Instituto de Matemática e Estatística

Hélio Hideki Assakura Moreira

EP4 - Artista Aleatório

São Paulo

2014

Objetivo do EP:

Estudar a aproximação da estrutura estatística de um texto usando o **Modelo de Markov**. Para isso, construiremos uma Cadeia de Markov, que é um processo estocástico onde a mudança de estado depende apenas de seu estado atual. No EP, o estado atual são os últimos **k** caracteres. O próximo caracter é gerado aleatoriamente, usando as probabilidades do modelo de Markov.

Teste Realizados:

Para a realização dos testes, será usado o livro “The Iliad”, de Homero.

Aqui serão colocados apenas os tempos obtidos para cada teste. Os textos gerados serão colocados em um arquivo anexado juntamente com o Relatório e com o arquivo fonte do EP.

k = 0, t = 100:

```
real    0m1.663s
user    0m0.100s
sys     0m0.000s
```

k = 1, t = 1000:

```
real    0m2.101s
user    0m0.142s
sys     0m0.004s
```

k = 3, t = 1000:

```
real    0m2.260s
user    0m0.305s
sys     0m0.008s
```

k = 3, t = 10000:

```
real    0m2.539s
user    0m0.308s
sys     0m0.004s
```

k = 50, t = 1000:

```
real    0m5.336s
user    0m1.846s
sys     0m0.052s
```

k = 100, t = 1000:

```
real    0m5.142s
user    0m2.004s
sys     0m0.088s
```

k = 100, t = 1.000.000:

```
real    0m10.234s
user    0m4.103s
sys     0m0.146s
```

k = 1000, t = 10000:

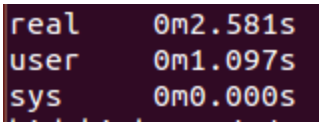
```
real    0m9.420s
user    0m4.790s
sys     0m0.304s
```

k = 5000, t = 100000:

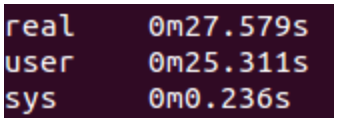
```
real    0m23.263s
user    0m17.406s
sys     0m1.239s
```

Para textos maiores, como o “Novo dicionário da língua portuguesa by Cândido de Figueiredo”, (http://www.gutenberg.org/ebooks/31552), que contém cerca de 12,6 milhões de caracteres, demora mais tempo:

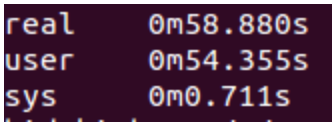
k = 0, t = 5000



k = 10, t = 5000



k = 100, t = 10000



Conclusões do Teste:

Para textos pequenos, a diferença do tempo de execução é bem pequena, mesmo variando **k** e **t**. Quando o texto começa a aumentar de tamanho, o tempo de execução tende a aumentar quanto maior o **k** e o tamanho do texto. Como visto nos exemplos acima, a diferença começa a ficar grande, como quando k = 0 e k = 5000, por exemplo. Além disso, com valores bem grandes de **t**, o tempo também aumenta. Isso deve-se ao fato de que o numero de comparações e a cópia de string começa a ficar muito grande, demandando muito tempo para ser realizada. Apesar de que quanto maior o **k**, mais semelhante o texto fica do original, é menos provável que ocorra a mesma sequência de **k** caracteres. Assim, a ABB começa a ficar maior, demorando mais para realizar a busca, cálculo da frequência relativa e acumulada, sorteio de **k** caracteres etc.

Quanto maior o texto, consequentemente, maior será a árvore. Se **k** também for alto o bastante para que não haja tantas ocorrências repetidas, acontece o que pôde se ver no exemplo em que foi usado o ***Novo dicionário da língua portuguesa***, onde o tempo de execução é extremamente maior que ***Ilíada***, com praticamente os mesmos parâmetros, por exemplo.