Teoria para o cálculo da curva

Fernando Pujaico Rivera

November 23, 2018

Abstract

1 Calculando o peço dos cúmulos

Dada uma imagem em branco e preto ($\mathbf{M}$), como mostrado na Figura 1, com

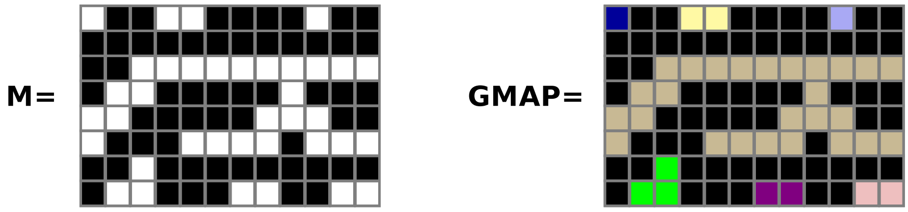

Figure 1: Imagem em branco e preto ($\mathbf{M}$), Imagem com cúmulos formados por grupos de pixels brancos ($\mathbf{GMAP}$)

valores binários onde os pixels com cor branca tem valor 1 e com cor preta 0. Se desejamos obter uma imagem com cúmulos formados pelos grupos de pixels brancos ($\mathbf{GMAP}$); onde cada pixel é mapeado com um índice ID (na figura este ID está representado com uma cor); devemos seguir o procedimento das seguintes seções.

1.1 Modelando o problema

Nosso problema principal é obter, a partir de uma matriz $\mathbf{M}$, uma matriz $\mathbf{GMAP}$ onde cade elemento contem o índice do cúmulo ao qual pertence, como mostra a Figura 1. Assim, pra cumprir este objetivo são usadas as funções $funA$ e $funB$, como mostra a Figura 2, onde são analisadas de maneira sequencial as colunas $\mathbf{M}(:,i)$ da matriz $\mathbf{M}$, desde $i=0$ ate $i=N_{col}-1$,

A função $funA$ recebe como parâmetro de entrada a coluna $\mathbf{M}(:,0)$ (primeira coluna da matriz $\mathbf{M}$) e retorna o valor dos índices em $\mathbf{GMAP}(:,0)$, adicionalmente a função retorna o valor do maior índice atribuído nas operações (MAX_ID).

Por outro lado a função $funB$ recebe como parâmetros de entrada a i-essima coluna ($\mathbf{M}(:,i)$) de $\mathbf{M}$, o mapa de índices $\mathbf{GMAP}(:,0:(i-1))$ que abrange informações de índice desde a primeira coluna ate a $(i-1)$-essima o valor do maior índice conhecido ate o momento, a função retorna o a matriz

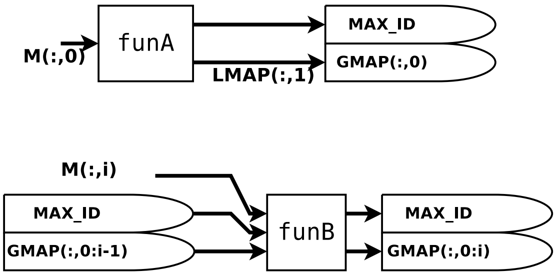

Figure 2: Diagrama de fluxo del algoritmo de criação de cúmulos

GMAP$(:, 0 : i)$ com os dados dos índices desde a primeira coluna ate a i-essima, adicionalmente a função retorna o valor do maior índice atribuído nas operações (MAX_ID).

1.1.1 Algoritmo da função $funA$

A função $funA$ recebe um vetor coluna **V** com uns e zeros, como é exemplificado na Figura 3, na saída a função retorna um vetor **LMAP** com o mesmo tamanho porem com elementos com valores inteiros, representado estes o mapeamento dos índices dos elementos em **V**, os elementos em **V** com valor zero recebem

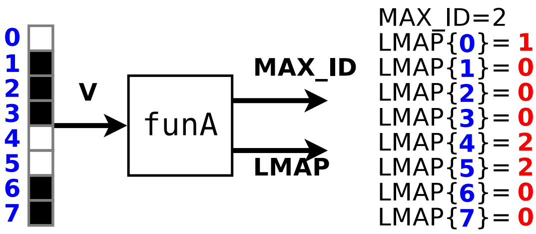

Figure 3: Descrição da função $funA$

automaticamente um índice de valor 0, já se uma região de uns em **V**, estão isolados, estes recebem em **LMAP** um índice com um valor maior em uma unidade do ultimo índice achado; a função $funA$ também retorna o máximo índice estabelecido (MAX_ID).

1.1.2 Algoritmo da função $funB$

A função $funB$ recebe como parâmetros de entrada, um vetor $\mathbf{M}(:, i)$ correspondente a i-essima coluna da matriz **M**, uma matriz **GMAP**$(:, 0 : (i-1))$ com o mapa de grupos desde a primeira coluna ate a $(i-1)$-essima coluna da matriz **GMAP**, e o máximo índice MAX_ID nos elementos da matriz **GMAP**$(:, 0 : (i-1))$, como é exemplificado na Figura 4. Na saída a função retorna uma matriz **GMAP**$(:, 0 : i)$ com o mapa de grupos desde a primeira coluna ate a i-essima coluna da matriz **GMAP**, e o novo máximo índice MAX_ID nos elementos da matriz **GMAP**$(:, 0 : i)$. A função $funB$ analisa os elementos do vetor $\mathbf{M}(:, i)$, desde a primeira linha ate a linha M_NLIN, mediante as

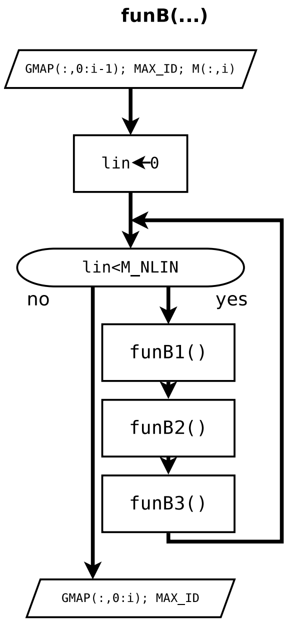

Figure 4: Descrição da função $funB$

funções auxiliares $funB1$, $funB2$ e $funB3$, estas funções modificam o conteúdo da matriz $\mathbf{GMAP}(:, 0:i)$ e a variável MAX_ID.

Algoritmo da função $funB1$: A função recebe como para metros de entrada os vetores $\mathbf{M}(:, i)$, $\mathbf{GMAP}(:, i)$ e uma variável com a linha lin onde inciara o analises. Se existem zeros em $\mathbf{M}(:, i)$, então se escrevem zeros nos elementos na mesma posição do vetor $\mathbf{GMAP}(:, i)$. Figura 5

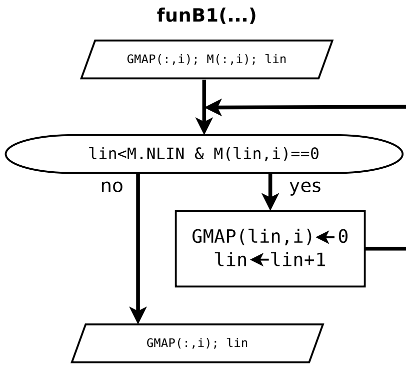

Figure 5: Descrição da função $funB1$

Algoritmo da função $funB2$: Verifico si existem uns; e acho o conjunto de ID globais ligados a esse grupo de uns, adicionalmente é contada a quantidade de uns. Figura 6

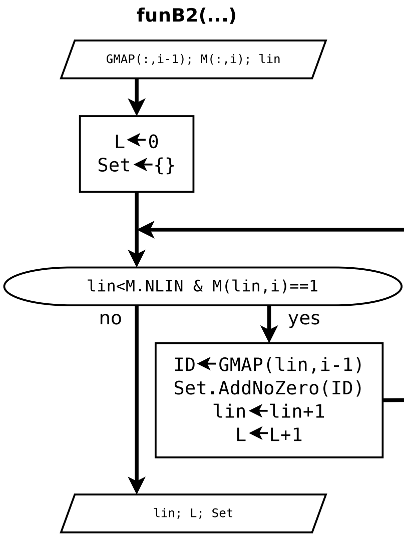

Figure 6: Descrição da função $funB$

1.2 Calculando a matriz GMAP e W

2 least-squares fitting of cubic splines (Method 1)

Este trabalho mostra como calcular o encaixe de uma curva $y = f(x)$ mediante mínimos quadrados sobre um conjunto de N pontos (x_n, y_n); $\forall n \in \mathtt{Z},\ 0 \leq n < N$; tendo cada ponto uma importância de w_n. Onde a curva $y = f(x)$ esta composta de um grupo de M splines cúbicos, como o exemplo da Figura 7.

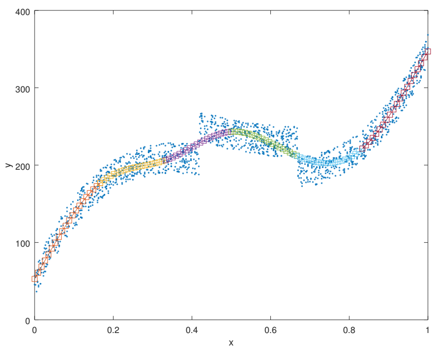

Figure 7: Fitting of $M = 6$ cubic splines in a set of $N = 500$ pontos (x_n, y_n).

Os splines tem seus limites de domínio, nas posições onde x tem valores $\varepsilon_0, \varepsilon_1, \varepsilon_2, ..., \varepsilon_M$.

Assim, usando as variáveis $\mathbf{E} = \begin{pmatrix}\varepsilon_0 & \varepsilon_1 & ... & \varepsilon_M\end{pmatrix}^T$ e $\mathbf{P} = \begin{pmatrix}P_0 & P_1 & ... & P_{4M-1}\end{pmatrix}^T$, podemos definir as seguintes funções,

$$G(x,a,b) = \begin{cases} 1 & if \quad a \leq x \leq b \\ 0 & if \quad other\ case \end{cases} \tag{1}$$

$$S_k(x,\mathbf{P}) = P_{4k}\ x^3 + P_{1+4k}\ x^2 + P_{2+4k}\ x + P_{3+4k}, \tag{2}$$

de modo que

$$f(x,\mathbf{P},\mathbf{E}) = \sum_{k=0}^{M-1} S_k(x,\mathbf{P}) G(x, \varepsilon_k, \varepsilon_{k+1}) \tag{3}$$

2.1 Modelando o problema

Para poder calcular o vetor $\mathbf{P} = \begin{pmatrix} P_0 & P_1 & P_2 & ... & P_{4M-1} \end{pmatrix}^T$, do problema spline descrito na Seção 2, a partir dos dados (x_n, y_n, w_n); $\forall n \in \mathtt{Z},\ 0 \leq n < N$ e escolhidos os valores $\mathbf{E} = \begin{pmatrix} \varepsilon_0 & \varepsilon_1 & ... & \varepsilon_M \end{pmatrix}^T$; ordenamos nossos dados nos seguintes vetores coluna:

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_{M-1} \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} \mathbf{Y}_0 \\ \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_{M-1} \end{pmatrix}, \quad \mathbf{W} = \begin{pmatrix} \mathbf{W}_0 \\ \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_{M-1} \end{pmatrix}, \tag{4}$$

onde os elementos de $\mathbf{X}_m$,$\mathbf{Y}_m$,$\mathbf{W}_m$ representam os valores de (x_n, y_n, w_n) que cumprem $\varepsilon_m \leq x_n \leq \varepsilon_{m+1}$.

2.1.1 Equação que relacionam os dados

Os dados (x_n, y_n) são agrupados em $\mathbf{Y} = \mathbf{AP}$, onde,

$$\mathbf{A}_m = \begin{pmatrix} \mathbf{X}_m^3 & \mathbf{X}_m^2 & \mathbf{X}_m & \mathbf{1} \end{pmatrix} \tag{5}$$

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_2 & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_{M-1} \end{pmatrix} \tag{6}$$

2.1.2 Equação de continuidade dos splines consecutivos

Dados os polinômios $S_{k-1}(x,\mathbf{P})$ e $S_k(x,\mathbf{P})$ do $(k-1)$-essimo e k-essimo spline, respetivamente. Podemos igualar ambos polinômios para garantir a continuidade dos splines em $x = \varepsilon_k$, de modo que,

$$S_{k-1}(\varepsilon_k, \mathbf{P}) = S_k(\varepsilon_k, \mathbf{P}) \tag{7}$$

Assim, agrupando as equações dos polinômios de $0 < k < M$, obtemos $\mathbf{0} = \mathbf{B}_0\mathbf{P}$, onde,

$$\mathbf{B}_0 = \begin{pmatrix} \mathbf{E}_1 & -\mathbf{E}_1 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_2 & -\mathbf{E}_2 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{E}_3 & -\mathbf{E}_3 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{E}_{M-1} & -\mathbf{E}_{M-1} \end{pmatrix}, \tag{8}$$

usando como variável auxiliar,

$$\mathbf{E}_k = \begin{pmatrix} \varepsilon_k^3 & \varepsilon_k^2 & \varepsilon_k & 1 \end{pmatrix}. \tag{9}$$

2.1.3 Equação de continuidade da 1ra derivada dos splines consecutivos

Dados os polinômios $S_{k-1}(x, \mathbf{P})$ e $S_k(x, \mathbf{P})$ do $(k-1)$-essimo e k-essimo spline, respetivamente. Podemos igualar a primeira derivada de ambos polinômios para garantir a continuidade dos splines em $x = \varepsilon_k$, de modo que,

$$\left.\frac{\partial\ S_{k-1}(x, \mathbf{P})}{\partial x}\right|_{x=\varepsilon_k} = \left.\frac{\partial\ S_k(x, \mathbf{P})}{\partial x}\right|_{x=\varepsilon_k} \tag{10}$$

Assim, agrupando as equações dos polinômios de $0 < k < M$, obtemos $\mathbf{0} = \mathbf{B}_1\mathbf{P}$, onde,

$$\mathbf{B}_1 = \begin{pmatrix} \partial\mathbf{E}_1 & -\partial\mathbf{E}_1 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \partial\mathbf{E}_2 & -\partial\mathbf{E}_2 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \partial\mathbf{E}_3 & -\partial\mathbf{E}_3 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \partial\mathbf{E}_{M-1} & -\partial\mathbf{E}_{M-1} \end{pmatrix}, \tag{11}$$

usando como variável auxiliar,

$$\partial\mathbf{E}_k = \begin{pmatrix} 3\varepsilon_k^2 & 2\varepsilon_k & 1 & 0 \end{pmatrix}. \tag{12}$$

2.1.4 Equação de continuidade da 2da derivada dos splines consecutivos

Dados os polinômios $S_{k-1}(x, \mathbf{P})$ e $S_k(x, \mathbf{P})$ do $(k-1)$-essimo e k-essimo spline, respetivamente. Podemos igualar a segunda derivada de ambos polinômios para garantir a continuidade dos splines em $x = \varepsilon_k$, de modo que,

$$\left.\frac{\partial^2\ S_{k-1}(x, \mathbf{P})}{\partial x^2}\right|_{x=\varepsilon_k} = \left.\frac{\partial^2\ S_k(x, \mathbf{P})}{\partial x^2}\right|_{x=\varepsilon_k} \tag{13}$$

Assim, agrupando as equações dos polinômios de $0 < k < M$, obtemos $\mathbf{0} = \mathbf{B}_2\mathbf{P}$, onde,

$$\mathbf{B}_2 = \begin{pmatrix} \partial^2\mathbf{E}_1 & -\partial^2\mathbf{E}_1 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \partial^2\mathbf{E}_2 & -\partial^2\mathbf{E}_2 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \partial^2\mathbf{E}_3 & -\partial^2\mathbf{E}_3 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \partial^2\mathbf{E}_{M-1} & -\partial^2\mathbf{E}_{M-1} \end{pmatrix}, \tag{14}$$

usando como variável auxiliar,

$$\partial^2\mathbf{E}_k = \begin{pmatrix} 6\varepsilon_k & 2 & 0 & 0 \end{pmatrix}. \tag{15}$$

2.2 Calculando os parâmetros P_j

Agrupando os resultados da Seção 2.1, podemos definir a seguinte equação,

$$\begin{pmatrix} \mathbf{Y} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{A} \\ \mathbf{B}_0 \\ \mathbf{B}_1 \\ \mathbf{B}_2 \end{pmatrix} \mathbf{P} \tag{16}$$

ou seu equivalente, $\mathbf{Z} = \mathbf{QP}$. Onde a única incógnita é $\mathbf{P}$.

Para resolver este problema aplicamos LMS (Least Mean Square) a equação, para isso definimos a regra de minimização $e(\mathbf{P})$,

$$e(\mathbf{P}) = ||\mathbf{Z} - \mathbf{QP}||^2_{diag(\mathbf{W})} + \alpha||\mathbf{P} - \mathbf{P}_*||. \tag{17}$$

Onde $\mathbf{P}_*$ é o valor de $\mathbf{P}$ na iteração anterior. Assim, aplicando LMS e a regularização de Tikhonov, o mínimo valor de $\mathbf{P}$ se obtêm iterativamente usando a seguinte equação

$$\mathbf{P}_{k+1} = \mathbf{P}_k + \left[\mathbf{Q}^T diag(\mathbf{W})\mathbf{Q} + \alpha\mathbf{I}\right]^{-1} \mathbf{Q}^T diag(\mathbf{W}) \left[\mathbf{Z} - \mathbf{QP}_k\right] \tag{18}$$

ate que os vetores $\mathbf{P}_k$ e $\mathbf{P}_{k+1}$ sejam muito próximos.