Университет ИТМО
Факультет ПИиКТ

Прикладная математика
Лабораторная работа №1
«Вычисление энтропии Шеннона»

Нестеров Дали Константинович
Группа P3302

Санкт-Петербург
2019

Цель работы:

Получить практические навыки решения задач на количественное измерение информационного объема текстовой информации.

Задание:

1. Реализовать процедуру вычисления энтропии для текстового файла. Требования к текстовому файлу:

• документ на английском языке

• размер текстового файла - 20-60кБ

• документ должен быть осмысленным

В процедуре необходимо подсчитывать частоты появления символов (прописные и заглавные буквы не отличаются, знаки препинания рассматриваются как один символ, пробел является самостоятельным символом), которые можно использовать как оценки вероятностей появления символов. Затем вычислить величину энтропии. Точность вычисления -- 4 знака после запятой. Обязательно предусмотреть возможность ввода имени файла, для которого будет вычисляться энтропия.

2. Проверить запрограммированную процедуру на нескольких файлах и заполнить таблицу 1.1. вычисленными значениями энтропии

3. Вычислить значение энтропии для тех же файлов, но с использованием частот вхождений пар символов и заполнить таблицу 1.2

4. Проанализировать полученные результаты.

Описание входных данных:

В качестве входных данных были выбраны отрывки из трех произведений на английском языке: «Nightfall», «Infinite jest» и «Evgeny Onegin»

Решение поставленной задачи:

Index.html(body):

```
<body>
<h2>Upload text file to calculate entropy</h2>
<input type="file" id="file-input">
<div class="item_small" id="entropy-H"></div>
<div class="item_small" id="entropy-H-star"></div>
<table class="item_small" id="result">
    <tr>
        <td>Symbol</td>
        <td>Probability P(x<sub>i</sub>), bit</td>
        <td>Entropy H(x<sub>i</sub>), bit</td>
    </tr>
</table>
<script src="script.js"></script>
</body>
```

Script.js():

```
document.getElementById('file-input').oninput = function () {
    const file = document.getElementById('file-input').files[0];
    if (file != null) { // don't do anything if no file has been chosen
        document.getElementById("result").innerHTML = "<tr>\n" +
            "         <td>Symbol</td>\n" +
            "         <td>Probability P(x<sub>i</sub>), bit</td>\n" +
            "         <td>Entropy H(x<sub>i</sub>), bit</td>\n" +
            "     </tr>";
        document.getElementById("entropy-H").innerText = "";
        document.getElementById("entropy-H-star").innerText = ""; // reset results
        const fileReader = new FileReader();
        fileReader.onload = function (e) {
            let p_i = new Map(); // keys: characters, values: their probabilities
            let h_i = new Map(); // keys: characters, values: their entropy
            let H = 0; // entropy for the whole text
            let HStar = 0; // linked entropy
            let sum = 0; // characters count
            let prevChar = " "; // used for calculating linked entropy
            let p_pairs = new Map(); // keys: pairs of characters, values: their probabilities
            // p_pair stores probabilities of encountering a char if a certain other char has appeared before
            const text = e.target.result;
            for (let i = 0; i < text.length; ++i) {
                let char = text[i].toUpperCase(); // ignore case of letters
                if (char.match("[\t\r\n\f]")) continue; // we don't include any space characters but whitespace
                if (!char.match("^[A-Z0-9 ]$")) char = "."; // all other symbols are treated as punctuation symbol
                if (p_i.get(char) === undefined) { // Map initializes with undefined values
                    p_i.set(char, 0); // but we have to convert them to 0 to increment values
                }
                p_i.set(char, p_i.get(char) + 1); // whenever we encounter a char we increment their "probability"

                if (sum !== 0) {
                    const pair = prevChar + char;
                    if (p_pairs.get(pair) === undefined) {
                        p_pairs.set(pair, 0);
                    }
                    p_pairs.set(pair, p_pairs.get(pair) + 1); // increment probability for a pair of chars
                }
                sum++;
                prevChar = char;
            }

            for (const char of Array.from(p_i.keys()).sort()) {
                p_i.set(char, p_i.get(char) / sum);  // actual probability = char count / text length
                h_i.set(char, Math.log2(1 / p_i.get(char))); // entropy of a character = (log(1/p(xi)))
                H -= p_i.get(char) * Math.log2(p_i.get(char)); //entropy = -SUM( p(xi) * log(p(xi)) )
                const newRow = document.getElementById("result").insertRow(); // add a row to the table
                newRow.innerHTML = `<td>${char}</td>
                                    <td>${p_i.get(char).toFixed(4)}</td>
                                    <td>${h_i.get(char).toFixed(4)}</td>`
            }
            for (const [pair, p] of p_pairs) {
                p_pairs.set(pair, p / (sum - 1)); // actual probability = pair count / (text length - 1)
                HStar -= p_pairs.get(pair) * p_i.get(pair[0]) * Math.log2(p_pairs.get(pair));
                // linked entropy = -SUM( p(xi/xj) * p(xj) * log(p(xi/xj)) )
            }
            document.getElementById("entropy-H").innerText = `Entropy H(X) = ${H.toFixed(4)} bit`;
            document.getElementById("entropy-H-star").innerText = `Entropy H*(X) = ${HStar.toFixed(4)} bit`;
        };
        fileReader.readAsText(file);
    }
};
```

Результат работы программы:

Таблица 1.2. для файла nightfall_demo.txt

Symbol	Probability $P(x_i)$, bit	Entropy $H(x_i)$, bit
	0.1709	2.5484
.	0.0501	4.3202
0	0.0000	14.4449
1	0.0000	15.4449
2	0.0001	12.8600
4	0.0000	14.4449
5	0.0002	12.4449
6	0.0000	14.4449
7	0.0002	12.6376
A	0.0598	4.0645
B	0.0109	6.5201
C	0.0181	5.7849
D	0.0330	4.9204
E	0.0965	3.3738
F	0.0168	5.8980
G	0.0142	6.1343
H	0.0473	4.4005
I	0.0515	4.2781
J	0.0012	9.6900
K	0.0060	7.3788
L	0.0329	4.9263
M	0.0174	5.8450
N	0.0530	4.2379
O	0.0626	3.9984
P	0.0126	6.3131
Q	0.0005	10.8600
R	0.0421	4.5699
S	0.0480	4.3808
T	0.0802	3.6404
U	0.0255	5.2952
V	0.0073	7.1051
W	0.0199	5.6521
X	0.0013	9.5380
Y	0.0191	5.7068
Z	0.0006	10.7445

Таблица 1.2.

Файл:	nightfall_demo.txt	infinite_jest_demo.txt	evgeny_onegin_demo.txt
Энтропия H(X), бит	4.1837	4.2148	3.9986
Энтропия H*(X), бит	0.5123	0.5103	0.6284

Вывод: в ходе лабораторной работы была реализована программа для расчета энтропии и были получены практические навыки решения задач на количественное измерение информационного объема текстовой информации. Энтропия везде была примерно одинаковой. Энтропия с учетом пар символов получается значительно меньше, чем без и условие 0 <= H*(X) <= H(X) соблюдается.