УДК 004.93'1:[519.814+519.857]
Нетрадиційні задачі розпізнавання тексту в рамках байєсівської теорії прийняття рішень.

Савчинський Б.Д.
Міжнародний науково-навчальний центр
інформаційних технологій та систем.
пр.Академіка Глушкова, 40, Київ т.266-62-08
bogdan@image.kiev.ua

Анотація. Розглядається байєсівський підхід до задачі розпізнавання текстових зображень. Існуючі її розв'язки аналізуються з точки зору постановок відповідних байєсівських задач розпізнавання. В рамках байєсівської теорії поставлені та розв'язані дві альтернативні задачі розпізнавання текстових зображень.

Вступ

Проблема розпізнавання текстів має свою історію, що триває вже кілька десятиліть. Дослідження цієї проблеми були започатковані основоположними результатами Ковалевського [1, 2], що зводили задачі розпізнавання до вирішення певних задач динамічного програмування. Застосування цього підходу дало змогу знаходити найімовірнішу послідовність літер, яка відповідає заданому зображенню рядка тексту. В цьому напрямі були вирішені численні прикладні проблеми. Завдяки одержаним успіхам такий підхід став свого роду аксіомою і набув значного поширення (див., напр., [3, 4, 5, 6]).

Дана робота відходить від встановленого стереотипу знаходження найімовірнішої послідовності літер, що в будь-якому разі є лише окремим випадком байєсівської теорії прийняття рішень. Розгляд задачі з точки зору цієї, більш загальної теорії, викладеної у [7], допоміг знайти нові, прийнятніші формулювання задачі розпізнавання текстів і, відповідно, побудувати нові алгоритми її вирішення.

Робота складається з чотирьох розділів. Перший розділ присвячений постановці і розв'язку задачі розпізнавання текстів як задачі, що формулюється в рамках байєсівського підходу. У другому розділі розглянута функція втрат, яка відповідає традиційній постановці задачі. В третьому розділі запропоновано дві інші функції втрат та алгоритми розв'язку задач, що їм відповідають. Четвертий розділ присвячений відкритим питанням.

1 Постановка задачі в рамках байєсівської теорії

Задача розпізнавання рядка тексту полягає у сегментації зображення рядка по довжині на фрагменти, кожному з яких поставлена у відповідність та чи інша літера алфавіту. Таким чином, кожній літері відповідає інтервал стовпчиків зображення. Задля лаконічності викладу ми оперуватимемо кожним стовпчиком як єдиним цілим і лише за потреби нагадуватимемо, що він сам є вертикальною послідовністю пікселів.

Назвемо *полем зору* множину $T = \{1, \ldots, |T|\}$. Елементами поля зору є порядкові номери вертикальних стовпчиків пікселів зображення.

Позначимо символом V множину сигналів – множину всіх можливих значень, які може приймати стовпчик пікселів зображення. Функцію вигляду $x : T \to V$ називатимемо *зображенням.* Множину всіх зображень позначатимемо через V^T.

Деяку скінчену множину A ми називатимемо *алфавітом.* Її елементами є імена літер тексту.

Кожній літері $\alpha \in A$ поставимо у відповідність її *ширину* $d(\alpha)$. Таким чином вважатимемо визначеною певну функцію $d : A \to \mathbf{N}$.

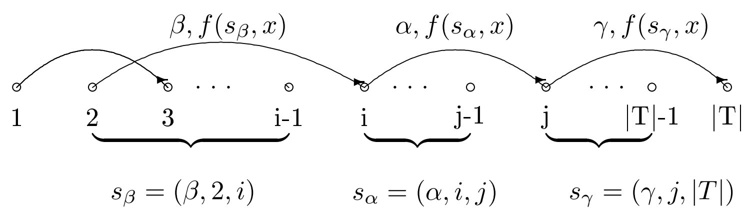

Рис. 1: Граф, що ілюструє модель рядка тексту. Фігурними дужками позначені сегменти, а орієнтованими дугами – ребра графа, які їм відповідають. На ребрах вказані імена відповідних сегментів та значення локальної функції відмінності.

Сегментом назвемо поіменований фрагмент поля зору, ширина якого дорівнює ширині літери – імені сегмента. Таким чином, сегмент – це сукупність трьох величин $(\alpha, b_l, b_r),\ \alpha \in A,\ b_l, b_r \in T,\ b_r - b_l = d(\alpha)$. Величини b_l та b_r, які ми називатимемо відповідно *лівою та правою границями сегмента*, визначають цілочисельний відрізок $[b_l, b_r - 1]$ на полі зору. Елементами цього відрізка є всі номери стовбців поля зору, які менші за b_r та не менші за b_l. Величину α називатимемо *ім'ям* або *позначкою* сегмента. Множину всіх сегментів позначатимемо S. Для позначення імені та границь конкретного сегмента $s \in S$ використовуватимемо відповідно позначення $\alpha(s), b_l(s), b_r(s)$.

Введемо поняття *сегментації* поля зору, як послідовності довільної кількості сегментів, які, щільно прилягаючи один до одного, покривають все поле зору:

$$\begin{gathered} \vec{s} = (s_1, s_2, \ldots, s_n),\ s_i \in S,\ 1 \le i \le n, \\ b_r(s_j) = b_l(s_{j+1}),\ 1 \le j < n,\ b_l(s_1) = 1,\ b_r(n) = |T|. \end{gathered} \tag{1}$$

Множину усіх можливих сегментацій позначатимемо $\vec{S}$.

Для i-того сегмента сегментації $\vec{s}$ використовуватимемо позначення s_i. Кількість сегментів у сегментації $\vec{s}$ позначатимемо $n(\vec{s})$. Функцію вигляду $f : S \times V^T \to R$ називатимемо *локальною функцією відмінності.* Її значення $f(s, x)$ визначає, наскільки зображення x в місці, визначеному сегментом s, не схоже на літеру, яка задається ім'ям сегмента $\alpha(s)$. Фактично величина $f(s, x)$ залежить не від усього зображення x, а лише від того його фрагменту, який задається границями сегмента s.

Введені поняття можуть бути проілюстровані за допомогою графа $G = (V, E)$, зображеного на рис. 1. Множина вершин графа збігається з полем зору, $V = T$, а множина ребер E – з множиною всіх сегментів S. А саме: сегменту $s = (\alpha, b_l, b_r)$ ставиться у відповідь ребро з початком у вершині b_l, кінцем у вершині b_r, позначкою α та вагою $f(s, x)$. Отже пару вершин графа b_l та b_r поєднує стільки ребер, скільки літер мають ширину $b_r - b_l$. Будь-якому шляху на графі, який починається у вершині 1 і закінчується у вершині $|T|$, відповідає деяка сегментація поля зору, і, навпаки, кожній сегментації відповідає деякий шлях на графі з початком у вершині 1 і кінцем у вершині $|T|$.

Для формулювання задачі в рамках байєсівської теорії (див.,напр., [7, 8]) нам потрібно визначити множину спостережень об'єкта, множину станів, розподіл їх спільних імовірностей, множину результатів розпізнавання та функцію втрат.

Множина спостережень об'єкта – це множина всіх можливих зображень V^T. *Множина станів об'єкта* є множиною сегментацій $\vec{S}$.

Вважатимемо, що апріорний розподіл $p(\vec{s})$ на множині всіх сегментацій $\vec{S}$ є рівномірним $p(\vec{s}) = p_0$, а умовний розподіл спостережень об'єкта при заданому його стані визначається як:

$$p(x/\vec{s}) = \prod_{i=1}^{n(\vec{s})} c^{d(\alpha(s_i))} \exp\{-f(s_i, x)\},$$

де c – нормувальна стала.

При таких припущеннях апостеріорний розподіл сегментації за умови фіксованого зображення матиме вигляд:

$$p(\vec{s}/x) = \frac{p(x/\vec{s})p(\vec{s})}{p(x)} =$$

$$= p(\vec{s})/p(x) \cdot \prod_{i=1}^{n(\vec{s})} c^{d(\alpha(s_i))} \exp\{-f(s_i, x)\} = \tag{2}$$

$$= p_0/p(x) \cdot c^{|T|} \cdot \prod_{i=1}^{n(\vec{s})} \exp\{-f(s_i, x)\} =$$

$$= C(x) \cdot \prod_{i=1}^{n(\vec{s})} \exp\{-f(s_i, x)\} =$$

$$= C(x) \cdot \exp\{-\sum_{i=1}^{n(\vec{s})} f(s_i, x)\},$$

де C – деяка функція від зображення x.

Множиною результатів розпізнавання доцільно вважати множину послідовностей літер – елементів алфавіту $\vec{A} = \{(\alpha_1, \alpha_1, ..., \alpha_n),\ \alpha_i \in A,\ i = 1..n,\ n \in \mathbf{N},\ \sum_{i=1}^{n} d(\alpha_i) = |T|\}$. Через $n(\vec{\alpha})$ позначатимемо кількість літер у послідовності $\vec{\alpha}$.

З будь-якою послідовністю $\vec{\alpha} \in \vec{A}$ пов'язана сегментація поля зору, яка кожному елементу послідовності α_i ставить у відповідність сегмент поля зору s_i: $segm(\vec{\alpha} = (\alpha_1, \alpha_2, \ldots \alpha_n)) = (s_1, s_2, \ldots s_n),\ s_i \in S,\ \alpha(s_i) = \alpha_i$. Сегмент з порядковим номером i сегментації $segm(\vec{\alpha})$ позначатимемо $segm_i(\vec{\alpha})$.

Функція втрат $W : \vec{A} \times \vec{S} \to R$ при цьому матиме таке значення: величина $W(\vec{\alpha}, \vec{s})$ визначатиме штраф за ситуацію, коли зображення, істинною сегментацією якого є $\vec{s}$, було розпізнане як $\vec{\alpha}$. Задача полягає у знаходженні такої послідовності $\vec{\alpha^*}$ літер алфавіту, яка б мінімізувала математичне сподівання втрат:

$$\vec{\alpha^*} = \arg\min_{\vec{\alpha} \in \vec{A}} \sum_{\vec{s} \in \vec{S}} p(\vec{s}/x) W(\vec{\alpha}, \vec{s}). \tag{3}$$

2 Функція втрат, що відповідає традиційній постановці задачі розпізнавання

Розглянемо функцію втрат, яка, як буде показано далі, веде до традиційної задачі знаходження найімовірнішої послідовності літер, вперше розглянутої і розв'язаної в [1]. Ми наведемо тут її постановку і розв'язок, щоб підкреслити її відмінність від задач, запропонованих нами далі.

Тож нехай функція втрат має вигляд:

$$W(\vec{\alpha}, \vec{s}) = \begin{cases} 0, & \text{якщо } \vec{s} = segm(\vec{\alpha}) \\ 1, & \text{якщо } \vec{s} \neq segm(\vec{\alpha}) \end{cases} . \tag{4}$$

Згідно з (3) та (4):

$$\vec{\alpha^*} = \arg\min_{\vec{\alpha} \in \vec{A}} \sum_{\substack{\vec{s} \in \vec{S} \\ \vec{s} \neq segm(\vec{\alpha})}} p(\vec{s}/x) =$$

$$= \arg\min_{\vec{\alpha} \in \vec{A}} (1 - p(segm(\vec{\alpha})/x)) =$$

$$= \arg\max_{\vec{\alpha} \in \vec{A}} p(segm(\vec{\alpha})/x) = \tag{5}$$

$$= \arg\max_{\vec{\alpha} \in \vec{A}} C(x) \cdot \exp\{-\sum_{i=1}^{n(\vec{\alpha})} f(segm_i(\vec{\alpha}), x)\} =$$

$$= \arg\min_{\vec{\alpha} \in \vec{A}} \sum_{i=1}^{n(\vec{\alpha})} f(segm_i(\vec{\alpha}), x). \tag{6}$$

Отже, функції втрат (4) відповідає оцінка максимальної правдоподібності (5), знаходження якої може власне саме слугувати постановкою задачі, що і спостерігаємо в роботах [1, 2, 3, 4, 5, 6].

Для розв'язку задачі (6) в роботах [1, 2] використано метод динамічного програмування, що полягає у наступному:

позначимо через $S_R(t) \subset S,\ t \in T$, множину тих сегментів, права границя яких рівна t: $S_R(t) = \{s \in S:\ b_r(s) = t\}$, а через $S_L(t) \subset S,\ t \in T$, множину тих сегментів, ліва границя яких рівна t: $S_L(t) = \{s \in S:\ b_l(s) = t\}$.

Перепишемо формулу (6) у наступній еквівалентній формі:

$$segm(\vec{\alpha}^*) = \arg\min_{\vec{s} \in \vec{S}} \sum_{i=1}^{n(\vec{s})} f(s_i, x) =$$

$$= \arg \min_{s_1 \in S_L(1)} \min_{s_2 \in S_L(b_r(s_1))} \cdots \min_{s_{n(\vec{s})} \in \left(S_L(b_r(s_{n(\vec{s})-1})) \bigcap S_R(|T|)\right)} \sum_{i=1}^{n(\vec{s})} f(s_i, x). \tag{7}$$

Введемо величини $F(s),\ s \in S$ згідно формул:

$$F(s) = \min_{s_1 \in S_L(1)} \min_{s_2 \in S_L(b_r(s_1))} \cdots \min_{s_i \in \left(S_L(b_r(s_{i-1})) \bigcap S_R(b_l(s))\right)} (f(s, x) + \sum_{j=1}^{i} f(s_j, x)).$$

Штраф за найкращу сегментацію, яка вибирається згідно формули (7), дорівнює значенню найменшої з величин $F(s)$ для тих s, права границя яких збігається з правою границею поля зору, а останнім сегментом s' у найкращій сегментації є той, на якому досягається вказаний мінімум $F(s)$:

$$s' = \arg \min_{s \in S_R(|T|)} F(s). \tag{8}$$

Очевидно, для тих сегментів s, ліва границя яких збігається з лівою границею поля зору, значення величин $F(s)$ дорівнює $f(s, x)$. Значення ж їх для усіх інших сегментів s задається наступною рекурентною формулою:

$$F(s) = \min_{s' \in S_R(b_l(s))} (F(s') + f(s, x)). \tag{9}$$

Для цих сегментів визначимо також функцію індексу $ind : S \to S$, таку, що її значення $ind(s)$ вказує на сегмент – попередник сегмента s:

$$ind(s) = \arg \min_{s' \in S_R(b_l(s))} (F(s') + f(s, x)). \tag{10}$$

Вирішення задачі (6) полягає у впорядкуванні сегментів за їхньою правою границею і послідовному застосуванні до них формул (9) та (10). Насамкінець, для сегментів, права границя яких збігається з правою границею поля зору, слід використати формулу (8). Найкраща сегментація $\vec{s^*}$ задається рекурентними формулами:

$$s_{n(\vec{s^*})} = s',\ s_i = ind(s_{i+1}),\ i = 0...n(\vec{s^*}) - 1. \tag{11}$$

Алгоритм (9) - (11) може розглядатись як алгоритм пошуку найкращого шляху, який починається у вершині 1 і закінчується у вершині $|T|$, зваженого орієнтованого графа, введеного в попередньому розділі і зображеного на рис. 1. Величини $F(s)$ в формулах (8), (9) приймають значення ваги найкращого шляху серед тих, які починаються у вершині 1, закінчуються у вершині $b_r(s)$ і містять ребро, що відповідає сегменту s.

Розглянута задача, як вказує формула (6), може розумітись також як задача апроксимації зображення послідовністю відомих еталонів і тому може бути поставлена без залучення будь-яких статистичних міркувань. Але якщо вже пов'язувати її зі статистичними міркуваннями, то, можливо, слід це робити інакше, використовуючи адекватніші за (4) функції втрат, наприклад так, як це зроблено в наступному розділі.

3 Постановка та розв'язок задачі для двох альтернативних функцій втрат

3.1 Постановка та загальний вигляд розв'язку

Розглянемо питання про доречність функції втрат вигляду (4) у поставленій задачі. Величина $W(\vec{\alpha}, \vec{s})$ визначає штраф за ситуацію, коли зображення, істинною сегментацією якого є $\vec{s}$, було розпізнане як $\vec{\alpha}$. Така функція втрат не враховує, наскільки відрізняється результат розпізнавання від

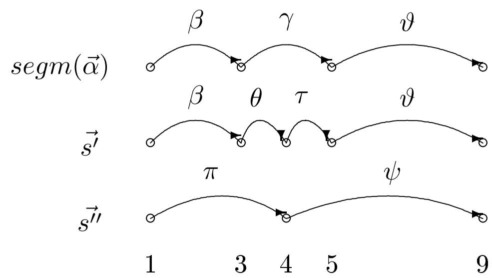

Рис. 2: Приклади сегментацій. У випадку функції втрат (12), (13), штрафи рівні $W(\vec{\alpha}, \vec{s'}) = 1$, $W(\vec{\alpha}, \vec{s''}) = 3$, а у випадку функції втрат (12), (14), $W(\vec{\alpha}, \vec{s'}) = 2$, $W(\vec{\alpha}, \vec{s''}) = 3$.

правильного результату. Так, наприклад, залишається неврахованою кількість неправильно розпізнаних літер, однаково штрафується рядок, в якому всі літери визначені хибно, і рядок, в якому є лише одна неправильна літера.

Більш реалістичними є, наприклад, функції втрат, що задаються парами формул (12), (13) та (12), (14):

$$W(\vec{\alpha}, \vec{s}) = \sum_{i=1}^{n(\vec{\alpha})} w_i(segm_i(\vec{\alpha}), \vec{s}), \tag{12}$$

$$w_i(s', \vec{s}) = \begin{cases} 0, & \text{якщо } \exists j : s_j = s' \\ 1 & \text{в усіх інших випадках} \end{cases}, \tag{13}$$

$$w_i(s', \vec{s}) = \begin{cases} 0, & \text{якщо } (n(\vec{s}) \geq i) \;\&\; (\alpha(s') = \alpha(s_i)) \\ 1 & \text{в усіх інших випадках} \end{cases}. \tag{14}$$

Значення функції (12), (13) рівне кількості сегментів, які є у результаті розпізнавання $\vec{\alpha}$, але відсутні в істинній сегментації $\vec{s}$. Нехай, наприклад, результатом розпізнавання є послідовність $\vec{\alpha}$, з якою пов'язана сегментація $segm(\vec{\alpha})$, що складається з трьох сегментів $(\beta, 1, 3), (\gamma, 3, 5), (\vartheta, 5, 9)$, а істинними є сегментації $\vec{s'} = (\beta, 1, 3), (\theta, 3, 4), (\tau, 4, 5), (\vartheta, 5, 9)$ та $\vec{s''} = (\pi, 1, 4), (\psi, 4, 9)$. Ці сегментації зображені на рис. 2. Значення функції втрат на парі $(\vec{\alpha}, \vec{s'})$ рівне 1, оскільки сегмент $(\gamma, 3, 5)$ відсутній у сегментації $\vec{s'}$, а на парі $(\vec{\alpha}, \vec{s''})$ рівне 3, оскільки жоден з трьох сегментів розв'язку $segm(\vec{\alpha})$ не присутній у сегментації $\vec{s''}$.

Функція втрат, яка задається формулами (12), (14), визначає штраф, що не залежить від границь сегментів, а визначається лише їхніми іменами та порядковими номерами в сегментації. За i-ий сегмент з іменем α призначається штраф 0, якщо ім'я i-ого сегмента правильної сегментації рівне α, і штраф 1 в протилежному випадку. Для сегментацій, зображених на рис. 2, штрафи відповідно рівні $W(\vec{\alpha}, \vec{s'}) = 2$ та $W(\vec{\alpha}, \vec{s''}) = 3$.

Розглянемо задачі, в яких за функцію втрат взята функція вигляду (12):

$$\begin{aligned} \vec{\alpha^*} &= \arg\min_{\vec{\alpha} \in \vec{A}} \sum_{\vec{s} \in \vec{S}} p(\vec{s}/x) W(\vec{\alpha}, \vec{s}) = \\ &= \arg\min_{\vec{\alpha} \in \vec{A}} \sum_{\vec{s} \in \vec{S}} p(\vec{s}/x) \sum_{i=1}^{n(\vec{\alpha})} w_i(segm_i(\vec{\alpha}), \vec{s}) = \\ &= \arg\min_{\vec{\alpha} \in \vec{A}} \sum_{\vec{s} \in \vec{S}} \sum_{i=1}^{n(\vec{\alpha})} p(\vec{s}/x) w_i(segm_i(\vec{\alpha}), \vec{s}) = \\ &= \arg\min_{\vec{\alpha} \in \vec{A}} \sum_{i=1}^{n(\vec{\alpha})} \sum_{\vec{s} \in \vec{S}} p(\vec{s}/x) w_i(segm_i(\vec{\alpha}), \vec{s}). \end{aligned} \tag{15}$$

Формула (15) вказує на загальний вигляд байєсівської задачі розпізнавання за умови, що функція штрафу має вигляд (12). В наступних двох підрозділах ми використаємо це для розв'язку задач, в яких функція w_i конкретизована за допомогою формул (13) та (14).

3.2 Розв'язок задачі для першої функції втрат

Введемо позначення $\vec{S}(s')$ для множини тих сегментацій, які містять сегмент $s' \in S$. Підставивши (13) в (15), отримаємо:

$$\begin{aligned}
\vec{\alpha^*} &= \arg\min_{\vec{\alpha}\in\vec{A}} \sum_{i=1}^{n(\vec{\alpha})} \sum_{\vec{s}\in\vec{S}\setminus\vec{S}(segm_i(\vec{\alpha}))} p(\vec{s}/x) = \\
&= \arg\min_{\vec{\alpha}\in\vec{A}} \sum_{i=1}^{n(\vec{\alpha})} \left(1 - \sum_{\vec{s}\in\vec{S}(segm_i(\vec{\alpha}))} p(\vec{s}/x)\right) = \\
&= \arg\min_{\vec{\alpha}\in\vec{A}} \sum_{i=1}^{n(\vec{\alpha})} \left(1 - p(segm_i(\vec{\alpha})/x)\right), \\
\text{де } p(segm_i(\vec{\alpha})/x) &= \sum_{\vec{s}\in\vec{S}(segm_i(\vec{\alpha}))} p(\vec{s}/x).
\end{aligned} \tag{16}$$

За умови знання величин $p(segm_i(\vec{\alpha})/x)$ задача (16) розв'язується аналогічно задачі (6) підстановкою $1 - p(segm_i(\vec{\alpha})/x)$ замість $f(segm_i(\vec{\alpha}), x)$, а отже, задача для функції втрат (12), (13) буде вважатись розв'язаною, як тільки ми вкажемо спосіб обчислення величин $p(s/x),\ s \in S$. Саме цьому присвячено решту цього підрозділу.

Запишемо необхідні для знаходження величин $p(s/x),\ s \in S$, перетворення, використавши (2) та позначивши $\varphi(s,x) = c^{d(\alpha(s))} \cdot \exp\{-f(s,x)\}$ та $Z(x) = \frac{p(\vec{s})}{p(x)}$:

$$\begin{aligned}
p(s/x) &= \sum_{\vec{s}\in\vec{S}(s)} p(\vec{s}/x) = \\
&= Z(x) \cdot \sum_{\vec{s}\in\vec{S}(s)} \prod_{i=1}^{n(\vec{s})} \varphi(s_i, x) = \\
&= Z(x) \cdot \sum_{j\in T} \sum_{\substack{\vec{s}\in\vec{S} \\ \vec{s}=(s_1,s_2,\ldots,s_{j-1},s,s_{j+1},\ldots,s_{n(\vec{s})})}} \left[\prod_{i=1}^{j-1} \varphi(s_i,x)\right] \times \\
&\times [\varphi(s,x)] \times \left[\prod_{i=j+1}^{n(\vec{s})} \varphi(s_i,x)\right] = \\
&= Z(x) \cdot \sum_{j\in T} \sum_{s_1\in S_L(1)} \sum_{s_2\in S_L(b_r(s_1))} \cdots \sum_{s_{j-1}\in\left(S_L(b_r(s_{j-2}))\bigcap S_R(b_l(s))\right)} \left[\prod_{i=1}^{j-1} \varphi(s_i,x)\right] \times \\
&\times \varphi(s,x) \times \\
&\times \sum_{n>j} \sum_{s_n\in S_R(|T|)} \sum_{s_{n-1}\in S_R(b_l(s_n))} \cdots \sum_{s_{j+1}\in\left(S_R(b_l(s_{j+2}))\bigcap S_L(b_r(s))\right)} \left[\prod_{i=j+1}^{n} \varphi(s_i,x)\right]
\end{aligned} \tag{17}$$

За умови знання величини $Z(x)$ обчислення (17) може бути зроблене таким чином. Введемо величини $F_{forw}(s)$ та $F_{back}(s),\ s \in S$, які визначимо так:

$$\begin{aligned}
F_{forw}(s) &= \sum_{j\in T} \sum_{s_1\in S_L(1)} \sum_{s_2\in S_L(b_r(s_1))} \cdots \sum_{s_{j-1}\in\left(S_L(b_r(s_{j-2}))\bigcap S_R(b_l(s))\right)} \left[\prod_{i=1}^{j-1} \varphi(s_i,x)\right] \cdot \varphi(s,x), \\
F_{back}(s) &= \sum_{j\in T} \sum_{s_1\in S_R(|T|)} \sum_{s_2\in S_R(b_l(s_1))} \cdots \sum_{s_{j-1}\in\left(S_R(b_l(s_{j-2}))\bigcap S_L(b_r(s))\right)} \left[\prod_{i=1}^{j-1} \varphi(s_i,x)\right] \cdot \varphi(s,x).
\end{aligned} \tag{18}$$

За допомогою введених позначень перепишемо (17):

$$p(s/x) = Z(x) \cdot \frac{F_{forw}(s) \cdot F_{back}(s)}{\varphi(s,x)}. \tag{19}$$

Очевидно, що значення $F_{forw}(s)$ для всіх сегментів s таких, що їхня ліва границя збігається з лівою границею поля зору, як і значення $F_{back}(s)$ для всіх сегментів таких, що їхня права границя збігається з правою границею поля зору, дорівнює $\varphi(s,x)$.

Значення ж величин $F_{forw}(s)$ та $F_{back}(s)$ для усіх інших сегментів s, очевидно, задаються рекурентними формулами:

$$F_{forw}(s) = \varphi(s,x) \cdot \sum_{s' \in S_R(b_l(s))} F_{forw}(s') \tag{20}$$

$$F_{back}(s) = \varphi(s,x) \cdot \sum_{s' \in S_L(b_r(s))} F_{back}(s'). \tag{21}$$

Алгоритм обчислення величин $p(s/x)$ полягає у впорядкуванні сегментів за їх правою границею, послідовному застосуванні до них формули (20), потім у впорядкуванні сегментів за лівою границею, послідовному застосуванні до них формули (21) і, насамкінець, обчислення $p(s/x)$ за формулою (19).

Ми не вказали лише, яким чином обчислюється величина $Z(x)$. Вкажемо, записавши її у вигляді:

$$\begin{aligned} Z(x) &= \frac{p(\vec{s})}{p(x)} = \\ &= \frac{p(\vec{s})}{\sum_{\vec{s} \in \vec{S}} p(x/\vec{s}) p(\vec{s})} = \\ &= \frac{p_0}{p_0 \cdot \sum_{\vec{s} \in \vec{S}} \prod_{i=1}^{n(\vec{s})} \varphi(s,x)} = \\ &= \frac{1}{\sum_{s \in S_R(|T|)} F_{forw}(s)}. \end{aligned}$$

Обчислення останнього виразу за умови знання $F_{forw}(s)$ не становить жодних труднощів і може бути виконане шляхом прямого застосування вказаних у ньому операцій.

Алгоритм, що визначається формулами (19) - (21), може розглядатись як алгоритм пошуку сумарної ваги усіх шляхів на графі, які проходять через задане його ребро. Розглянемо граф, введений раніше для ілюстрації розв'язку задачі (6), приписавши ребру, яке відповідає сегменту s, замість значення ваги $f(s,x)$ значення $\varphi(s,x)$. Вважатимемо також, що вагою будь-якого шляху, який починається у вершині 1 графа і закінчується у вершині $|T|$, є добуток ваг усіх ребер шляху. Тоді знаходження $p(s/x)$ еквівалентне знаходженню сумарної ваги всіх таких шляхів, які окрім того містять ребро графа, що відповідає сегменту s. Величини $F_{forw}(s)$ мають значення сумарної ваги всіх шляхів, які починаються у вершині 1, закінчуються у вершині $b_r(s)$ і містять ребро, що відповідає сегменту s, а величини $F_{back}(s)$ – сумарної ваги всіх шляхів, які починаються у $b_l(s)$, закінчуються у $|T|$ і містять ребро, що відповідає сегменту s.

3.3 Розв'язок задачі для другої функції втрат

Введемо позначення $\vec{S}(i,\alpha),\ i \in T,\ \alpha \in A$ для множини тих сегментацій, i-ий сегмент яких має ім'я α. Для розв'язку задачі з функцією втрат (12), (14) підставимо її в (15):

$$\begin{aligned} \vec{\alpha}^* = \arg\min_{\vec{\alpha} \in \vec{A}} \sum_{i=1}^{n(\vec{\alpha})} \sum_{\vec{s} \in \vec{S}} p(\vec{s}/x) w_i(segm_i(\vec{\alpha}), \vec{s}) = \\ = \arg\min_{\vec{\alpha} \in \vec{A}} \sum_{i=1}^{n(\vec{\alpha})} \sum_{\vec{s} \in \vec{S} \setminus \vec{S}(i,\alpha_i)} p(\vec{s}/x) = \\ = \arg\min_{\vec{\alpha} \in \vec{A}} \sum_{i=1}^{n(\vec{\alpha})} \left(1 - \sum_{\vec{s} \in \vec{S}(i,\alpha_i)} p(\vec{s}/x) \right) = \end{aligned}$$

$$= \arg\min_{\vec{\alpha}\in\vec{A}} \sum_{i=1}^{n(\vec{\alpha})} \left(1 - p_i(\alpha_i/x)\right), \tag{22}$$

$$\text{де } p_i(\alpha_i/x) = \sum_{\vec{s}\in\vec{S}(i,\alpha_i)} p(\vec{s}/x). \tag{23}$$

Величини $p_i(\alpha/x)$ можуть розумітись як маргінальні ймовірності того, що на i-тому місці рядка $\vec{\alpha}$ знаходиться літера α.

На перший погляд формула (22) аналогічна формулам (6) і (16), і, здавалося б, алгоритми їх обчислення повинні співпадати. Але це лише ілюзорна схожість, оскільки параметр $1 - p_i(\alpha_i/x)$, що є аргументом суми, залежить від параметра i – індекса, по якому відбувається додавання. Тому знаходження (22) є обчислювально складнішою задачею, яка потребує окремого розгляду. Ми розглянемо її у два етапи: спочатку вкажемо шлях обчислення (22), вважаючи відомими величини $p_i(\alpha_i/x)$, а потім наведемо алгоритм знаходження $p_i(\alpha_i/x)$.

3.3.1 Розв'язок задачі за умови знання величин $p_i(\alpha_i/x)$.

Перепишемо (22) у еквівалентній формі:

$$segm(\vec{\alpha^*}) = \arg\min_{\vec{s}\in\vec{S}} \sum_{i=1}^{n(\vec{s})} (1 - p_i(\alpha(s_i)/x)) = \tag{24}$$

$$= \arg \min_{s_1\in S_L(1)} \min_{s_2\in S_L(b_r(s_1))} \cdots \min_{s_{n(\vec{s})}\in\left(S_L(b_r(s_{n(\vec{s})-1}))\bigcap S_R(|T|)\right)} \sum_{i=1}^{n(\vec{s})}(1 - p_i(\alpha(s_i)/x)).$$

Введемо величини $F_i(s)$ згідно формул:

$$F_i(s) = \min_{s_1\in S_L(1)} \min_{s_2\in S_L(b_r(s_1))} \cdots \min_{s_{i-1}\in\left(S_L(b_r(s_{i-2}))\bigcap S_R(b_l(s))\right)} \left(1 - p_i(\alpha(s)/x) + \sum_{j=1}^{i-1}(1 - p_j(\alpha(s_j)/x))\right).$$

Очевидно, для тих сегментів s, ліва границя яких збігається з лівою границею поля зору, значення величин $F_1(s)$ рівне $(1 - p_1(\alpha(s)/x))$. Значення ж їх для усіх інших сегментів s та індексів i задаються наступною рекурентною формулою:

$$F_i(s) = \min_{s'\in S_R(b_l(s))} (F_{i-1}(s') + 1 - p_i(\alpha(s)/x)). \tag{25}$$

Для цих сегментів та індексів визначимо також функцію індексу $ind_i : S \to S$, таку, що її значення $ind_i(s)$ вказує на сегмент – попередник сегмента s:

$$ind_i(s) = \arg \min_{s'\in S_R(b_l(s))} (F_{i-1}(s') + 1 - p_i(\alpha(s)/x)). \tag{26}$$

Штраф за найкращу сегментацію, яка вибирається згідно формули (24), дорівнює значенню найменшої з величин $F_i(s)$, $i \in T$, для тих s, права границя яких збігається з правою границею поля зору, а останнім сегментом s' у найкращій сегментації є той, на якому досягається вказаний мінімум $F_i(s)$:

$$s' = \arg\min_i \min_{s\in S_R(|T|)} F_i(s). \tag{27}$$

Вирішення задачі (24) полягає у впорядкуванні сегментів за їхньою правою границею і послідовному застосуванні до них формул (25) та (26). Насамкінець, для сегментів, права границя яких збігається з правою границею поля зору, слід використати формулу (27). Найкраща сегментація $\vec{s^*} = segm(\vec{\alpha^*})$ задається рекурентними формулами:

$$segm_{n(\vec{\alpha^*})}(\vec{\alpha^*}) = s',\ segm_i(\vec{\alpha^*}) = ind_{i+1}(segm_{i+1}(\vec{\alpha^*})),\ i = 0...n(\vec{\alpha^*}) - 1. \tag{28}$$

Алгоритм, що визначається формулами (25) - (28), також може розглядатись як алгоритм пошуку найкращого шляху на графі. Розглянемо граф, введений раніше для ілюстрації розв'язку задачі (6). Для кожного сегмента s замінимо одне ребро, яке йому відповідає і з'єднує вершини

$b_l(s)$ та $b_r(s)$, $|T|$ ребрами, кожне з яких з'єднує вершини $b_l(s)$ та $b_r(s)$. Занумеруємо ці ребра від 1 до $|T|$, приписавши кожному з них у відповідності з номером i значення ваги $1 - p_i(\alpha(s)/x)$. В множині всіх шляхів на графі, які починаються у вершині 1, є підмножина шляхів така, що номери ребер кожного шляху зростають рівно на одиницю при переході від поточного ребра до наступного. Формула (24) визначає шлях, що закінчується у вершині $|T|$, є елементом цієї підмножини і має найменшу сумарну вагу ребер. Величини $F_i(s)$ мають значення ваги найкращого шляху з вершини 1 у вершину $b_r(s)$ серед тих шляхів, які належать вказаній підмножині та містять рівно i ребер, останнім з яких є ребро, що відповідає сегменту s.

3.3.2 Обчислення маргінальних ймовірностей літер $p_i(\alpha_i/x)$.

В цьому підрозділі ми вкажемо, яким чином обчислюються величини $p_i(\alpha/x),\ \alpha \in A$, що задаються формулою (23). Введемо позначення $\vec{S}(i,s),\ i \in T,\ s \in S$ для множини тих сегментацій, i-им сегментом яких є сегмент s. Запишемо величини $p_i(\alpha/x)$ у вигляді:

$$p_i(\alpha/x) = \sum_{\substack{s \in S \\ \alpha(s)=\alpha}} p_i(s/x),$$

$$\text{де } p_i(s/x) = \sum_{\vec{s} \in \vec{S}(i,s)} p(\vec{s}/x) = Z(x) \cdot \sum_{\vec{s} \in \vec{S}(i,s)} \prod_{i=1}^{n(\vec{s})} \varphi(s,x) =$$

$$= Z(x) \cdot \sum_{s_1 \in S_L(1)} \sum_{s_2 \in S_L(b_r(s_1))} \cdots \sum_{s_{i-1} \in \left(S_L(b_r(s_{i-2})) \bigcap S_R(b_l(s))\right)} \left[\prod_{j=1}^{i-1} \varphi(s_j,x)\right] \times$$

$$\times \varphi(s,x) \times$$

$$\times \sum_{n>i} \sum_{s_n \in S_R(|T|)} \sum_{s_{n-1} \in S_R(b_l(s_n))} \cdots \sum_{s_{i+1} \in \left(S_R(b_l(s_{i+2})) \bigcap S_L(b_r(s))\right)} \left[\prod_{j=i+1}^{n} \varphi(s_j,x)\right]. \tag{29}$$

Для обчислення величин $p_i(s/x)$ введемо величини $F^i_{forw}(s),\ i \in T,\ s \in S$, які визначимо наступним чином:

$$F^i_{forw}(s) = \sum_{s_1 \in S_L(1)} \sum_{s_2 \in S_L(b_r(s_1))} \cdots \sum_{s_{i-1} \in \left(S_L(b_r(s_{i-2})) \bigcap S_R(b_l(s))\right)} \left[\prod_{j=1}^{i-1} \varphi(s_j,x)\right] \cdot \varphi(s,x).$$

Введемо також величини $F_{back}(s),\ s \in S$, що визначаються згідно формули (18). Алгоритм їх обчислення поданий у підрозділі 3.2.

У введених позначеннях формула (29) для обчислення величин $p_i(s/x)$ приймає вигляд:

$$p_i(s/x) = Z(x) \cdot \frac{F^i_{forw}(s) \cdot F_{back}(s)}{\varphi(s,x)}. \tag{30}$$

Очевидно, що значення $F^1_{forw}(s)$ для всіх сегментів s таких, що їх ліва границя збігається з лівою границею поля зору, дорівнює $\varphi(s,x)$. Значення ж величин $F^i_{forw}(s)$ для усіх інших значень i та s, очевидно, задається рекурентною формулою:

$$F^i_{forw}(s) = \varphi(s,x) \cdot \sum_{s' \in S_R(b_l(s))} F^{i-1}_{forw}(s'). \tag{31}$$

Алгоритм обчислення величин $p_i(s/x)$ полягає у впорядкуванні сегментів за їхньою правою границею, послідовному застосуванні до них формули (31) і використанні формули (30).

Алгоритм (30) - (31) може розглядатись як алгоритм пошуку сумарної ваги усіх шляхів на графі, у яких задане ребро зустрічається під заданим порядковим номером від початку шляху. Розглянемо граф, введений раніше для ілюстрації розв'язку задачі (17). Знаходження $p_i(s/x)$ еквівалентне знаходженню сумарної ваги всіх шляхів, у яких ребро графа, що відповідає сегменту s, має i-тий порядковий номер. Величини $F^i_{forw}(s)$ мають значення сумарної ваги всіх тих шляхів з вершини 1 у вершину $b_r(s)$, що містять рівно i ребер, останнім з яких є ребро, яке відповідає сегменту s.

4 Відкриті питання.

Таким чином, постановка задачі розпізнавання тексту як задачі пошуку найімовірнішого рядка не повинна бути догмою. Існують і інші, не менш обґрунтовані постановки цієї задачі, що також допускають ефективні алгоритми її розв'язку. Ми вказали два приклади таких задач. Звичайно, окрім наведених нами в цих задачах функцій втрат, існують також інші, які видаються інтуїтивно ще кращими. Однією з найприйнятніших, на наш погляд, є задача, в якій функція втрат дорівнює відстані редагування (відстані Левенштейна) (див., напр., [7]) між двома рядками. Знаходження її точного розв'язку нам видається доволі непростою, і, безумовно, честолюбною задачею, вирішення якої означало б певний прорив у структурному статистичному розпізнаванні.

Література

[1] Ковалевский В.А. Оптимальный алгоритм распознавания некоторых последовательностей изображений. *Кибернетика*, (4), 1967.

[2] Ковалевский В.А. *Методы оптимальных решений в распознавании изображений.* Наука, Москва, 1976.

[3] Rejean Plamodon, Sargur N. Srihari. On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. *IEEE Trans. on PAMI*, 22(1):63–84, January 2000.

[4] Amlan Kundu, Yang He, Mou-Yen Chen. Alternatives to Variable Duration HMM in Handwriting Recognition. *IEEE Trans. on PAMI*, 20(11):1275–1280, November 1998.

[5] U. Marti and H. Bunke. Handwritten sentence recognition. In *Proceedings of 15 th International Conference on Pattern Recognition*, volume 3, pages 467–470, Barcelona, 2000.

[6] K. Aas, L. Eikvil, and T. Andersen. Text recognition from grey level images using hidden Markov models. *Lecture Notes in Computer Science*, 970:503–508, 1995.

[7] Michail I. Schlesinger, Vaclav Hlavač. *Ten lectures on statistical and structural pattern recognition.* Kluwer Academic Publishers, Dordrecht/Boston/London, 2002.

[8] Дуда Р., Харт П. *Распознавание образов и анализ сцен.* Мир, Москва, 1976.