Algoritmos Avançados
3rd Project — Most Frequent Letters
Csurös' counter
Lossy-Count

Eva Pomposo Bartolomeu

Resumo **- Neste relatório apresenta-se análise formal e experimental do problema proposto na cadeira Algoritmos Avançados, identificar as letras mais frequentes em ficheiros de texto usando diferentes métodos, e avaliar a qualidade das estimativas em relação às contagens puras. As análises são realizadas a três métodos de resoluções diferentes, o Exact Counter, o Csurös' counter e o Lossy-Count**

Abstract **- This report presents a formal and experimental analysis of the proposed problem in the Advanced Algorithms course, identifies the most frequent letters in text files using different methods, and evaluates the quality of the estimates in relation to pure counts. The analyzes are carried out using three different resolution methods, the Exact Counter, the Csurös' counter and the Lossy-Count.**

I. INTRODUÇÃO

No âmbito da cadeira de Algoritmos Avançados, foi-me proposto um trabalho acerca da identificação das letras mais frequentes em ficheiros de texto.

Perante este problema, e de modo a resolver o problema proposto, foi pedido a implementação do método exact counters, da implementação e testagem do método Csurös' counter (approximate counters) e por fim, o desenvolvimento do algoritmo Lossy-Count (algorithm to identify frequent items in data streams).

O método exact counters é a contagem pura de cada letra num dado ficheiro.

Csurös' counter [1] é um algoritmo para prever a contagem de cada letra em um data stream. Estas contagens não são precisas, mas através de vários testes é possível chegar a resultados bons.

Csurös' counter usa uma representação de floating-point, com uma parte de fixed-point e uma parte de floating-point. A parte do fixed-point é incrementada cada vez que uma letra aparece, e a parte floating-point é incrementada com uma probabilidade que diminui ao longo do tempo, usando assim o algoritmo uma quantidade fixa de memória.

A precisão das estimativas é gerida pela precisão da representação do floating-point e pelo número de bits usados para a parte do fixed-point.

Lossy Count [2] é um algoritmo de streaming para prever a contagem de cada letra em um data stream, estas contagens também não são precisas, são um aproximação.

O algoritmo é baseado no conceito "sketch", uma estrutura de dados probabilística que guarda um resumo do data stream. As estimativas dependem do tamanho do "sketch" e da distribuição dos dados no data stream.

Após o desenvolvimento das abordagens realizar uma análise da eficiência computacional e das limitações das abordagens desenvolvidas. Avaliar a qualidade das estimativas e o desempenho dos contadores aproximados e do algoritmo de fluxo de dados, em relação às contagens exatas.

Os objetivos deste relatório são:

- Identificar as letras mais frequentes em ficheiros de texto usando diferentes métodos, e avaliar a qualidade das estimativas em relação às contagens puras.

- Desenvolver um pré-processamento de ficheiros de texto adequado para o Project Gutenberg;
- Implementar os métodos exact counters, Csurös' counter e Lossy-Count como resolução ao problema;
- Realizar vários testes, repetindo algumas vezes as approximate counts.
- Estimar as 3, 5 e 10 letras mais frequentes, executando o data stream algorithm.
- Fazer uma análise da eficiência computacional e das limitações das abordagens desenvolvidas.
- Comparar o desempenho dos approximate counters e do data stream algorithm, entre si e em relação às exact counts.

II. ANÁLISE FORMAL DOS ALGORITMOS

A. Exact counter

O método exact counter percorre cada letra do texto processado num ciclo. Em cada ciclo incrementa a contagem da respectiva letra. As contagens das letras são guardadas num defaultdict.

B. Csurös' counter

O algoritmo Csurös' counter desenvolvido percorre cada letra do texto. Em cada iteração incrementa a contagem da respectiva letra, usando a função *fp_increment*. As contagens das letras são guardadas num dicionário.

A função *fp_increment* recebe uma contagem como entrada e retorna uma nova contagem. Primeiro divide a contagem de entrada por M (um valor fixo que será explicado melhor mais à frente) . De seguida, entra num loop onde 1/M é a probabilidade de retornar a contagem de entrada e 1-1/M é a probabilidade de retornar a contagem de entrada mais um. Ou seja, a contagem só é incrementada com uma probabilidade de 1-1/M.

C. Lossy-Count

O algoritmo Lossy-Count cria um defaultdict para armazenar as contagens de cada letra. Depois percorre cada letra do texto processado num ciclo. No ciclo, o algoritmo incrementa a contagem da respectiva letra, e além disso, após essa incrementação, verifica se o número de letras no defaultdict é maior que o valor de k, se sim, remove a letra com a contagem mais baixa.

Este algoritmo retorna uma estimativa das contagens das k letras mais frequentes.

III. DADOS PARA AS EXPERIÊNCIAS COMPUTACIONAIS

De modo a testar os algoritmos desenvolvidos foram usados arquivos de texto de obras literárias, em diferentes idiomas, do Projeto Gutenberg [9]. Usei a obra "Nazareth: a morality in one act" [8] no idioma inglês a obra "Romeo and Juliet" em inglês [3], francês [4], finlandês [7], alemão [5] e holandês [6] e a obra "The Lusiad or The Discovery of India, an Epic Poem" [10] no idioma inglês.

Antes de executar qualquer algoritmo com um texto é necessário fazer o pré-processamento do texto. Cada arquivo de texto foi processado, onde removemos os cabeçalhos, removemos todas as stop-words e sinais de pontuação e convertemos todas as letras em maiúsculas.

IV. RESULTS

Em cada arquivo de texto o método exact counter foi executado uma vez. Além disso, calculei o número total de bits precisos para guardar as contagens devolvidas por este método.

O algoritmo Csurös' counter foi testado 20 vezes, com um valor de M de 100. Após vários execuções, o valor 20 para o número de testes e o valor 100 para a variável M, foram os melhor valores definidos de modo a que o algoritmo tenha um tempo de execução parecido ao dos outros algoritmos, de modo a que dê resultados aproximados para os ficheiros de texto em uso, e para o código desenvolvido.

Ainda para o algoritmo Csurös' counter calculei os erros absolutos e relativos (valor mais baixo, valor mais alto, valor médio), determinei a média das contagens de cada letra nos testes realizados e a média do número total

de bits usados para guardar as contagens devolvidas.

Por fim, o Lossy-Count foi executado 3 vezes, de modo a estimar as 3/5/10 letras mais frequentes, aqui determinei também os erros absolutos e relativos e o número total de bits utilizados para guardar as contagens em cada execução.

A. Contagens e ordem

Como já dito anteriormente, o método Exact Counter devolve sempre as contagens corretas, enquanto que os outros dois métodos não, estes devolvem uma estimativa. Nas Figuras 1 a 5 pode se ver o resultado dos algoritmos desenvolvidos para o ficheiro Nazareth.txt.

Em relação ao valor das contagens de cada letra, podemos ver que o método Lossy-Count é mais preciso, dá valores mais corretos, que o Csurös' counter. Por exemplo, na Figura 5 e 6 podemos ver que o Lossy-Count estima que a letra E aparece 1745 vezes, um valor praticamente igual a 1746, que é a contagem determinada no Exact Counter (Figura 1). Já o Csurös' counter estima um valor médio 415.95 (Figura 2), um valor ainda bastante diferente.

Quanto maior for o valor de k, no método Lossy-Count, mais precisos são os resultados, tal como podemos ver nas Figuras 3 e 5, com k = 3 a contagem da letra A não é tão precisa como com o k = 10. Apesar disso, a contagem da letra A com o método Lossy-Count com k = 3, continua mais preciso que o método Csurös' counter (Figura 2).

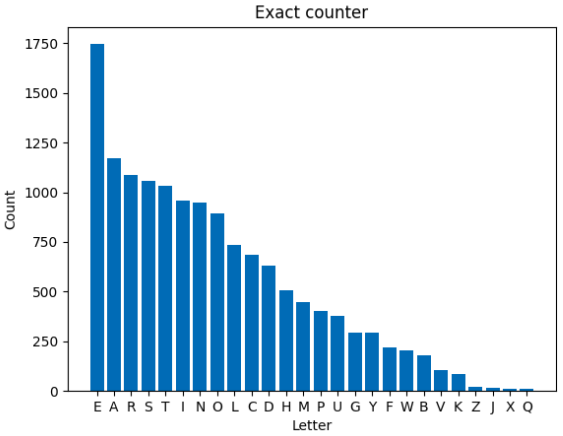

Fig. 1 - Contagens de cada letra obtidas pelo método Exact counter com o ficheiro Nazareth.txt.

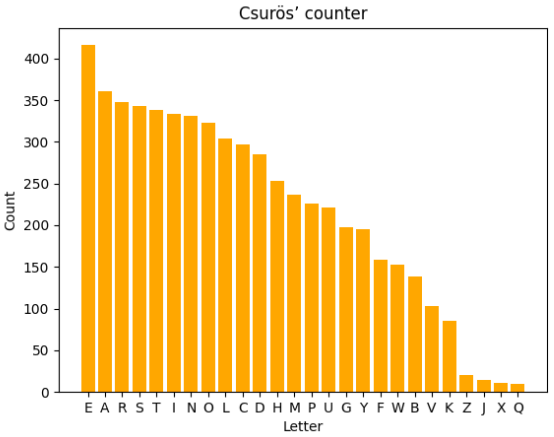

Fig. 2 - Médias das contagens de cada letra estimadas nos 20 testes realizados ao método Csurös' counter com o ficheiro Nazareth.txt.

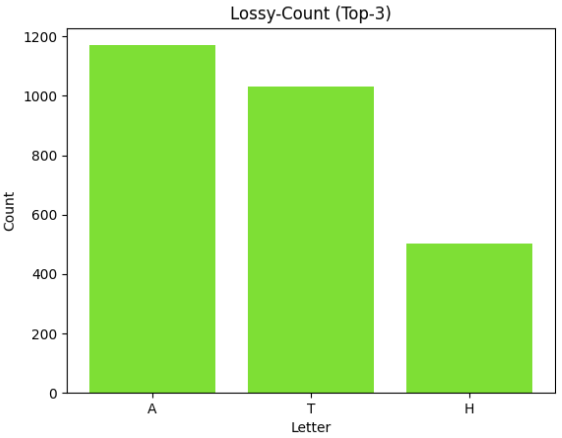

Fig. 3 - Estimativa das contagens das 3 letras mais frequentes obtidas pelo método Lossy-Count com o ficheiro Nazareth.txt.

Fig. 4 - Estimativa das contagens das 5 letras mais frequentes obtidas pelo método Lossy-Count com o ficheiro Nazareth.txt.

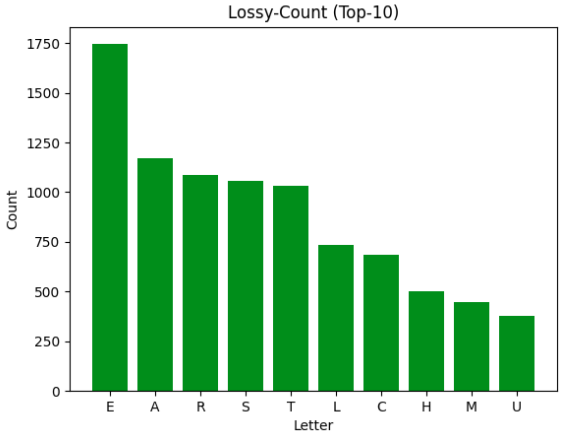

Fig. 5 - Estimativa das contagens das 10 letras mais frequentes obtidas pelo método Lossy-Count com o ficheiro Nazareth.txt.

Em relação à identificação das letras mais frequentes e à respectiva ordem, podemos observar que o método Csurös' counter, comparado com o algoritmo Lossy-Count, identifica e posiciona melhor as letras mais frequentes, isto é, de acordo com o método Exact Counter. Na tabela 1, podemos analisar que o Csurös' counter identifica e posiciona as 10 letras mais frequentes de igual maneira ao do método Exact Counter. Já o algoritmo Lossy-Count, com k = 10, identifica e posiciona da maneira correta apenas as 5 letras mais frequentes, as restantes não o faz corretamente.

Além disso, no Lossy-Count, quanto menor o k pior será a identificação e o posicionamento das letras mais frequentes, tal como podemos ver na Figura 3, que identifica a letra A, T e H como as letras mais frequentes, e com essa ordem, o que está errado.

Letter	Posição no Exact Counter	Posição no Csurös' counter	Posição no Lossy-Count (k=10)
E	1	1	1
A	2	2	2
R	3	3	3
S	4	4	4
T	5	5	5
I	6	6	
N	7	7	
O	8	8	
L	9	9	6
C	10	10	7
H			8
M			9
U			10

Tabela 1 - As 10 letras mais frequentes e a respectiva posição obtidas nos 3 métodos com o ficheiro Nazareth.txt.

Apesar de não mostrar em nenhum gráfico o algoritmo Csurös' counter com diferentes valores para o número de testes e diferentes valores para a variável M, foram realizadas várias experiências com diferentes valores dos mesmos. Dessas experiências, pode-se concluir que quanto maior o valor de M e maior o número de testes, mais precisos são os resultados, e também melhor será a identificação e o posicionamento das letras mais frequentes, no entanto, também maior será o tempo de execução.

B. Erros absolutos e relativos

Uma vez que os erros absolutos e relativos calculados são sobre os valores das contagens de cada letra, os erros obtidos no Lossy-Count são

menores que no Csurös' counter (Tabela 2), o que vai de encontro com as conclusões retiradas anteriormente. Pois concluímos atrás que o Lossy-Count é mais preciso, que o Csurös' counter, logo se é mais preciso os erros dos valores das contagens têm que dar valores mais baixos.

	Csurös' counter	Lossy-Count (k=3)	Lossy-Count (k=5)	Lossy-Count (k=10)
Average absolute error	323.67	2.67	3.60	0.70
Minimum absolute error	0	0	0	0
Maximum absolute error	1344	5	11	5
Average relative error	0.40%	0%	0%	0%
Minimum relative error	0%	0%	0%	0%
Maximum relative error	0.77%	0.01%	0.01%	0.01%

Tabela 2 - Erros absolutos e relativos determinados com o ficheiro Nazareth.txt.

Com esta tabela, podemos concluir que nos 3 algoritmos pelo menos uma letra teve a contagem igual ao do método Exact Counter, pois o Minimum absolute error e o Minimum relative error deram respectivamente 0 e 0%.

O erro máximo obtido pelo o Csurös' counter, foi de 1344, um erro relativo de 0.77%, já no Lossy-Count foi de 11, um erro relativo de 0.01%. O erro máximo do Lossy-Count foi muito mais baixo que no Csurös' counter, tal como esperado.

A média de erros obtidos pelo o Csurös' counter, foi de 323.67, um erro relativo de 0.40%, já no Lossy-Count foi de 3.60, um erro relativo de 0%. A média dos erros no Lossy-Count foi muito mais baixo que no Csurös' counter, tal como esperado.

C. Contagens em idiomas diferentes

Na Figura 6 são retratadas as contagens de cada letra obtidas pelos 3 métodos, com o ficheiro Romeo and Juliet e com os seguintes idiomas: holandês, inglês, finlandês, francês e alemão.

Em todas as línguas, a letra mais frequente é o E, com exceção do idioma finlandês. Já a segunda letra mais frequente é diferente em todos os idiomas menos no inglês e alemão, que têm como segunda letra mais frequente o T.

O idioma holandes, finlandês e alemão têm a letra X como a menos frequente. A letra Q é também das letras menos frequentes nestes gráficos.

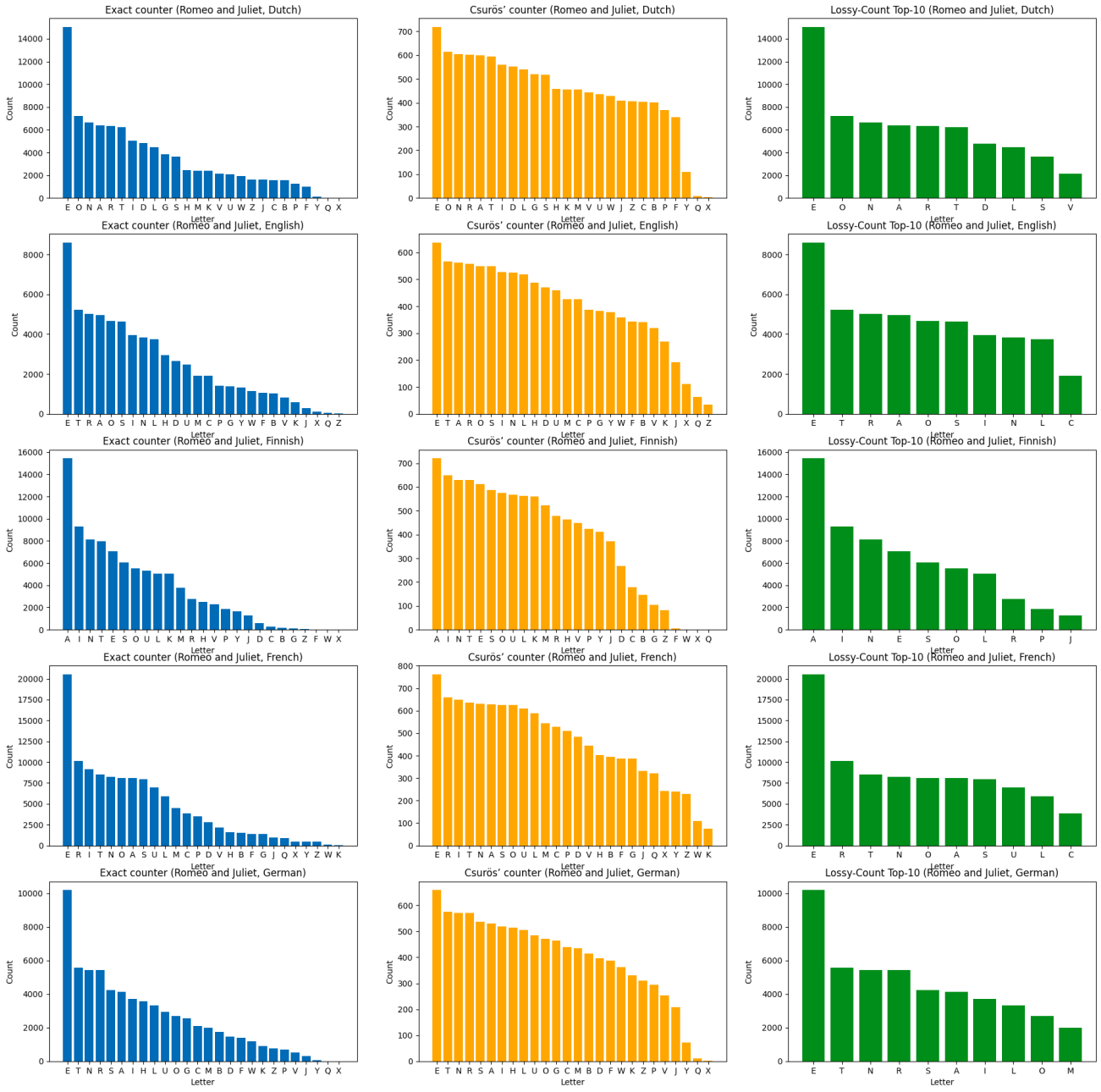

Fig. 6 - Contagens de cada letra obtidas pelos 3 métodos, com o ficheiro Romeo and Juliet em várias línguas.

C. Desempenho

A Tabela 3 apresenta o número total de bits usados para guardar as contagens, com o ficheiro lusiadas.txt.

O método Exact Counter é o que precisa de mais bits para guardar as contagens. Para o exemplo do ficheiro lusiadas.txt é usado 350 bits, enquanto que no Csurös' counter só são usados 253 bits (uma grande diferença), e no Lossy-Count são usados 44, 73, 147 bits para o k=3, k=5 e k=10 respectivamente.

Em termos de números de bits usados, não dá para dizer se o Csurös' counter é melhor que o Lossy-Count, pois depende dos valores de k para o segundo método, e depende dos valores de M para o primeiro.

Quanto maior o valor de k no Lossy-Count, e quanto maior o valor de M no Csurös' counter, maior será o números de bits usados para guardar as contagens.

Exact Counter	Csurös' counter	Lossy-Count (k=3)	Lossy-Count (k=5)	Lossy-Count (k=10)
350 bits	253 média de bits	44 bits	73 bits	147 bits

Tabela 3 - Número total de bits usados para guardar as contagens, com o ficheiro lusiadas.txt.

VI. Conclusion

O problema de identificação das letras mais frequentes em ficheiros de texto pode ter vários métodos de resolução.

A quantidade de memória usada, e os tempos de execução, são valores que podem ser extremamente grandes na abordagem Exact Counter, quando são usados textos muito grandes. Em resposta a este problema temos o método Csurös' counter e Lossy-Count, que apresentam respostas com uma quantidade de memória pequena, apesar de não nos darem sempre a contagem de letras correta, dão-nos uma contagem bem aproximada.

Quanto maior o valor de M e maior o número de testes no Csurös' counter, e quanto maior for o valor de k no método Lossy-Count, mais precisos são os resultados, e também melhor será a identificação e o posicionamento das letras mais frequentes.

No entanto, também maior será o tempo de execução e os números de bits usados para guardar as contagens. Portanto é preciso chegar a um equilíbrio, para que a memória e o tempo de execução não sejam um problema, e para que as contagens estejam bem aproximadas.

O método Lossy-Count dá contagens mais precisas, enquanto que o método Csurös' counter identifica e posiciona melhor as letras mais frequentes.

É de realçar que não há um algoritmo melhor que o outro, apenas em certas situações uma abordagem pode ser mais adequada que a outra.

O algoritmo Exact Counter é adequado para textos não muito grandes, e quando queremos saber respostas 100% corretas. Já o Csurös' counter é apropriado para textos muito grandes, e quando nos interessa apenas quais são as letras mais frequentes e a sua ordem. Por fim, o Lossy-Count é adequado para textos muito grandes, e quando nos interessa saber apenas quais sao as K letras mais frequentes, e o número de vezes que aparecem no texto.

References

[1] https://arxiv.org/abs/0904.3062

[2] https://en.wikipedia.org/wiki/Lossy_Count_Algorithm

[3] https://www.gutenberg.org/ebooks/1513

[4] https://www.gutenberg.org/ebooks/18143

[5] https://www.gutenberg.org/ebooks/6996

[6] https://www.gutenberg.org/ebooks/49880

[7] https://www.gutenberg.org/ebooks/15643

[8] https://www.gutenberg.org/ebooks/69644

[9] https://www.gutenberg.org/

[10] https://www.gutenberg.org/ebooks/32528