Uso de Árvores de Sufixos para Implementação de String Kernel e Aplicação na Predição do Genoma Humano

Using Suffix Trees for String Kernel Implementation and Application in Human Genome Prediction

José Victor Piccin[1], André Yoshiaki Kashiwabara[2]

RESUMO

Problemas relacionados à classificação e predição são desafios cruciais no campo de *aprendizado de máquina*, exigindo abordagens sofisticadas para alcançar métricas de avaliação de alta qualidade e a manipulação de dados complexos. Essas preocupações também são de grande relevância na análise de sequências biológicas, como o DNA e o mRNA. Nesse contexto, a busca por métodos eficazes é fundamental. Uma abordagem interessante é a utilização de *métodos de kernel*, que se diferenciam dos métodos convencionais ao medir as distâncias entre objetos por meio de funções kernel. Esses métodos oferecem soluções valiosas além de permitirem um processo conhecido como kernelização. O objetivo do trabalho está na implementação e validação de um *string kernel* denominado *Fast String Kernel*, que possui complexidade temporal O(n). Baseado em uma estrutura de dados chamada de árvore de sufixos além de utilizar o *matching statistics* um algoritmo de busca e correspondência. Foram conduzidos experimentos de predição para validar o kernel, utilizando amostras de RNAs do genoma humano, onde os resultados alcançados, neste dataset, demonstram predições significativas com uma taxa de acurácia média de 87,6% (±0,052), uma precisão de 92,4% (±0,068) e 81,6% (±0,108) de recall.
PALAVRAS-CHAVE: Aprendizado de máquina; mRNA; Métodos de Kernel; Predição; Kernel de String.

ABSTRACT

Problems related to classification and prediction are crucial challenges in the field of *machine learning*, requiring sophisticated approaches to achieve high-quality evaluation metrics as well as handling of complex data. These concerns are also highly relevant in the analysis of biological sequences, such as DNA and mRNA. In this context, the search for effective methods is essential. An interesting approach is the use of *kernel methods*, which differ from conventional methods by measuring distances between objects through kernel functions. These methods offer valuable solutions and enable a process known as kernelization. The objective of the work is the implementation and validation of a string kernel called Fast String Kernel, which has a time complexity of O(n). It is based on a data structure called a suffix tree and uses matching statistics, a search and matching algorithm. Prediction experiments were conducted to validate the kernel using samples of RNAs from the human genome, where the results achieved in this dataset demonstrate significant predictions with an average accuracy rate of 87.6% (±0.052), a precision of 92.4% (±0.068), and a recall of 81.6% (±0.108).
KEYWORDS: Machine Learning; mRNA; Kernel Methods; Prediction; String Kernel.

[1] Bolsista da UTFPR. Universidade Técnológica Federal Do Paraná, Cornélio Procópio, Paraná, Brasil E-mail: josevictor@alunos.utfpr.edu.br. ID Lattes: http://lattes.cnpq.br/8566959150351982.

[2] Docente no Departamento Acadêmico de Computação e Programa de pós-graduação em Bioinformática. Universidade Técnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil. E-mail: kashiwabara@utfpr.edu.br. ID Lattes: http://lattes.cnpq.br/3194328548975437.

INTRODUÇÃO

O uso de *aprendizado de máquina* para a predição e análise de dados tem sido amplamente empregado em diversas áreas do conhecimento. Entretanto, nem todos os métodos são adequados para lidar com dados complexos, como dados não lineares (HOFMANN; SCHÖLKOPF; SMOLA, 2008). Na área da *bioinformática*, essa problemática também se faz presente, uma vez que a maioria dos métodos convencionais de análise de dados moleculares biológicos foi projetada para tratar de estruturas de dados simples. Com o propósito de abordar essa questão, será apresentado um método que propõe uma abordagem para lidar com dados complexos (SCHÖLKOPF; TSUDA; VERT, 2004). A utilização desses métodos resulta em contribuições importantes para otimizar questões relacionadas às complexidades de tempo e espaço.

No contexto da análise de sequências biológicas, é crucial compreender que essas sequências consistem em cadeias de nucleotídeos. Em sequências de DNA, há quatro tipos de nucleotídeos (A, C, G e T), enquanto no mRNA, um deles é substituído por U (uracilo). A análise envolve a segmentação das sequências e a comparação para identificar correlações por meio da medição das associações. Basicamente, sequências biológicas podem ser tratadas como *strings*, onde cada caractere representa um nucleotídeo (SCHÖLKOPF; TSUDA; VERT, 2004). Este trabalho visa implementar e validar um *string kernel* chamado *Fast String Kernel* para classificar amostras de RNA do genoma humano. O objetivo principal é demonstrar a utilidade e eficácia do kernel e dos métodos de kernel em problemas de predição de dados biológicos, por meio de resultados de validação.

FUNDAMENTAÇÃO TEÓRICA

MÉTODOS DE KERNEL E KERNELS POSITIVOS DEFINIDOS

Os Métodos de Kernel são técnicas usadas em *aprendizado de máquina* e *inferência empírica* baseada em kernels, que podem ser vistos como funções que medem a similaridade entre objetos. Uma vantagem desses métodos está na possibilidade de realizar análises estatísticas sem a necessidade de criar representações individuais para cada objeto, em vez disso, os objetos são comparados em pares. Os kernels abordados são denominados positivos definidos, nos quais a matriz K resultante deve ser simétrica positiva definida. Um exemplo de kernel definido positivo é o kernel linear, conhecido por realizar o cálculo do produto interno entre dois vetores. Sendo sua necessidade de utilizar vetores sua maior limitação, porém essa limitação pode ser superada ao substituir o espaço vetorial X por um espaço vetorial de Hilbert F, conhecido como *Feature Space*. Onde, os objetos são substituídos por seus vetores (SCHÖLKOPF; TSUDA; VERT, 2004).

Esse processo pode ser generalizado, sendo conhecido como *Kernel Trick*, permitindo a incorporação de kernels em qualquer algoritmo existente que opere em função de vetores. O processo de incorporar o *Kernel Trick* em um algoritmo é conhecido como Kernelização (SCHÖLKOPF; TSUDA; VERT, 2004). A kernelização pode trazer benefícios, como a redução das complexidades temporais. Isso pode ser observado na versão kernelizada *KL-Transformer* do modelo de linguagem *Transformer*, que traz a redução da complexidade temporal de quadrática para linear (CHOWDHURY et al., 2022).

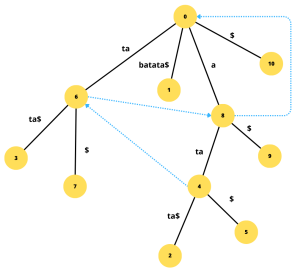
FAST STRING KERNEL, ÁRVORE DE SUFIXOS DE UKKONENS E MATCHING STATISTICS

Fast String Kernel ou FSK, como será chamado o *string kernel* implementado. Trata-se de um kernel desenvolvido com o intuito de determinar a correspondência de objetos discretos, tais como *strings* ou árvores, em tempo linear O(n). Baseado em pesos onde, por meio de um único kernel, é viável alcançar o desempenho equivalente ao de outros kernels. O kernel funciona com base na busca de subpalavras utilizando um algoritmo de busca *matching statistics* aplicado a uma árvore de sufixos. O *FSK* é determinado pela Equação 1 (VISHWANATHAN; SMOLA, 2002).

$$k(x,y) = \sum_{i=1}^{|x|} val(x[i:vi]) = \sum_{i=1}^{|x|} val(ci) + lvs(ceil(x[i:vi]))W(y,x[i:vi]) \quad (1)$$

A árvore de sufixos é uma estrutura de dados que representa todos os sufixos de uma palavra (UKKONEN, 1995). A árvore resultante para uma palavra y é denotada por S(y) (VISHWANATHAN; SMOLA, 2002). O modelo utilizado foi proposto por Ukkonen, que envolve a construção da árvore de sufixos em tempo O(n). Onde, a árvore é construída à medida que cada letra é lida, adicionando essas letras aos nós antes de avançar para a próxima letra ou fase (UKKONEN, 1995), além de adicionar ligações entre os nos internos chamados de *suffix links*. A Figura 1 demonstra o formato da árvore de sufixos para a palavra “batata”. O caractere "$"é usado exclusivamente para indicar o fim de uma palavra e evitar a criação de sufixos fora do alfabeto da palavra (VISHWANATHAN; SMOLA, 2002).

Figura 1 – Árvore de Sufixos para a palavra “Batata”



Fonte: Desenvolvido pelos autores com base em: https://brenden.github.io/ukkonen-animation/.

O algoritmo *Matching Statistics*, baseado em árvores de sufixos, realiza busca e correspondência de padrões em tempo O(n). Tornando possível percorrer a árvore de sufixos S(y) ao mesmo tempo em que se compara os caracteres de x com aqueles presentes em y, pela busca em nós de S(y) que sejam correspondentes às subpalavras de x. O principal benefício da árvore de sufixos é a utilização dos *suffix links*, que permitem uma mudança para outro nó interno, evitando com que uma subpalavra tenha que ser encontrada novamente. Uma intuição que permeia é a que quanto mais profunda uma busca desce na árvore, maior é a sua similaridade entre as palavras (CHANG; LAWLER, 1994).

MATERIAIS E MÉTODOS

MATERIAIS

A realização dos experimentos foi conduzida utilizando o dataset disponibilizado pela ferramenta CodAn (NACHTIGALL; KASHIWABARA; DURHAM, 2021), de onde foram extraídos os arquivos fasta. O kernel foi implementado em ANSI C para melhor desempenho. Uma árvore de sufixos de Ukkonen funcional foi escolhida de uma implementação existente implementada por Matt Porrit, disponível sob a licença GNU v3.0 em (https://github.com/mattporritt/suffix_tree). O *matching statistics* foi implementado seguindo o pseudocódigo e a lógica demonstrada anteriormente. O kernel foi implementado juntamente com os diferentes pesos, seguindo estritamente suas equações, e sua complexidade temporal foi empiricamente comprovada como O(n).

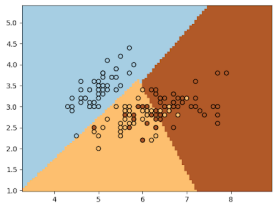
A análise dos kernels realizou-se em scripts em Python 3.11 juntamente com as bibliotecas: Ctypes, que permite a comunicação entre Python e C; Scikit Learn para realizar análises com o kernel desenvolvido por meio de uma SVM (Support Vector Machine) e métricas de aprendizado de máquina; NP Arrays para manipulação e criação de Arrays; Pandas para criação e manipulação de dataframes; Biopython para leitura dos arquivos fasta.

MÉTODOS

Para esse experimento, foi utilizado apenas o genoma de *H. sapiens* e foram selecionadas aleatoriamente 500 ncRNA's e 500 mRNA. Esta restrição no tamanho do dataset é devido à complexidade quadrática do cálculo da matriz de Kernel. Essa complexidade temporal representa uma importante limitação para esse método. Embora existam esforços para o cálculo de forma sub-quadrática para alguns tipos de kernel (BAKSHI et al., 2020), para *string kernels* ainda não há uma forma eficiente. O grupo de teste foi selecionado como um subconjunto contendo 30% dos objetos analisados. Os testes, predições e experimentos de validação foram conduzidos para duas configurações de pesos que dependem do tamanho da *string*, sendo o peso constante e o de peso intervalo limitado.

Com o kernel, foi possível aplicar a Máquina de Vetores de Suporte (SVM), um renomado algoritmo de classificação. Que conduz a classificação na diferenciação entre objetos utilizando interações entre *Feature Spaces* (SCHÖLKOPF; TSUDA; VERT, 2004), como ilustrado na Figura 2. Onde destacam-se a presença de falsos positivos e falsos negativos entre esses planos, evidenciados por pontos de cores diferentes em áreas que não correspondem às cores esperadas.

Figura 2 – Representação da classificação por SVM utilizando o Kernel Linear



Fonte: (ScikitLearn, 2023). Disponível em: https://scikitlearn.org/stable/auto_examples/svm/plot_custom_kernel.

Para validar o kernel implementado em um cenário de predição, usou-se métricas como acurácia, precisão, recall e F1 score.

RESULTADOS E DISCUSSÕES

Foram realizados cinco testes, tanto para o peso constante quanto para o peso de intervalo limitado, levando em consideração os mesmos objetos analisados. As Tabelas 1 e 2 refletem as predições realizadas com o peso Constante e o peso de intervalo limitado.

Tabela 1 – Resultados da predição utilizando o Fast String Kernel com o Peso Constante.

Métricas	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Médias	Desvio Padrão %
Acurácia	0,88	0,8	0,9	0,86	0,94	0,876	0,052
Precision Score	0,97	0,89	0,97	0,82	0,97	0,924	0,068
Recall Score	0,78	0,65	0,83	0,92	0,9	0,816	0,108
F1 Score	0,86	0,75	0,9	0,86	0,93	0,86	0,068

Fonte: Desenvolvido pelos autores (2023)

Tabela 2 – Resultados da predição utilizando o Fast String Kernel com o Peso de intervalo limitado.

Métricas	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Médias	Desvio Padrão %
Acurácia	0,88	0,8	0,9	0,86	0,94	0,876	0,052
Precision Score	0,97	0,89	0,97	0,82	0,97	0,924	0,068
Recall Score	0,78	0,65	0,83	0,92	0,9	0,816	0,108
F1 Score	0,86	0,75	0,9	0,86	0,93	0,86	0,068

Fonte: Desenvolvido pelos autores (2023)

A análise dos resultados mostra que o kernels com peso constante e peso de intervalo limitado têm resultados semelhantes, com alta acurácia média de 87,6% (±0,052) e precisão média elevada de 92,4% (±0,068), embora com um pouco mais de dispersão na precisão. O recall médio é mais baixo, com 81,6% (±0,108), mostrando maior variabilidade e sensibilidade a falsos negativos. O F1 Score fica próximo da fronteira entre precisão e recall, com média de 86%, indicando um equilíbrio geral na capacidade do modelo de classificar verdadeiros positivos e minimizar falsos positivos.

O *FSK* possui uma matriz K assimétrica, devido à sua abordagem de comparação entre elementos de x com S(y). Uma solução para essa assimetria é proposta, permitindo predições em tempo linear (VISHWANATHAN; SMOLA, 2002). No entanto essa abordagem não foi implementada. Valores extremos nas métricas podem ter sido afetados por essa característica do kernel, introduzindo instabilidade nas predições.

CONCLUSÃO

Os resultados obtidos com o kernel implementado mostraram boas métricas de avaliação com um certo nível de confiabilidade em suas análises, apesar de sua abordagem única. Sua versatilidade, potencializada pelos pesos, o torna interessante para a realização de análises. Além

disso, é importante destacar que esse kernel pode ser aplicado a qualquer sequência biológica e ainda pode ser aplicado a uma ampla variedade de dados no formato de *strings*. Portanto, sua utilidade e eficácia transcende o campo da bioinformática, oferecendo uma abordagem valiosa para diversas áreas de pesquisa e análise de dados.

AGRADECIMENTOS

Agradeço ao Prof. Dr. André Yoshiaki Kashiwabara pela confiança, paciência e entusiasmo ao transmitir o conhecimento e agradeço a Universidade Tecnológica Federal do Paraná pelo financiamento.

DISPONIBILIDADE DO CÓDIGO

O código está disponível para uso e pode ser encontrada no (Repositório-Fast-String-Kernel)

CONFLITO DE INTERESSES

Não há conflito de interesses.

REFERÊNCIAS

BAKSHI, Ainesh et al. Sub-quadratic Algorithms for Kernel Matrices via Kernel Density Estimation. arXiv, 2020.

CHANG, W. I.; LAWLER, E. L. Sublinear approximate string matching and biological applications. **Algorithmica**, v. 12, n. 4, p. 327–344, 1994.

CHOWDHURY, Sankalan Pal et al. Learning the Transformer Kernel. **Transactions on Machine Learning Research**, 2022.

HOFMANN, Thomas; SCHÖLKOPF, Bernhard; SMOLA, Alexander J. Kernel methods in machine learning. **The Annals of Statistics**, v. 36, n. 3, p. 1171–1220, 2008.

NACHTIGALL, Pedro G; KASHIWABARA, Andre Y; DURHAM, Alan M. CodAn: predictive models for precise identification of coding regions in eukaryotic transcripts. **Briefings in Bioinformatics**, v. 22, n. 3, 2021.

SCHÖLKOPF, Bernhard; TSUDA, Koji; VERT, Jean-Philippe. **Kernel Methods in Computational Biology**. [S.l.]: The MIT Press, 2004.

UKKONEN, E. On-line construction of suffix trees. **Algorithmica**, v. 14, n. 3, p. 249–260, 1995.

VISHWANATHAN, S. V. N; SMOLA, Alexander J. Fast kernels for string and tree matching. **Proceedings of the 15th International Conference on Neural Information Processing Systems**, p. 585–592, 2002.