

Problema do Máximo Subarray

Uma Análise de sua Complexidade

Matheus Garay Trindade¹, Guilherme de Freitas Gaiardo¹

¹Departamento de Eletrônica e Computação – Universidade Federal de Santa Maria
(UFSM)

97.105-900 – Santa Maria – RS – Brazil

{mtrindade, ggaiardo}@inf.ufsm.br

Abstract. *This paper analyses the classic computing problem of the maximum subarray. Three solutions with different complexities will be presented. This complexities will then be proved. For better understanding of what these complexities actually mean, various simulations were conducted with inputs of different sizes. With so, it is possible to perceive how the algorithm complexity impacts on its performance.*

Resumo. *Este artigo faz uma análise do clássico problema do subarray máximo. Serão apresentadas três soluções com diferentes complexidades. Essas complexidades serão então demonstradas. Para melhor entendimento do que essas complexidades significam, diversas simulações foram feitas com entradas de diferentes tamanhos. Com isso, é possível perceber o quanto a forma de crescimento do algoritmo em função da entrada têm um impacto direto na performance.*

1. Definição matemática

Para esta técnica, devemos escolher o hiperplano que melhor separa duas classes de dados em um espaço n -dimensional. Para ilustrar o ponto de partida, temos a Figura , onde os pontos positivos estão acima do hiperplano e os pontos negativos abaixo.

IMAGEM!!!!

Para encontrar o hiperplano de fato, é mais conveniente começar por definir o classificador. Para isso, define-se uma função que, dependendo do seu resultado, mostrará se o dado é de uma classe ou de outra, e.g., positivo ou negativo. Assumindo que se conhece hiperplano, pega-se um vetor \vec{w} perpendicular à ele. Temos que a projeção P de um dado \vec{d} sobre esse vetor tem valor dado pela seguinte função:

$$P(\vec{d}) = \vec{d} \cdot \vec{w} = \|\vec{d}\| \cdot \|\vec{w}\| \cdot \cos(\theta) = C \cdot \|\vec{d}\| \cdot \cos(\theta) \quad (1)$$

Onde θ é o ângulo entre \vec{d} e \vec{w} . Como a Equação 1 mostra, temos em $\|\vec{d}\| \cdot \cos(\theta)$ o comprimeto projeção de um dado sobre o vetor \vec{w} . Caso essa projeção seja maior que uma constante c , fica evidente que o dado está acima do hiperplano, sendo positivo, se for menor, está abaixo, sendo negativo, e se for igual a essa constante, é ponto do hiperplano. Formalmente, temos:

$$f(n) = \begin{cases} \vec{w} \cdot \vec{d} > c, & \text{se } \vec{d} \text{ é positivo} \\ \vec{w} \cdot \vec{d} < c & \text{se } \vec{d} \text{ é negativo} \\ \vec{w} \cdot \vec{d} = c & \text{se } \vec{d} \text{ está no hiperplano} \end{cases} \quad (2)$$

Ou ainda:

$$f(n) = \begin{cases} \vec{w} \cdot \vec{d} + b > 0, & \text{se } \vec{d} \text{ é positivo} \\ \vec{w} \cdot \vec{d} + b < 0 & \text{se } \vec{d} \text{ é negativo} \\ \vec{w} \cdot \vec{d} + b = 0 & \text{se } \vec{d} \text{ está no hiperplano} \end{cases} \quad (3)$$

A seguir, deve-se definir algumas condições. O classificador terá resultado 1 para vetores de suporte positivo e -1 para vetores de suporte negativo:

$$f(n) = \begin{cases} \vec{w} \cdot \vec{d} + b = 1, & \text{se } \vec{d} \text{ é positivo e é vetore de suporte} \\ \vec{w} \cdot \vec{d} + b = -1 & \text{se } \vec{d} \text{ é negativo e é vetore de suporte} \end{cases} \quad (4)$$

Assim, podemos então maximizar essa função de forma a encontrar o hiperplano que maximiza a distância entre as classes. Para isso, como se tem um função com restrições, para maximizá-la, usa-se multiplicadores de Lagrange. Assim, encontra-se o classificador ótimo. Vale que ressaltar que usa-se uma solução numérica para encontrá-lo, mas como o espaço do problema é convexo, ela sempre converge para o classificador ótimo.

2. SVM Multiclasses

SVM é intrinsicamente uma técnica para classificação binária, entretanto pode facilmente ser convertida para classificar um dado entre mais de duas classes. A seguir, serão apresentadas técnicas usadas para obter essa funcionalidade.

2.1. One-vs-All

Essa técnica consiste em criar uma SVM para cada classe separando-a das outras. Ou seja, se o problema for constituído de 3 classes, e.g, Azul, Vermelho e Amarelo, 3 SVMs serão construídas: uma que separa Azul de Vermelho e Amarelo, de Vermelho separado de Azul e Amarelo e outra que separa Amarelo de Azul e Vermelho. Após isso, entramos com um dado nessas 3 SVMs e avaliamos o resultado.