УДК 004.93'1:[519.814+519.857]
Нетрадиционные задачи распознавания текста в рамках байесовской теории принятия решений.

Савчинский Б.Д.
Международный научно-учебный центр информационных технологий и систем.
пр.Академика Глушкова, 40, Киев 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.

Аннотация. Рассматривается байесовский подход к задаче распознавания текстовых изображений. Существующие её решения анализируются с точки зрения постановок соответствующих им байесовских задач распознавания. В рамках байесовской теории поставлены и решены две альтернативные задачи распознавания текстовых изображений.

Abstract. Bayesian approach to an optical text recognition task is considered. Existing solutions of the task are analysed from the point of view of appropriate bayesian task's statements. Two alternative tasks of an optical text recognition are stated and solved in the network of bayesian theory.

Вступление

Проблема распознавания текстов имеет свою историю, длящуюся уже несколько десятилетий. Начало исследований этой проблемы было ознаменовано основополагающими результатами Ковалевского [1, 2], которые сводили задачи распознавания к решению определенных задач динамического программирования.

Использование этого подхода дало возможность находить наиболее вероятную последовательность литер, которая соответствует заданному изображению строки текста. В этом направлении были решены многочисленные прикладные задачи. Благодаря полученным успехам этот подход стал своего рода аксиомой и получил значительное распространение (см., напр., [3, 4, 5, 6]).

Данная работа отходит от установленного стереотипа нахождения наиболее вероятной последовательности литер, что всегда является лишь частным случаем байесовской теории принятия решений. Рассмотрение задачи с точки зрения этой, более общей теории, изложенной в [7], помогло найти новые, более приемлемые формулировки задачи распознавания текстов и, соответственно, построить новые алгоритмы её решения.

Работа состоит из четырех разделов. Первый раздел посвящен постановке и решению задачи распознавания текстов как задачи, которая формулируется в рамках байесовского подхода. Во втором разделе рассмотрена функция потерь, соответствующая традиционной постановке задачи распознавания. В третьем разделе предложено две другие функции потерь и соответствующие им алгоритмы решения задач. Четвертый раздел посвящен открытым вопросам.

1 Постановка задачи в рамках байесовской теории

Задача распознавания строки текста состоит в сегментации изображения строки по длине на фрагменты, каждому из которых поставлена в соответствие та или иная литера алфавита. Таким образом, каждой литере соответствует интервал столбиков изображения. В целях лаконичности изложения мы будем оперировать каждым столбиком как единым целым и лишь при необходимости будем вспоминать, что он сам является вертикальной последовательностью пикселов.

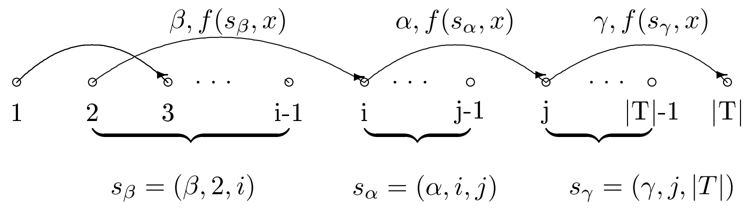

Рис. 1: Граф, иллюстрирующий модель строки текста. Фигурными скобками обозначены сегменты, а ориентированными дугами – соответствующие им рёбра графа. На рёбрах указаны имена соответствующих сегментов и значение локальной функции отличия.

Назвем *полем зрения* множество $T = \{1, \ldots, |T|\}$. Элементами поля зрения являются порядковые номера столбиков пикселов изображения.

Обозначим символом V множество сигналов – множество всех возможных значений, которые может принимать столбик пикселов изображения. Функцию вида $x : T \to V$ будем называть *изображением*. Множество всех изображений обозначим V^T.

Некоторое конечное множество A будем называть *алфавитом*. Его элементами являются имена литер текста.

Каждой литере $\alpha \in A$ поставим в соответствие её *ширину* $d(\alpha)$. Таким образом будем считать определённой некоторую функцию $d : A \to \mathbf{N}$.

Сегментом назовём поименованый фрагмент поля зрения, ширина которого равна ширине литеры – имени сегмента. Таким образом, сегмент – это совокупность трёх величин $(\alpha, b_l, b_r),\ \alpha \in A,\ b_l, b_r \in T,\ b_r - b_l = d(\alpha)$. Величины b_l и b_r, которые мы будем называть соответственно *левым и правым краями сегмента*, определяют целочисленный отрезок $[b_l, b_r - 1]$ поля зрения. Элементами этого отрезка являются все номера столбиков поля зрения, которые меньше, чем b_r, и не больше, чем b_l. Величину α будем называть *именем* или *меткой* сегмента. Множество всех сегментов обозначим S. Для обозначения имени и краёв конкретного сегмента $s \in S$ будем использовать соответственно обозначения $\alpha(s), b_l(s), b_r(s)$.

Введём понятие *сегментации* поля зрения, как последовательности произвольного количества сегментов, которые, плотно прилегая друг к другу, покрывают всё поле зрения:

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

Множество всех возможных сегментаций обозначим $\vec{S}$.

Для i-того сегмента сегментации $\vec{s}$ будем использовать обозначение s_i. Количество сегментов в сегментации $\vec{s}$ обозначим $n(\vec{s})$. Функцию вида $f : S \times V^T \to R$ назовем *локальной функцией отличия*. Её значение $f(s, x)$ определяет, насколько изображение x в месте, определенном сегментом s, не похоже на литеру, задаваемую именем сегмента $\alpha(s)$. Фактически величина $f(s, x)$ зависит не от всего изображения x, а лишь от его фрагмента, задаваемого краями сегмента s.

Введенные понятия могут быть проиллюстрированы с помощью графа $G = (V, E)$, изображенного на рис. 1. Множество вершин графа совпадает с полем зрения, $V = T$, а множество ребер E – с множеством всех сегментов S. А именно: сегменту $s = (\alpha, b_l, b_r)$ ставится в соответствие ребро с началом в вершине b_l, концом в вершине b_r, меткой α и весом $f(s, x)$. Таким образом пару вершин графа b_l и b_r соединяет столько рёбер, сколько литер имеют ширину $b_r - b_l$. Любому пути на графе, начинаюшемуся в вершине 1 и заканчивающемуся в вершине $|T|$, соответствует некоторая сегментация поля зрения, и, наоборот, каждой сегментации соответствует некоторый путь на графе с началом в вершине 1 и концом в вершине $|T|$.

Для формулирования задачи в рамках байесовской теории (см.,напр., [7, 8]) нам необходимо определить множество наблюдений объекта, множество состояний, распределение их совместных вероятностей, множество результатов распознавания и функцию потерь.

Множество наблюдений объекта – это множество всех возможных изображений V^T. *Множество состояний объекта* является множеством сегментаций $\vec{S}$.

Будем считать, что априорное распределение $p(\vec{s})$ на множестве всех сегментаций $\vec{S}$ является равномерным $p(\vec{s}) = p_0$, а условное распределение наблюдений объекта при заданном его состоянии определяется как:

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

где c – нормировочный множитель.

В этом случае апостериорное распределение сегментации при условии фиксированного изображения будет иметь вид:

$$\begin{aligned} p(\vec{s}/x) = \frac{p(x/\vec{s})p(\vec{s})}{p(x)} = \\ = p(\vec{s})/p(x) \cdot \prod_{i=1}^{n(\vec{s})} c^{d(\alpha(s_i))} \exp\{-f(s_i, x)\} = \\ = p_0/p(x) \cdot c^{|T|} \cdot \prod_{i=1}^{n(\vec{s})} \exp\{-f(s_i, x)\} = \\ = C(x) \cdot \prod_{i=1}^{n(\vec{s})} \exp\{-f(s_i, x)\} = \\ = C(x) \cdot \exp\{-\sum_{i=1}^{n(\vec{s})} f(s_i, x)\}, \end{aligned} \tag{2}$$

где C – некоторая функция от изображения x.

Множеством результатов распознавания целесообразно считать множество последовательностей литер – элементов алфавита $\vec{A} = \{(\alpha_1, \alpha_1, ..., \alpha_n),\ \alpha_i \in A,\ i = 1..n,\ n \in \mathbf{N},\ \sum_{i=1}^{n} d(\alpha_i) = |T|\}$. Количество литер в последовательности $\vec{\alpha}$ будем обозначать $n(\vec{\alpha})$.

С любой последовательностью $\vec{\alpha} \in \vec{A}$ связана сегментация поля зрения, ставящая в соответствие каждому элементу α_i последовательности сегмент поля зрения 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 с порядковым номером i сегментации $segm(\vec{\alpha})$ обозначим $segm_i(\vec{\alpha})$.

Функция потерь $W : \vec{A} \times \vec{S} \to R$ при этом будет иметь такой смысл: величина $W(\vec{\alpha}, \vec{s})$ определяет штраф за ситуацию, когда изображение, истинной сегментацией которого есть $\vec{s}$, было распознано как $\vec{\alpha}$. Задача состоит в нахождениии последовательности $\vec{\alpha^*}$ литер алфавита, минимизирующей математическое ожидание потерь:

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

2 Функция потерь, соответствующая традиционной постановке задачи распознавания

Рассмотрим функцию потерь, которая, как будет показано ниже, ведет к традиционной задаче нахождения наиболее вероятной последовательности литер, впервые рассмотренной и решенной в [1]. Мы приведем тут её постановку и решение, чтобы подчеркнуть отличие от задач, предложенных нами далее.

Итак, пусть функция потерь имеет вид:

$$W(\vec{\alpha}, \vec{s}) = \left\{ \begin{array}{ll} 0, & \text{если } \vec{s} = segm(\vec{\alpha}) \\ 1, & \text{если } \vec{s} \neq segm(\vec{\alpha}) \end{array} \right. . \tag{4}$$

Согласно (3) и (4):

$$\begin{aligned} \vec{\alpha^*} = \arg\min_{\vec{\alpha} \in \vec{A}} \sum_{\substack{\vec{s} \in \vec{S} \\ \vec{s} \neq segm(\vec{\alpha})}} p(\vec{s}/x) = \\ = \arg\min_{\vec{\alpha} \in \vec{A}} (1 - p(segm(\vec{\alpha})/x)) = \end{aligned}$$

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

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

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

Таким образом, функции потерь (4) соответствует оценка максимального правдоподобия (5), нахождение которой может само по себе служить постановкой задачи, что и наблюдается в работах [1, 2, 3, 4, 5, 6].

Для решения задачи (6) в работах [1, 2] использован метод динамического программирования, состоящий в следующем: обозначим с помощью $S_R(t)\subset S,\ t\in T$, множество тех сегментов, правый край которых равен t: $S_R(t)=\{s\in S:\ b_r(s)=t\}$, а с помощью $S_L(t)\subset S,\ t\in T$, множество тех сегментов, левый край которых равен t: $S_L(t)=\{s\in S:\ b_l(s)=t\}$.

Перепишем формулу (6) в следующем эквивалентном виде:

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

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

Введём величины $F(s),\ s\in S$ согласно 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 наилучшую сегментацию, которая выбирается согласно формулы (7), равен значению наименьшей из величин $F(s)$ для тех s, правый край которых совпадает с правым краем поля зрения, а последним сегментом s' в наилучшей сегментации является тот, на котором достигается указанный минимум $F(s)$:

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

Очевидно, что для тех сегментов s, левый край которых совпадает с левым краем поля зрения, значения величин $F(s)$ равно $f(s,x)$. Значения же их для всех других сегментов s задаются следующей рекуррентной формулой:

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

Для этих сегментов определим также функцию индекса $ind:S\to S$, такую, что её значение $ind(s)$ указывает на сегмент – предшественник сегмента s:

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

Решение задачи (6) состоит в упорядочении сегментов по их правому краю и последовательному применению к ним формул (9) и (10). После этого для сегментов, правый край которых совпадает с правым краем поля зрения, следует воспользоваться формулой (8). Наилучшая сегментация $\vec{s^*}$ задаётся рекуррентными формулами:

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

Алгоритм (9) - (11) может рассматриваться как алгоритм поиска наилучшего пути, начинающегося в вершине 1 и заканчивающегося в вершине $|T|$, взвешенного ориентированного графа, введенного в предыдущем разделе и изображенного на рис. 1. Величины $F(s)$ в формулах (8), (9) принимают значение веса наилучшего пути среди путей, начинающихся в вершине 1, заканчивающихся в вершине $b_r(s)$ и содержащих ребро, соответствующее сегменту s.

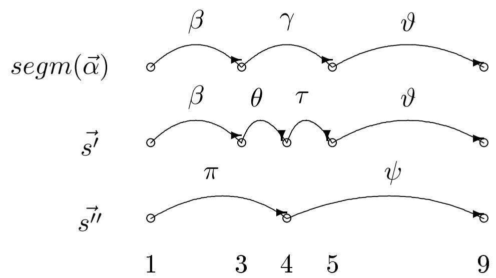

Рис. 2: Примеры сегментаций. В случае функции потерь (12), (13), штрафы равны $W(\vec{\alpha}, \vec{s'}) = 1$, $W(\vec{\alpha}, \vec{s''}) = 3$, а в случае функции потерь (12), (14), $W(\vec{\alpha}, \vec{s'}) = 2$, $W(\vec{\alpha}, \vec{s''}) = 3$.

Рассмотренная задача, как указывает формула (6), также может пониматься как задача аппроксимации изображения последовательностью известных эталонов и поэтому она может быть поставлена без привлечения каких-либо статистических соображений. Но если уж связывать её со статистическими соображениями, то, возможно, следует это делать иначе, используя функции потерь, более адекватные, чем (4), например такие, как это сделано в следующем разделе.

3 Постановка и решение задачи для двух альтернативных функций потерь

3.1 Постановка и общий вид решения

Рассмотрим вопрос об уместности функции потерь вида (4) в поставленной задаче. Величина $W(\vec{\alpha}, \vec{s})$ определяет штраф за ситуацию, в которой изображение, истинной сегментацией которого является $\vec{s}$, было распознано как $\vec{\alpha}$. Такая функция потерь не учитывает, насколько отличается результат распознавания от правильного результата. Так, например, остаётся неучтённым количество неправильно распознанных литер, одинаково штрафуется строка, в которой все литеры определены неправильно, и строка, в которой есть только одна неправильная литера.

Более реалистичными являются, например, функции потерь, задаваемые парами формул (12), (13) и (12), (14):

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

$$w_i(s', \vec{s}) = \begin{cases} 0, & \text{если } \exists j : s_j = s' \\ 1 & \text{во всех остальных случаях} \end{cases}, \tag{13}$$

$$w_i(s', \vec{s}) = \begin{cases} 0, & \text{если } (n(\vec{s}) \geq i) \ \& \ (\alpha(s') = \alpha(s_i)) \\ 1 & \text{во всех остальных случаях} \end{cases}. \tag{14}$$

Значение функции (12), (13) равно количеству сегментов, присутствующих в результате распознавания $\vec{\alpha}$, но отсутствующих в истинной сегментации $\vec{s}$. Пусть, например, результатом распознавания является последовательность $\vec{\alpha}$, с которой связана сегментация $segm(\vec{\alpha})$, состоящая из трёх сегментов $(\beta, 1, 3), (\gamma, 3, 5), (\vartheta, 5, 9)$, а истинными есть сегментации $\vec{s'} = (\beta, 1, 3), (\theta, 3, 4), (\tau, 4, 5), (\vartheta, 5, 9)$ и $\vec{s''} = (\pi, 1, 4), (\psi, 4, 9)$. Эти сегментации изображены на рис. 2. Значение функции потерь на паре $(\vec{\alpha}, \vec{s'})$ равно 1, поскольку сегмент $(\gamma, 3, 5)$ отсутствует в сегментации $\vec{s'}$, а на паре $(\vec{\alpha}, \vec{s''})$ равно 3, поскольку ни один из трёх сегментов решения $segm(\vec{\alpha})$ не присутствует в сегментации $\vec{s''}$.

Функция потерь, задаваемая формулами (12), (14), определяет штраф, не зависящий от положения краев сегментов, а зависящий только от их имён и порядковых номеров в сегментации. За i-ый сегмент с именем α назначается штраф 0, если имя i-ого сегмента правильной сегментации равно α, и штраф 1 в противном случае. Для сегментаций, изображенных на рис. 2, штрафы соответственно равны $W(\vec{\alpha}, \vec{s'}) = 2$ и $W(\vec{\alpha}, \vec{s''}) = 3$.

Рассмотрим задачи, в которых в качестве функции потерь взята функция вида (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) указывает общий вид байесовской задачи распознавания при условии, что функция потерь имеет вид (12). В следующих двух подразделах мы используем это для решения задач, в которых функция w_i конкретизирована с помощью формул (13) и (14).

3.2 Решение задачи для первой функции потерь

Введём обозначение $\vec{S}(s')$ для множества сегментаций, содержащих сегмент $s' \in S$. Подставив (13) в (15), получим:

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

$$\text{где } p(segm_i(\vec{\alpha})/x) = \sum_{\vec{s}\in\vec{S}(segm_i(\vec{\alpha}))} p(\vec{s}/x).$$

При известных величинах $p(segm_i(\vec{\alpha})/x)$ задача (16) решается аналогично задаче (6) подстановкой $1-p(segm_i(\vec{\alpha})/x)$ вместо $f(segm_i(\vec{\alpha}),x)$, а следовательно, задача для функции потерь (12), (13) будет считаться решенной, как только мы укажем способ вычисления величин $p(s/x)$, $s \in S$. Именно этому посвящена остальная часть этого подраздела.

Запишем необходимые для нахождения величин $p(s/x)$, $s \in S$, преобразования, использовав (2) и обозначив $\varphi(s,x) = c^{d(\alpha(s))} \cdot \exp\{-f(s,x)\}$ и $Z(x) = \frac{p(\vec{s})}{p(x)}$:

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

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

$$\times\sum_{n>j}\sum_{s_n\in S_R(|T|)}\sum_{s_{n-1}\in S_R(b_l(s_n))}\cdots\sum_{s_{j+1}\in\left(S_R(b_l(s_{j+2}))\bigcap S_L(b_r(s))\right)}\left[\prod_{i=j+1}^{n}\varphi(s_i,x)\right] \tag{17}$$

При условии знания величин $Z(x)$ вычисление (17) может быть сделано следующим образом. Введём величины $F_{forw}(s)$ и $F_{back}(s),\ s\in S$, которые определим так:

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

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

При помощи введенных обозначений перепишем (17):

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

Очевидно, что значения $F_{forw}(s)$ для всех сегментов s таких, что их левый край совпадает с левым краем поля зрения, как и значения $F_{back}(s)$ для всех сегментов таких, что их правый край совпадает с правым краем поля зрения, равно $\varphi(s,x)$.

Значения же величин $F_{forw}(s)$ и $F_{back}(s)$ для остальных сегментов s, очевидно, задаются рекуррентными 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 вычисления величин $p(s/x)$ состоит в упорядочивании сегментов согласно их правому краю, последовательному применению к ним формулы (20), потом в упорядочивании сегментов по левому краю, последовательному применению к ним формулы (21) и, наконец, вычислении $p(s/x)$ по формуле (19).

Мы не указали лишь, каким образом вычисляется величина $Z(x)$. Укажем, записав её в виде:

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

Вычисление последнего выражения при условии знания величин $F_{forw}(s)$ не содержит никаких сложностей и может быть сделано путём прямого использования указанных в нём операций.

Алгоритм, определяемый формулами (19) - (21), может рассматриваться как алгоритм поиска суммарного веса всех путей на графе, которые проходят через заданное его ребро. Рассмотрим граф, введенный ранее для иллюстрации решения задачи (6), приписав ребру, соответствующему сегменту s, вместо веса $f(s,x)$ вес $\varphi(s,x)$. Будем также считать, что весом любого пути, начинающегося в вершине 1 графа и заканчивающегося в вершине $|T|$, является произведение весов всех его рёбер. Тогда нахождение $p(s/x)$ эквивалентно нахождению суммарного веса всех таких путей, содержащих ребро, соответствующее сегменту s. Величины $F_{forw}(s)$ принимают значение суммарного веса всех путей, начинающихся в вершине 1, заканчивающихся в вершине $b_r(s)$ и содержащих ребро, соответствующее сегменту s, а величины $F_{back}(s)$ – суммарного веса всех путей, начинающихся в $b_l(s)$, заканчивающихся в $|T|$ и содержащих ребро, соответствующее сегменту s.

3.3 Решение задачи для второй функции потерь

Введём обозначение $\vec{S}(i,\alpha),\ i \in T,\ \alpha \in A$ для множества тех сегментаций, i-ый сегмент которых имеет имя α. Для решения задачи с функцией потерь (12), (14) подставим её в (15):

$$\begin{aligned}
\vec{\alpha^*} &= \arg\min_{\vec{\alpha}\in\vec{A}} \sum_{i=1}^{n(\vec{\alpha})} \sum_{\vec{s}\in\vec{S}} p(\vec{s}/x) w_i(segm_i(\vec{\alpha}),\vec{s}) = \\
&= \arg\min_{\vec{\alpha}\in\vec{A}} \sum_{i=1}^{n(\vec{\alpha})} \sum_{\vec{s}\in\vec{S}\setminus\vec{S}(i,\alpha_i)} p(\vec{s}/x) = \\
&= \arg\min_{\vec{\alpha}\in\vec{A}} \sum_{i=1}^{n(\vec{\alpha})} \left(1 - \sum_{\vec{s}\in\vec{S}(i,\alpha_i)} p(\vec{s}/x)\right) = \\
&= \arg\min_{\vec{\alpha}\in\vec{A}} \sum_{i=1}^{n(\vec{\alpha})} \left(1 - p_i(\alpha_i/x)\right), \qquad (22)
\end{aligned}$$

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

Величины $p_i(\alpha/x)$ могут пониматься, как маргинальные вероятности того, что на i-том месте строки $\vec{\alpha}$ находится литера α.

На первый взгляд формула (22) аналогична формулам (6) и (16), поэтому, казалось бы, алгоритмы их вычисления должны совпадать. Но это лишь иллюзорная схожесть, так как параметр $1 - p_i(\alpha_i/x)$, являющийся аргументом суммы, зависит от параметра i – индекса, по которому происходит суммирование. Поэтому нахождение (22) является вычислительно более сложной задачей, требующей отдельного рассмотрения. Мы рассмотрим её в два этапа: сначала укажем способ вычисления (22), считая известными величины $p_i(\alpha_i/x)$, а потом рассмотрим алгоритм нахождения $p_i(\alpha_i/x)$.

3.3.1 Решение задачи при условии знания величин $p_i(\alpha_i/x)$.

Перепишем (22) в эквивалентной форме:

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

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

Введём величины $F_i(s)$ согласно 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, что для тех сегментов s, левый край которых совпадает с левым краем поля зрения, значения величин $F_1(s)$ равны $(1 - p_1(\alpha(s)/x))$. Значения же их для всех остальных сегментов s и индексов i задаются следующей рекуррентной формулой:

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

Для этих сегментов определим также функцию индекса $ind_i : S \to S$, такую, что её значение $ind_i(s)$ указывает на сегмент – предшественник сегмента s:

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

Штраф за наилучшую сегментацию, выбираемую согласно формулы (24), равен значению наименьшей из величин $F_i(s),\ i \in T$, для тех s, правый край которых совпадает с правым краем поля зрения, а последним сегментом s' в наилучшей сегментации является тот, на котором достигается минимум $F_i(s)$:

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

Решение задачи (24) состоит в упорядочении сегментов по их правому краю и последовательному применению к ним формул (25) и (26). В самом конце для сегментов, правый край которых сопадает с правым краем поля зрения, следует воспользоваться формулой (27). Наилучшая сегментация $\vec{s^*} = segm(\vec{\alpha^*})$ задается рекуррентными 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, определяемый формулами (25) - (28), также может рассматриваться как алгоритм поиска наилучшего пути на графе. Рассмотрим граф, введенный ранее для иллюстрации решения задачи (6). Для каждого сегмента s заменим одно ребро, соответствующее ему и соединяющее вершины $b_l(s)$ и $b_r(s)$, $|T|$ рёбрами, каждое из которых соединяет вершины $b_l(s)$ и $b_r(s)$. Пронумеруем эти рёбра от 1 до $|T|$, приписав каждому из них в соответствии с номером i вес $1 - p_i(\alpha(s)/x)$. В множестве всех путей на графе, начинающихся в вершине 1, есть такое подмножество путей, что номера рёбер каждого пути из этого подмножества возрастают ровно на единицу при переходе от текущего ребра к следующему. Формула (24) определяет путь, заканчивающийся в вершине $|T|$, являющийся элементом этого множества и имеющий минимальный суммарный вес рёбер. Величины $F_i(s)$ равны значениям весов наилучших путей из вершины 1 в вершину $b_r(s)$ среди путей, принадлежащих указанному подмножеству и содержащих ровно i рёбер, последним из которых является ребро, соответствующее сегменту s.

3.3.2 Вычисление маргинальных вероятностей литер $p_i(\alpha_i/x)$.

В этом подразделе мы укажем, каким образом вычисляются величины $p_i(\alpha/x), \; \alpha \in A$, задаваемые формулой (23). Введём обозначение $\vec{S}(i,s), \; i \in T, \; s \in S$ для множества тех сегментаций, i-ым сегментом которых является сегмент s. Запишем величины $p_i(\alpha/x)$ в виде:

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

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

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

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

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

Для вычисления величин $p_i(s/x)$ введём величины $F^i_{forw}(s), \; i \in T, \; s \in S$, которые определим следующим образом:

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

Введем также величины $F_{back}(s), \; s \in S$, определяемые согласно формулы (18). Алгоритм их вычисления подан в подразделе 3.2.

В этих обозначениях формула (29) для вычисления величин $p_i(s/x)$ принимает вид:

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

Очевидно, что значения $F^1_{forw}(s)$ для всех сегментов s таких, что их левый край совпадает с левым краем поля зрения, равны $\varphi(s,x)$. Значения же величин $F^i_{forw}(s)$ для всех остальных значений i и s, очевидно, задаются рекуррентной формулой:

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

Алгоритм вычисления величин $p_i(s/x)$ состоит в упорядочении сегментов по их правому краю, последовательному применению к ним формулы (31) и использовании формулы (30).

Алгоритм (30) - (31) может рассматриваться как алгоритм поиска суммарного веса всех путей на графе, в которых заданное ребро встречается под заданным порядковым номером от начала пути. Рассмотрим граф, введенный ранее для иллюстрации решения задачи (17). Нахождение $p_i(s/x)$ эквивалентно нахождению суммарного веса всех путей, у которых ребро графа, соответствующее сегменту s, имеет i-тый порядковый номер. Величины $F^i_{forw}(s)$ принимают значение суммарного веса всех путей из вершины 1 в вершину $b_r(s)$, содержащих ровно i ребер, последним из которых является ребро, соответствующее сегменту s.

4 Открытые вопросы.

Таким образом, постановка задачи распознавания текста как задачи поиска наиболее вероятной строки не должна быть догмой. Существуют и другие, не менее обоснованные постановки этой задачи, которые также допускают эффективные алгоритмы её решения. Мы указали два примера таких задач. Конечно, кроме приведенных нами в этих задачах функций потерь, существуют также другие, интуитивно представляющиеся ещё лучшими. Одной из наиболее приемлемых, на наш взгляд, является задача с функцией потерь, которая равна расстоянию редактирования (расстоянию Левенштейна) (см., напр., [7]) между двумя строками. Нахождение её точного решения нам представляется достаточно сложной, и, безусловно, честолюбивой задачей, решение которой означало бы определенный прорыв в структурном статистическом распознавании.

Список литературы

[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.