Записки Ленинградского ордена Ленина, ордена Октябрьской Революции
и ордена Трудового Красного Знамени горного института
им. Г.В.Плеханова, т. XVIII, вып. 1, 1975 г.

МЕТОД ЭФФЕКТИВНОГО КОДИРОВАНИЯ ИНФОРМАЦИОННЫХ ТЕКСТОВ, ОСНОВАННЫЙ НА ОПТИМАЛЬНОМ РАСШИРЕНИИ АЛФАВИТА

А.Н.Лившиц

Задача оптимального кодирования информационных текстов в значительной степени определяет эффективность ИПС и АСУ. Решение этой задачи приводит к минимизации объема необходимой памяти ЭВМ и повышению быстродействия информационного поиска.

Оптимальное кодирование информационных текстов, в смысле минимума средней кодовой длины на символ алфавита, обеспечивается применением неравномерных кодов Хаффмена.[x] Кодирование по Хаффмену основано на знании вектора вероятностей P принятого алфавита $P=\{p_i\}$, $(i=1,2,...,n)$, где p_i – вероятность появления в рассматриваемом тексте i-го символа алфавита; n – число символов алфавита.

В случае представления текста над естественным алфавитом вектор P определяется однозначно и, следовательно, применением кодов Хаффмена задача оптимизации кодовой длины текста полностью решена.

Однако, как показывают наши исследования, эффективность кодирования по методу Хаффмена может быть значительно усилена, если рассматривается представление текста над расширенным алфавитом.

Целесообразность расширения естественного алфавита обусловлена значительной избыточностью естественных языков, а также языков программирования. Возникающие при этом специфические трудности решения задачи оптимального кодирования заключаются в неопределенности выбора расширенного алфавита, адекватного заданному тексту, и в неоднозначности представления текста над расширенным алфавитом.

---

[x] См.: Д.А.Хаффмен. Метод построения кодов с минимальной избыточностью. – Кибернетический сборник № 3. М., ИЛ, 1961.

Таким образом, можно сформулировать две основные задачи: 1) найти оптимальное расширение естественного алфавита; 2) определить оптимальное представление текста над построенным расширенным алфавитом.

В работе рассматривается совместное решение этих задач, обеспечивающее оптимальное кодирование любых информационных текстов.

Дадим некоторые определения, важные для дальнейшего изложения.

1. Естественным алфавитом $A=(\alpha_1, \alpha_2, \ldots, \alpha_n)$ назовем алфавит естественного или алгоритмического языка. Здесь $\alpha_i \in A$ – символы естественного алфавита; n – длина алфавита.

2. Словом s над алфавитом A назовем упорядоченный набор символов $\alpha_i \in A$

$$S = \alpha_{i_1} \alpha_{i_2} \ldots \alpha_{i_t} ,$$

где t – исходная длина слова s, т.е. число символов

$$\alpha_{i_z} \in A , \quad z = 1,2, \ldots, t , \quad 1 \leq i_z \leq n .$$

3. Текстом T назовем множество слов естественного или алгоритмического языка $\{s_j\}$, $(j=1,2,\ldots, M)$, где M – число слов в T.

Решение задачи оптимального кодирования рассмотрим применительно к любому заданному тексту T.

4. Любое множество слов $B=\{\beta_k\} \supset A$ назовем расширенным алфавитом (в частности, возможно, что $\exists k\, \beta_k \in T$).

5. Множество ΔB_j, порожденное словом $s_j = \alpha_{i_1} \alpha_{i_2} \ldots \alpha_{i_{t_j}}$, назовем расширением алфавита, если

$$\Delta B_j = \{\alpha_{i_k} \alpha_{i_{k+1}} \ldots \alpha_{i_{k+r}}\} , \quad 1 \leq k \leq t_j , \quad k+r \leq t_j .$$

6. Полным расширенным алфавитом B_T, порожденным текстом T, назовем множество

$$B_T = \bigcup_{j=0}^{M} \Delta B_j ,$$

где

$$\Delta B_0 = A .$$

7. Пусть B – некоторый расширенный алфавит. Генерацией слова s_j над B назовем любое представление s_j в виде

$$s_j = \beta_{i_1} \beta_{i_2} \ldots \beta_{i_{t_j}} ,$$

если

$$\forall i_k \beta_{i_k} \in B \cap \Delta B_j .$$

Фиксируя генерацию q_j для каждого $s_j \in T$, получим генерацию текста T^q. Для больших текстов число возможных генераций комбинаторно велико. Очевидно, что каждой генерации текста однозначно соответствует вектор вероятностей

$$P^G = \left(P_1^G, P_2^G, \ldots, P_N^G\right),$$

где N – число символов в алфавите B .

Итак, пусть задан текст T и множество C алфавитов, порождаемых этим текстом:

$$C = \left(B_0, B_1, \ldots, B_T\right);$$

$$B_0 = A; \quad \forall i\, B_i \subset B_T .$$

Тогда задача оптимизации кодового представления текста T заключается в поиске экстремума функции

$$L\left(T^G\right)_{B_i} + L\left(B_i\right)_A \xrightarrow[B_i \subset C, G]{} \min , \qquad (1)$$

где $L\left(T^G\right)_{B_i}$ – кодовая длина G-й генерации текста T над алфавитом $B_i \subset C$; $L\left(B_i\right)_A$ – кодовая длина расширенного алфавита B_i над естественным алфавитом A .

В частном случае, когда алфавит B предварительно определен, задача

$$L\left(T^G\right)_B + L\left(B\right)_A \xrightarrow[G]{} \min \qquad (2)$$

решена совместно с А.Д.Школьниковым.

Следует отметить, что при $B = A$ кодовая длина $L(A)$ пренебрежимо мала по сравнению с кодовой длиной текста $L(T)$. В случае расширенного алфавита $B \supset A$ вес $L\left(B\right)_A$ в целевых функциях (1), (2) может быть весьма значительным.

Рассмотрим некоторую генерацию G^0 текста T над заданным алфавитом B . По определению генерации текста можно вычислить вектор вероятностей P^0 и построить коды Хаффмена для каждого символа $\beta_i \in B$. Тогда каждому слову s_j в исходной генерации G^0 соответствует кодовая длина l_j^0 . Найдем генерацию этого слова, минимизирующую его кодовую длину.

В результате определим генерацию, которой соответствует кодовая длина $l_j^1 \leq l_j^0$. Повторяя эту процедуру для всех слов текста, получим генерацию G^1 с кодовой длиной $L^1 = \sum_{j=1}^{M} l_j^1$. На основании генерации G^1 может быть определен новый вектор P^1 и соответствующие ему коды Хаффмена. В результате применения полученных кодов к генерации G^1 определим кодовую длину

$$\tilde{L}_H^1 = \sum_{j=1}^{M} \tilde{l}_j^1 \leq L^1 = \sum_{j=1}^{M} l_j^1 .$$

(в силу оптимальности кодирования по Хаффмену).

Отыскивая для каждого слова s_j генерацию, минимизирующую его кодовую длину, получаем $l_j^2 \leq \tilde{l}_j^1$ и новую генерацию текста G^2 с длиной

$$L^2 = \sum_{j=1}^{M} l_j^2 \leq \tilde{L}^1 .$$

Продолжая далее этот итеративный процесс, приходим к последовательности генераций $G^1, G^2, G^3, \ldots$, которым соответствует последовательность кодовых длин текста $L^1 \geq L^2 \geq L^3 \geq \ldots$

Конечность этого итеративного процесса очевидна. Изложенный метод итераций, в общем случае, приводит к локальному экстремуму целевой функции (2).

Эффективность предложенного метода определяется, в общем случае, выбором алфавита B и заданием начального приближения целевой функции (2).

Требование уменьшения избыточности текста T влечет за собой стремление к расширению естественного алфавита A. С другой стороны, оптимальность метода итераций приводит, согласно выражению (2), к минимизации принятого расширенного алфавита $B = A + \Delta B_1$.

Кроме того, если учесть эвристичность выбора начального расширения алфавита ΔB, становится понятной возможность получения на каждом шаге итераций символов $\beta_i \in B$ с нулевыми вероятностями. В этой связи условие задачи (2) $B = \text{const}$ может быть реализовано следующим приемом.

Пусть на k-й итерации получена генерация текста G^k и соответствующий вектор вероятностей

$$P^k = (p_1, p_2, \ldots, p_r, p_{r+1}, \ldots, p_m),$$

где

$$p_{r+1} = p_{r+2} = \ldots = p_m = 0 .$$

Применение оптимальной процедуры кодирования по Хаффмену основано на таком векторе P, что $\forall i\, p_i \neq 0$, при этом обеспечивается минимум средней кодовой длины $L_{ср}$ на символ алфавита. Таким образом,

$$L_{ср} = \sum_{i=1}^{r} p_i l_i .$$

Не ухудшая кодирования по Хаффмену, символам $\beta_{r+1}, \beta_{r+2}, \ldots, \beta_m$ присваиваются коды минимально возможной длины при сохранении свойства префиксности на алфавите B. Такой подход обеспечивает на последнем шаге итерационного процесса получение алфавита $B_{кон} \subset B$, соответствующего локальному экстремуму (2), т.е. $\forall \beta_i \in B_{кон}\, p_i \neq 0$.

На основании сказанного наиболее целесообразно было бы в качестве исходного использовать полный расширенный алфавит B_T. Однако реализация метода итераций (в дальнейшем – процедура 1) ввиду чрезвычайно большой длины алфавита B_T практически неосуществима. Поэтому расширенный алфавит B и начальное значение вектора P могут определяться эвристически на основании статистического анализа

заданного текста T. Такой подход, как показывают наши исследования, позволяет получить коэффициент сжатия кодового предотавления текста, равный 1,6–1,9.

Отмеченные недостатки процедуры 1 могут быть значительно ослаблены. Решение задачи (1), т.е. построение оптимального расширенного алфавита и определение генерации текста, минимизирующей его кодовую длину, реализуется следующим образом.

Метод представляет собой многошаговый процесс, на каждом k–м шаге ($k = 1,2,\ldots,M$) которого выполняется процедура 1 с одновременным формированием оптимального k–го расширения алфавита.

Первый шаг метода. Рассмотрим слово $s_1 \in T$ и построим расширение алфавита A.[x] В качестве начального вектора вероятностей примем

$$P^* = (p_1, p_2, \ldots, p_n, p_{n+1}, p_{n+2}, \ldots, p_m),$$

где m – длина алфавита $A + \Delta B_1$; $p_{n+1} = p_{n+2} = \ldots = p_m = 0$.

Правило построения кодов для символов $\beta_{n+1}, \beta_{n+2}, \ldots, \beta_m$ изложено выше.

В результате выполнения процедуры 1 использованные символы $\beta_j \in \Delta B_1$ получат ненулевые финальные вероятности. Итак, пусть P_1^{φ} – вектор финальных вероятностей. Рассмотрим подмножество B_1 алфавита $A + \Delta B_1$, которому соответствуют отличные от нуля компоненты P_1^{φ}. Построенный алфавит B_1 локально оптимален, поскольку он получен в результате применения к полному расширенному алфавиту $A + \Delta B_1$ локально экстремальной процедуры 1.

Общий (k–й) шаг метода. Пусть на ($k-1$)–м шаге построен локально оптимальный алфавит B_{k-1} и отвечающий ему стохастический вектор P_{k-1}. Построим алфавит $B_{k-1} + \Delta B_k$, приняв за начальное приближение вектор вероятностей

$$P_k^* = (p_1^{k-1}, p_2^{k-1}, \ldots, p_r^{k-1}, p_1^k, p_2^k, \ldots, p_{m_k}^k),$$

где

$$m = \dim P_{k-1};$$

$$p_1^k = p_2^k = \ldots = p_{m_k}^k = 0.$$

В результате выполнения процедуры 1 получим финальный вектор P_k^{φ} и соответствующий ему локально экстремальный алфавит B_k.

Таким образом, после выполнения M–го шага будет получен локально оптимальный для текста T расширенный алфавит B_M^*, оптимальная генерация текста, финальный вектор P_M^{φ} и соответствующие ему оптимальные коды Хаффмена.

Изложенный метод позволяет полностью решить, поставленную в настоящей работе задачу (1) и, тем самым, определить кодовое представление информационных текстов, минимизирующее необходимый объем памяти ЭВМ.

---

[x] В зависимости от возможностей ЭВМ целесообразно на каждом шаге получать расширение алфавита, порожденное группой слов.