАСТИВОСТІ ε-УЗГОДЖЕНОГО НАБОРУ ВЕРШИН ТА РЕБЕР

Введемо функцію $\varphi: (\Gamma \cup \Gamma^{-1}) \times K \to \mathbb{R}$ та позначимо

$$q_t^{\varphi}(k_t) = q_t(k_t) + \sum_{t' \in N_t} \varphi_{tt'}(k_t),$$

$$g_{tt'}^{\varphi}(k_t, k_{t'}) = g_{tt'}(k_t, k_{t'}) - \varphi_{tt'}(k_t) - \varphi_{t't}(k_{t'}).$$

Введемо функцію

$$E(q, g, \varphi) = \sum_{t \in T} \max_{\ell \in K} q_t^{\varphi}(\ell) + \sum_{tt' \in \Gamma} \max_{\substack{\ell \in K \\ \ell' \in K}} g_{tt'}^{\varphi}(\ell, \ell').$$

Для супермодулярних задач відомо [3, 6, 15, 16], що

$$\min_{\varphi} E(q, g, \varphi) = \max_{k: T \to K} G(k).$$

Функція E є опуклою за аргументом φ, тому її розв'язок можна шукати, зокрема, за допомогою субградієнтного спуску [6, 12].

Позначимо підмножину $V \subseteq T \times K$ вершин і підмножину $A \subset (T \times K)^2$ ребер. Набір (V, A) називають ε-узгодженим набором вершин та ребер, якщо для вибраного значення $\varepsilon \in \mathbb{R}_+$ до $V \neq \varnothing$ і $A \neq \varnothing$ входять лише ті вершини та ребра, що відрізняються від кращих не більше ніж на ε:

$$\max_{\ell \in K} q_t^{\varphi}(\ell) - q_t^{\varphi}(k) \le \varepsilon, \forall (t, k) \in V,$$

$$\max_{\substack{\ell \in K \\ \ell' \in K}} g_{tt'}^{\varphi}(\ell, \ell') - g_{tt'}^{\varphi}(k, k') \le \varepsilon, \ \forall ((t, k), (t', k')) \in A,$$

кожне ребро з A спирається на вершини з набору V

$$((t, k), (t', k')) \in A \Rightarrow \{(t, k), (t', k')\} \subseteq V,$$

а для кожної вершини $(t, k) \in V$ і кожного об'єкта $t' \in N_t$ знайдеться хоча б одне ребро з A, яке з'єднує вершину (t, k) з вершиною, що знаходиться в об'єкті t':

$$(t, k) \in V \Rightarrow \forall t' \in N_t \ \exists k' \in K\colon ((t, k), (t', k')) \in A \cup A^{-1},$$

де $A^{-1} = \{((t', k'), (t, k))\colon ((t, k), (t', k')) \in A\}$ є множиною обернених ребер. Для пошуку цього набору можна застосувати алгоритм викреслювання другого порядку [6, 15, 17, 18]. Зауважимо, що результатом роботи алгоритму можуть бути пусті множини A і V — у цьому випадку кажуть, що ε-узгодженого набору вершин і ребер для даного φ не існує.

Якщо для супермодулярної задачі з упорядкованістю n (яка може бути невідомою) знайшовся ε-узгоджений набір вершин та ребер, для кожної пари сусідніх об'єктів $tt' \in \Gamma$ можна обрати таку пару ребер $((t, k_t), (t', \ell_{t'})) \in A$ і $((t, \ell_t), (t', k_{t'})) \in A$ (необов'язково різних), що $n_t(k_t) \ge n_t(\ell_t)$ і $n_{t'}(k_{t'}) \ge n_{t'}(\ell_{t'})$, для яких справджується нерівність

$$g_{tt'}(k_t, k_{t'}) + g_{tt'}(\ell_t, \ell_{t'}) \ge g_{tt'}(k_t, \ell_{t'}) + g_{tt'}(\ell_t, k_{t'}).$$

З визначення ε-узгодженості маємо нерівності

$$g_{tt'}(k_t, \ell_{t'}) \ge \max_{\substack{\ell \in K \\ \ell' \in K}} g_{tt'}(\ell, \ell') - \varepsilon,$$

$$g_{tt'}(\ell_t, k_{t'}) \ge \max_{\substack{\ell \in K \\ \ell' \in K}} g_{tt'}(\ell, \ell') - \varepsilon,$$

отже

$$g_{tt'}(k_t, k_{t'}) + g_{tt'}(\ell_t, \ell_{t'}) \ge g_{tt'}(k_t, \ell_{t'}) + g_{tt'}(\ell_t, k_{t'}) \ge 2 \cdot \max_{\substack{\ell \in K \\ \ell' \in K}} g_{tt'}(\ell, \ell') - 2 \cdot \varepsilon.$$

З визначення максимального елементу

$$-g_{tt'}(\ell_t, \ell_{t'}) \ge -\max_{\substack{\ell \in K \\ \ell' \in K}} g_{tt'}(\ell, \ell'),$$

тому

$$g_{tt'}(k_t, k_{t'}) \ge \max_{\substack{\ell \in K \\ \ell' \in K}} g_{tt'}(\ell, \ell') - 2 \cdot \varepsilon.$$

Зауважимо, що за умови

$$\max_{\substack{\ell\in K\\ \ell'\in K}} g_{tt'}(\ell,\ell') - \varepsilon > g_{tt'}(k_t, k_{t'}) \geq \max_{\substack{\ell\in K\\ \ell'\in K}} g_{tt'}(\ell,\ell') - 2\cdot\varepsilon$$

ребро $((t, k_t), (t', k_{t'}))$ не належить ε-узгодженому набору A ребер, проте може належати оптимальній розмітці.

Якщо задача є супермодулярною і знайшовся ε-узгоджений набір вершин та ребер, існує така розмітка $k\colon T \to K$, для якої справджується вираз

$$\sum_{t\in T} q_t(k_t) + \sum_{tt'\in\Gamma} g_{tt'}(k_t, k_{t'}) = \sum_{t\in T} q_t^{\varphi}(k_t) + \sum_{tt'\in\Gamma} g_{tt'}^{\varphi}(k_t, k_{t'}) \geq$$

$$\geq \sum_{t\in T} (\max_{\ell\in K} q_t^{\varphi}(\ell) - \varepsilon) + \sum_{tt'\in\Gamma} \max_{\substack{\ell\in K\\ \ell'\in K}} (g_{tt'}^{\varphi}(\ell,\ell') - 2\cdot\varepsilon) =$$

$$= \sum_{t\in T} \max_{\ell\in K} q_t^{\varphi}(\ell) - |T|\cdot\varepsilon + \sum_{tt'\in\Gamma} \max_{\substack{\ell\in K\\ \ell'\in K}} g_{tt'}^{\varphi}(\ell,\ell') - 2\cdot|\Gamma|\cdot\varepsilon =$$

$$= \sum_{t\in T} \max_{\ell\in K} q_t^{\varphi}(\ell) + \sum_{tt'\in\Gamma} \max_{\substack{\ell\in K\\ \ell'\in K}} g_{tt'}^{\varphi}(\ell,\ell') - (|T| + 2\cdot|\Gamma|)\cdot\varepsilon.$$

Найкраща розмітка k^* не може мати більшу вагу ніж сума найбільших значень q та g. Це означає, що за умови

$$(|T| + 2\cdot|\Gamma|)\cdot\varepsilon < 1$$

вага розмітки k дорівнює вазі найкращої розмітки, бо функції q і g є цілочисельними [8].

Якщо задача є супермодулярною та має цілочисельні ваги, наведений алгоритм пошуку розмітки надасть оптимальну розмітку для

$$\varepsilon \leq \frac{1}{|T| + 2\cdot|\Gamma| + 1}.$$

ПОШУК РОЗМІТКИ

Візьмемо вагові функції q і g і константу $\varepsilon \in \mathbb{R}_+$. Застосуємо субградієнтний спуск до функції E за аргументом φ та отримаємо послідовність значень аргументу для кожного кроку субградієнтного спуску. Зупинимо роботу субградієнтного спуску на найменшому номері $i \in \mathbb{N}_0$ кроку, за якого для q^{φ^i} і g^{φ^i} існує ε-узгоджений набір вершин та ребер. Результат застосування цієї процедури для вибраних q, g і ε позначимо

$$\phi(q, g; \varepsilon) = \varphi^i.$$

Такий номер i обов'язково знайдеться, адже відомо, що для будь-якого $\varepsilon \in \mathbb{R}_+$ субградієнтний спуск за скінченну кількість $i \in \mathbb{N}_0$ кроків дійде до такого значення φ^i, за якого існує ε-узгоджений набір вершин та ребер [13].

Пошук однієї з найкращих розміток здійснюють у такий спосіб.

1. Введемо функцію $q^0 \equiv q$ та номер кроку $j = 0$.

2. Якщо кожен об'єкт $t \in T$ має лише по одній мітці зі скінченною вагою, алгоритм завершується, а розмітку k^* будують у такий спосіб, що

$$q_t^j(k_t^*) > -\infty \quad \forall t \in T.$$

3. Для довільного об'єкта $t \in T$, у якому залишилося більше однієї мітки зі скінченною вагою, фіксуємо довільну мітку $k \in K$. Позначимо функцію

$q' : T \times K \to \mathbb{R} \cup \{-\infty\}$, яка в об'єкті t забороняє будь-які мітки окрім k

$$q'_{t'}(k') = \begin{cases} -\infty, & t' = t, \quad k' \neq k, \\ q^j_{t'}(k'), & \text{інакше.} \end{cases}$$

Якщо подана на вхід задача була супермодулярною, така модифікація ваг вершин залишить задачу супермодулярною, адже визначення супермодулярності не накладає обмежень на ваги вершин.

4. Виконуємо субградієнтний спуск, доки не отримаємо ε-узгоджений набір вершин та ребер, для пошуку значення $E(q', g; \varphi(q', g, \varepsilon))$. Якщо

$$\lfloor E(q', g; \varphi(q', g, \varepsilon)) \rfloor = \lfloor E(q, g; \varphi(q, g, \varepsilon)) \rfloor,$$

переходимо на наступну ітерацію $j := j+1$, створюємо функцію $q^j \equiv q'$ (яка забороняє в об'єкті t усі мітки окрім обраної мітки k) та повертаємося до другого кроку (перевірка того, чи завершився алгоритм).

Якщо отриманий результат менший за той, що був отриманий до початку роботи алгоритму пошуку розмітки

$$\lfloor E(q', g; \varphi(q', g, \varepsilon)) \rfloor < \lfloor E(q, g; \varphi(q, g, \varepsilon)) \rfloor,$$

завершуємо алгоритм з відповіддю «задача не є супермодулярною», тому що було доведено, що у випадку супермодулярної задачі значення $E(q, g; \varphi(q, g, \varepsilon))$ дорівнює вазі $G(k^*)$ оптимальної розмітки k^*, а наведена нерівність означатиме, що під час роботи алгоритму може бути знайдена розмітка з меншою вагою. Якщо отриманий результат більший за той, що був отриманий до початку роботи алгоритму пошуку розмітки

$$\lfloor E(q', g; \varphi(q', g, \varepsilon)) \rfloor > \lfloor E(q, g; \varphi(q, g, \varepsilon)) \rfloor,$$

виконуємо третій крок алгоритму для іншої мітки k (яку не було вибрано раніше у даному об'єкті). Якщо ця нерівність виконується для всіх міток в об'єкті, завершуємо алгоритм з відповіддю «задача не є супермодулярною», тому що за умови супермодулярності задачі обов'язково можна знайти мітку, що належить оптимальній розмітці, вага якої дорівнює $\lfloor E(q, g; \varphi(q, g, \varepsilon)) \rfloor$.

Цей алгоритм потребує не більше ніж $|T| \cdot |K|$ розрахунків величин $\lfloor E(q', g; \varphi(q', g, \varepsilon)) \rfloor$ для різних q'. Наведену оцінку складності можна покращити, якщо на етапі пошуку мітки $k_t \in K$ для об'єкта $t \in T$ на кожній ітерації j використовувати двійковий пошук: розбити множину K на дві такі підмножини K_1 і K_2, що

$$K_1 \cup K_2 = K,$$
$$K_1 \cap K_2 = \varnothing,$$
$$\| K_1 | - | K_2 \| \leq 1,$$

та ввести функцію q'

$$q'_{t'}(k') = \begin{cases} -\infty, & t' = t, \; k' \in K_1, \\ q^j_{t'}(k'), & \text{інакше.} \end{cases}$$

Якщо $\lfloor E(q', g; \varphi(q', g, \varepsilon)) \rfloor = \lfloor E(q, g; \varphi(q, g, \varepsilon)) \rfloor$, шукане k може знаходитись у K_2, і його треба шукати у цій множині. Якщо $\lfloor E(q', g; \varphi(q', g, \varepsilon)) \rfloor <$ $< \lfloor E(q, g; \varphi(q, g, \varepsilon)) \rfloor$, задача не є супермодулярною. Якщо $\lfloor E(q', g; \varphi(q', g, \varepsilon)) \rfloor >$ $< \lfloor E(q, g; \varphi(q, g, \varepsilon)) \rfloor$, шукаємо відповідь у множині K_1. Пошук у вибраній множині K_i, $i \in \{1, 2\}$ виконують за тим самим принципом. Наприкінці, коли залишається лише одне k, потрібно перевірити рівність $\lfloor E(q', g; \varphi(q', g, \varepsilon)) \rfloor =$ $= \lfloor E(q, g; \varphi(q, g, \varepsilon)) \rfloor$, заборонивши всі інші мітки в об'єкті, бо коли множина K_1 не підходила, ми одразу вибирали K_2, але не перевіряли, чи вона підходить. Отже, для наведеного алгоритму пошуку розмітки потрібно не більше ніж $|T|\cdot \log_2 |K|+1$ розрахунків величин $\lfloor E(q', g; \varphi(q', g, \varepsilon))\rfloor$ для різних q'.

У загальному випадку значення функції E може як завгодно відрізнятися від розв'язку відповідної задачі розмітки для будь-яких значень φ [13]. Також відомо, що у загальному випадку (max,+) задачі розмітки належать класу складності EXP-APX — класу задач, для яких за поліноміальний від розміру вхідних даних час можна знайти наближення, похибка якого обмежена експонентою від розміру даних [19]. Проте зауважимо, що у тому разі, коли задача не є супермодулярною, проте має цілочисельні ваги і алгоритм надав розмітку k^*, ця розмітка буде оптимальною, оскільки за побудовою алгоритму ціла частина значення функції E не змінюється і дорівнює $G(k^*)$.

СПОСІБ ПЕРЕВІРКИ ОТРИМАНОГО РОЗВ'ЯЗКУ

Важливою властивістю отриманого розв'язку є те, що його коректність можна перевірити за скінченний час. Функція φ є так званим сертифікатом [20] цієї задачі. Це набір чисел, розмір якого має поліноміальну залежність від розміру вхідних даних задачі, та за допомогою якого можна перевірити коректність розв'язку за кількість операцій, що має поліноміальну залежність від розміру вхідних даних задачі. Маючи $\varphi=\phi(q, g; \varepsilon)$ та k^*, можна легко перевірити нерівність

$$G(k^*) > E(q, g; \varphi) - 1.$$

Справедливість цієї нерівності означає, що знайдена розмітка k^* дійсно є розв'язком поданої на вхід алгоритму (max,+) задачі розмітки з цілочисельними вагами ребер та вершин.

Приклад 1. Візьмемо довільне додатне парне число $x \ge 10$. Нехай задачу задано множиною об'єктів $T=\{t_1, t_2, t_3, t_4\}$, структурою сусідства $\Gamma=\{(t_1, t_2), (t_1, t_3), (t_2, t_4), (t_3, t_4)\}$, множиною міток $K=\{0,1,2\}$, вагами вершин

$$q_{t_1}(0)=q_{t_2}(0)=q_{t_3}(0)=0,$$
$$q_{t_1}(1)=q_{t_2}(1)=q_{t_3}(1)=-\frac{x}{2},$$
$$q_{t_1}(2)=q_{t_2}(2)=q_{t_3}(2)=-x,$$
$$q_{t_4}(0)=-x+1,$$
$$q_{t_4}(1)=-\frac{x}{2}-1,$$
$$q_{t_4}(2)=-1$$

і вагами ребер

$$g_{tt'}(k, k')=-\left(\frac{x}{2}+1\right)\cdot [k \ne k'],$$

за яких задача не є супермодулярною.

Запустимо процедуру субградієнтного спуску. Виберемо початкову репараметризацію

$$\varphi_{tt'}(k)=0, \ \forall k \in K, \ \forall t \in T, \ \forall t' \in N_t$$

і крок субградієнтного спуску

$$\gamma_i=\left(\frac{x}{2}+1\right)/ i.$$

Виконаємо дві ітерації субградієнтного спуску й отримаємо значення репараметризації

$$\varphi_{t_3t_4}(0)=\left(\frac{x}{2}+1\right)\Big/ 2, \ \varphi_{t_3t_4}(2)=-\left(\frac{x}{2}+1\right)\Big/ 2,$$

$$\varphi_{t_2 t_4}(0) = \left(\frac{x}{2}+1\right) \Big/ 2, \ \varphi_{t_2 t_4}(2) = -\left(\frac{x}{2}+1\right) \Big/ 2,$$

$$\varphi_{t_4 t_3}(0) = -\left(\frac{x}{2}+1\right) \Big/ 2, \ \varphi_{t_4 t_3}(2) = \left(\frac{x}{2}+1\right) \Big/ 2,$$

$$\varphi_{t_4 t_2}(0) = -\left(\frac{x}{2}+1\right) \Big/ 2, \ \varphi_{t_4 t_2}(2) = \left(\frac{x}{2}+1\right) \Big/ 2.$$

Всі інші значення репараметризації φ не змінилися. За такої репараметризації і $\varepsilon \le \dfrac{1}{|T|+2\cdot|\Gamma|+1} \approx 0.077$ задача є ε-узгодженою. Значення двоїстої функції

$$E(q, g, \varphi) = -x+1.$$

У кожному об'єкті всі мітки мають скінченну вагу, тому другий крок алгоритму пропустимо. Обхід об'єктів графу здійснюємо у порядку (t_1, t_2, t_3, t_4), а обхід міток в об'єкті виконуємо у порядку $(0,1,2)$. Нехай $K_1 = \{0\}$, $K_2 = \{1, 2\}$. Зафіксуємо мітку 0 в об'єкті t_1 та заборонимо інші мітки у цьому об'єкті

$$q'_{t_1}(0) = 0, \ q'_{t_1}(1) = \infty, \ q'_{t_1}(2) = \infty, \ q'_t(k) = q_t(k) \ \ \forall t \in \{t_2, t_3, t_4\}, \ \forall k \in K.$$

Використаємо значення φ, отримані на попередньому кроці. Одразу маємо ε-узгоджений набір вершин та ребер, а також

$$E(q', g, \varphi) = E(q, g, \varphi) = -x+1.$$

В об'єкті t_1 залишимо мітку 0, тобто $k^*_{t_1} = 0$. Проведемо аналогічну процедуру для всіх інших об'єктів. У результаті отримаємо оптимальну розмітку k^*

$$k^*(t_1) = k^*(t_2) = k^*(t_3) = k^*(t_4) = 0$$

з

$$G(k^*) = -x+1 > E(q, g; \varphi) - 1 = -x.$$

Авторам нині невідомо, яким є співвідношення між кількістю супермодулярних задач та кількістю тих задач, які може розв'язувати наведений алгоритм. Водночас цей приклад свідчить про те, що множина не супермодулярних задач, які наведений алгоритм може розв'язувати, не є пустою.

СПИСОК ЛІТЕРАТУРИ

1. Boykov Y., Veksler O., Zabih R. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2001. Vol. 23, Iss. 11. P. 1222–1239. https://doi.org/10.1109/34.969114.
2. Boykov Y., Kolmogorov V. An experimental comparison of min-cut/maxflow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2004. Vol. 26, Iss. 9. P. 1124–1137. https://doi.org/10.1109/TPAMI.2004.60.
3. Шлезингер М., Гигиняк В. Решение (max, +)-задач структурного распознавания с помощью их эквивалентных преобразований. *Управляющие системы и машины*. 2007. № 1. С. 3–15.
4. Szeliski R., Zabih R., Scharstein D. et al. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2008. Vol. 30, Iss. 6. P. 1068–1080.
5. Wang C., Komodakis N., Paragios N. Markov random field modeling, inference & learning in computer vision & image understanding: a survey. *Computer Vision and Image Understanding (CVIU)*. 2013. Vol. 117, Iss. 11. P. 1610–1627.
6. Savchynskyy B. Discrete graphical models — an optimization perspective. *Foundations and Trends in Computer Graphics and Vision*. 2019. Vol. 11, N 3–4. P. 160–429.

7. Ishikawa H. Exact optimization for Markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2003. Vol. 25, Iss. 10. P. 1333–1336.
8. Шлезингер М.И., Антонюк К.В. Анализ алгоритмов диффузии для решения оптимизационных задач структурного распознавания. *Кибернетика и системный анализ*. 2011. Т. 47, № 2. С. 3–20.
9. Шлезингер М.И. Распознавание образов как реализация определенного подкласса процессов мышления. *Управляющие системы и машины*. 2017. № 2. С. 20–37.
10. Vodolazskii E., Flach B., Schlesinger M. Minimax problems of discrete optimization invariant under majority operators. *Computational Mathematics and Mathematical Physics*. 2014. Vol. 54, Iss. 8. P. 1327–1336.
11. Водолазский Е.В. Обобщенные задачи разметки с мажоритарным полиморфизмом для некоторого класса полуколец. *Управляющие системы и машины.* 2015. № 6. С. 3–7.
12. Shor N. Minimization methods for non-differentiable functions. Springer Series in Computational Mathematics. Berlin: Springer-Verlag, 1985. Vol. 3. P. 22–48.
13. Schlesinger M., Vodolazskiy E., Lopatka N. Stop condition for subgradient minimization in dual relaxed (max, +) problem. *Proc. 8th International Conference on EMMCVPR (Energy Minimization Methods in Computer Vision and Pattern Recognition)* (25-27 July, 2011 Saint Petersburg, Russia). Saint Petersburg, 2011. P. 118–131.
14. Schlesinger D., Flach B. Transforming an arbitrary MinSum problem into a binary one. Tech. report. TU Dresden, Fak. Informatik. 2006.
15. Shlezinger M. Syntactic analysis of two-dimensional visual signals in the presence of noise. *Cybernetics.* 1976. Vol. 12, N 4. P. 612–628.
16. Werner T. Revisiting the linear programming relaxation approach to Gibbs energy minimization and weighted constraint satisfaction. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2010. Vol. 32, Iss. 8. P. 1474–1488.
17. Коваль В., Шлезингер М. Двумерное программирование в задачах анализа изображений. *Автоматика и телемеханика.* 1976. Т. 37, № 8. С. 149–168.
18. Rossi F., van Beek P., Walsh T. Handbook of Constraint Programming. Elsevier Science, 2006. 957 p.
19. Li M., Shekhovtsov A., Huber D. Complexity of discrete energy minimization problems. In: Computer Vision — ECCV 2016. Leibe B., Matas J., Sebe N., Welling M. (Eds.). Cham: Springer International Publishing, 2016. P. 834–852.
20. Arora S., Barak B. Computational Complexity: A Modern Approach. USA: Cambridge University Press, 2009. 594 p.

V. Krygin, R. Khomenko

SELF-DRIVEN ALGORITHM FOR SOLVING SUPERMODULAR (max,+) LABELING PROBLEMS BASED ON SUBGRADIENT DESCENT

Abstract. An algorithm presented in this article gives a correct answer to one of the two questions for any (max, +) labeling problem with integer weights: either “What is the best labeling?” or “Is the problem supermodular?”, and this answer is guaranteed to be correct. The algorithm is called self-driven because the user cannot decide which of the two questions will be answered — this decision is up to the algorithm. Also, the algorithm does not need to know the order of labels if the problem is supermodular. The finite execution time of the algorithm is guaranteed for integer weights of vertices and edges and use of subgradient descent.

Keywords: (max,+) labeling problems, supermodular labeling problems, self-driven pattern recognition, discrete optimization, graphical models, structural pattern recognition.

Надійшла до редакції 05.03.2022