

5

Representação de variedades implícitas e Partição da Unidade

Um dos objetivos deste trabalho é apresentar um método construtivo para aproximar uma variedade implícita e, utilizar essa aproximação para realizar uma regressão. Muitos estudos têm sido feitos para reconhecimento de formas e objetos, os quais utilizam métodos implícitos para reconstrução de curvas e superfícies a partir de dados esparsos. Tasdizen et al. em (70) apresenta um método de ajuste linear baseado em *ridge regression* para representar formas e utiliza curvas algébricas. Na dissertação de Arouca (2), propõe-se um método implícito para reconstrução de curvas, que emprega a partição da unidade para obter uma aproximação global a partir das aproximações locais. A solução de decompor o domínio e aproximar os dados em cada subdomínio separadamente e, então, juntar essas soluções locais para obter uma descrição global do objeto, foi proposta no trabalho de Ohtake et al.(49), que apresentou um método implícito para reconstrução de superfícies em \mathbb{R}^3 , chamado *Multi-level Partition of Unity Implicits* ou MPU. Uma nova proposta para a reconstrução de superfícies no \mathbb{R}^3 é estudada em Mederos et al. (42), a qual se baseia no método MPU, mas utiliza *ridge regression* e técnicas de *weighted gradient one fitting*. Já Macedo, em sua dissertação (1), apresenta um método para reconstrução de superfícies baseado em Funções de Base Radial Hermite Implícitas (*HRBF Implicits*). Em termos de visualização, o artigo de Vital Brazil et al. (9), apresenta técnicas de renderização de superfícies implícitas em diferentes estilos de caneta e tinta (*pen-and-ink*) e a dissertação de Azevedo (3) apresenta ainda um método para visualização por pontos de superfícies implícitas no \mathbb{R}^4 com técnicas parecidas.

Neste trabalho, estamos interessados no estudo de variedades M de dimensão m , as quais são subconjuntos de espaços euclidianos \mathbb{R}^n . Consideraremos $m = 1, 2, \dots, n - 1$. O caso $m = 0$ implica apenas na presença de pontos isolados em \mathbb{R}^n , assim, nenhum cálculo pode ser feito, já o caso $m = n$ significa que se tem um conjunto aberto M em \mathbb{R}^n e, dessa maneira, M e \mathbb{R}^n são localmente os mesmos. Temos que:

- $m = 1$ implica em variedades de dimensão 1, “curvas”;

- $m = 2$ implica em variedades de dimensão 2, “superfícies”;
- $m = n - 1$ implica em variedades de dimensão $n - 1$, chamadas de “hipervariedades”

Inicialmente, trabalharemos com hipervariedades. Um exemplo de hipervariabilidade é a esfera unitária em \mathbb{R}^n :

$$S(\mathbf{0}, 1) = \{\mathbf{x} | \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\| = 1\}$$

Considerando $m = n - 1$, vamos obter a descrição implícita dessa variedade como em Jones (28).

Teorema 5.1 (Teorema da Função Implícita para Hipervariedades). *Suponha que a hipervariabilidade $M \subset \mathbb{R}^n$ é descrita como o conjunto de nível $F(\mathbf{x}) = 0$, onde $F : \mathbb{R}^n \rightarrow \mathbb{R}$ é de classe \mathcal{C}^1 e $\nabla F(\mathbf{x}) \neq 0, \forall \mathbf{x} \in M$. Seja $\mathbf{x}_0 \in M$. Suponha que $\frac{\partial F}{\partial x_i}(\mathbf{x}_0) \neq 0$. Então existe $g : \mathbb{R}^n \rightarrow \mathbb{R}$ de classe \mathcal{C}^1 tal que para todo \mathbf{x} em uma vizinhança suficientemente pequena de \mathbf{x}_0 ,*

$$F(\mathbf{x}) = 0 \leftrightarrow x_i = g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n).$$

Teorema 5.2. *Dada uma hipervariabilidade $M \subset \mathbb{R}^n$, descrita como o conjunto de nível $\{\mathbf{x} \in \mathbb{R}^n | F(\mathbf{x}) = 0\}$, onde $\nabla F(\mathbf{x}) \neq 0, \forall \mathbf{x} \in M$. Então $\nabla F(\mathbf{x})$ é unicamente determinado por M , a menos de um múltiplo escalar não nulo (o qual pode ser uma função de \mathbf{x} .)*

Vamos definir vetores tangentes a variedade. Seja $M \subset \mathbb{R}^n$ uma variedade e considere $\mathbf{x}_0 \in M$. Vamos definir os vetores tangentes a M em \mathbf{x}_0 . Para isso, considere curvas que estão na variedade.

Definição 5.1. *Considere todas as curvas $\gamma : \mathbb{R} \rightarrow \mathbb{R}^n$ que satisfazem: $\gamma(t) \in M, \forall t; \gamma(0) = \mathbf{x}_0, \gamma$ é de classe \mathcal{C}^1 . Então o vetor velocidade $\gamma'(0)$ é chamado vetor tangente a M em \mathbf{x}_0 . Note que $\gamma'(0) \in \mathbb{R}^n$. O conjunto formado por todos esses vetores é chamado o espaço tangente a M em \mathbf{x}_0 e é escrito por $T_{\mathbf{x}_0}M$.*

Teorema 5.3. *Seja M uma hipervariabilidade em \mathbb{R}^n , descrita implicitamente como um conjunto de nível $F(\mathbf{x}) = 0$, onde F é de classe \mathcal{C}^1 e $\nabla F(\mathbf{x}) \neq 0, \forall \mathbf{x} \in M$. Então:*

$$T_{\mathbf{x}_0}M = \{\mathbf{h} \in \mathbb{R}^n | \nabla F(\mathbf{x}_0)\mathbf{h} = 0\}$$

Em particular, $T_{\mathbf{x}_0}M$ é um subespaço de dimensão $n - 1$ em \mathbb{R}^n .

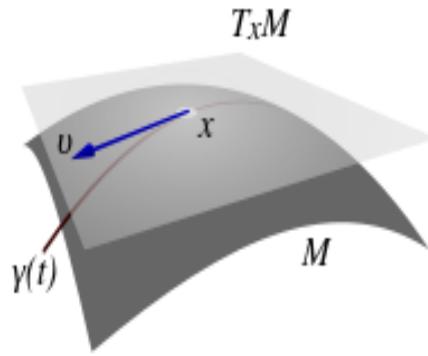


Figura 5.1: $T_x M$

Uma variedade $M \subset \mathbb{R}^n$ de dimensão m , $1 \leq m \leq n-1$ será (localmente) representada como a interseção de $(n-m)$ hipervariiedades dada pelo conjunto de nível zero de uma função: $F : \mathbb{R}^n \rightarrow \mathbb{R}^{n-m}$, onde o inteiro $(n-m)$ é frequentemente chamado a codimensão de M . Portanto, queremos descrever M localmente como o conjunto de todos os pontos $\mathbf{x} \in \mathbb{R}^n$ satisfazendo as equações:

$$\begin{pmatrix} F_1(\mathbf{x}) = 0 \\ F_2(\mathbf{x}) = 0 \\ \vdots \\ F_{n-m}(\mathbf{x}) = 0 \end{pmatrix}$$

onde $F : \mathbb{R}^n \rightarrow \mathbb{R}^{n-m}$.

Assumimos que cada função F_i é pelo menos de classe C^1 e que os gradientes $\nabla F_i(\mathbf{x})$ são linearmente independentes para qualquer $\mathbf{x} \in M$. Para exemplos da necessidade de cada uma das restrições, veja (28), que também apresenta a representação explícita e paramétrica de M e a equivalência entre essas representações.

Dadas as hipóteses a serem consideradas ao se definir implicitamente uma variedade, vejamos a definição seguinte.

Definição 5.2. *Seja $M \subset \mathbb{R}^n$ uma variedade de dimensão m , $1 \leq m \leq n-1$. Suponha que em uma vizinhança de qualquer ponto $\mathbf{x}_0 \in M$ existam funções F_1, \dots, F_{n-m} de classe C^1 tais que:*

- próximo a \mathbf{x}_0 , $\mathbf{x} \in M \leftrightarrow F_i(\mathbf{x}) = 0$ para $1 \leq i \leq n-m$.
- $\nabla F_1(\mathbf{x}_0), \nabla F_2(\mathbf{x}_0), \dots, \nabla F_{n-m}(\mathbf{x}_0)$ são linearmente independentes.

então M é uma variedade implícita.

A condição de independência linear pode ser considerada na forma da matriz jacobiana. Assim, DF deve ter posto máximo. Podemos escrever:

$$F(\mathbf{x}) = \begin{pmatrix} F_1(\mathbf{x}) \\ F_2(\mathbf{x}) \\ \vdots \\ F_{n-m}(\mathbf{x}) \end{pmatrix}$$

onde $F : \mathbb{R}^n \rightarrow \mathbb{R}^{n-m}$. Então:

$$DF(\mathbf{x}) = \begin{pmatrix} \partial F_1(\mathbf{x})/\partial x_1 & \dots & \partial F_1(\mathbf{x})/\partial x_n \\ \vdots & \ddots & \vdots \\ \partial F_{n-m}(\mathbf{x})/\partial x_1 & \dots & \partial F_{n-m}(\mathbf{x})/\partial x_n \end{pmatrix}$$

representa a matriz jacobiana de F . A matriz é $(n-m) \times n$ e deve satisfazer a condição de que em $\mathbf{x} = \mathbf{x}_0$ suas linhas devem ser linearmente independentes. O teorema a seguir é uma generalização do teorema 5.1, o qual foi descrito para hipervariiedades.

Teorema 5.4 (Teorema da Função Implícita). *Seja $1 \leq k \leq n-1$ e suponha que $F : \mathbb{R}^n \rightarrow \mathbb{R}^k$ é de classe \mathcal{C}^1 em uma vizinhança de \mathbf{x}_0 com $F(\mathbf{x}_0) = 0$. Suponha ainda que a matriz jacobiana $DF(\mathbf{x}_0)$ tenha posto máximo (igual a k). Então existem k colunas de $DF(\mathbf{x}_0)$ linearmente independentes. Para facilitar a notação, suponha que as últimas k colunas são linearmente independentes e representaremos $\mathbb{R}^n = \mathbb{R}^m \times \mathbb{R}^k$. Assim, pontos em \mathbb{R}^n serão representados como $\mathbf{x} = (x_1, \dots, x_m, x_{m+1}, \dots, x_n) = (\mathbf{x}'; \mathbf{x}'')$. Portanto, existem k funções unicamente determinadas:*

$$g_i : \mathbb{R}^m \rightarrow \mathbb{R}, \quad 1 \leq i \leq k,$$

de classe \mathcal{C}^1 próxima ao ponto \mathbf{x}'_0 , tal que para todo \mathbf{x} em alguma vizinhança de \mathbf{x}_0 ,

$$F(\mathbf{x}) = 0 \leftrightarrow x_{m+i} = g_i(\mathbf{x}'), \quad \text{para } 1 \leq i \leq k.$$

Ademais, se F é de classe \mathcal{C}^l , então g_i é de classe $\mathcal{C}^l, \forall 1 \leq i \leq k$.

A conclusão do teorema nos mostra que conseguimos resolver o sistema de equações $F(\mathbf{x}) = 0$ para k das coordenadas em termos das outras m coordenadas. Ademais, a variedade M tem sua representação implícita dada por:

$$\begin{cases} F_1(\mathbf{x}) = 0, \\ F_2(\mathbf{x}) = 0, \\ \vdots \\ F_k(\mathbf{x}) = 0. \end{cases}$$

Considere a notação do espaço tangente a variedade M em $\mathbf{x}_0 : T_{\mathbf{x}_0}M$, que implicitamente é dado por:

$$h \in T_{\mathbf{x}_0}M \leftrightarrow \nabla F_i(\mathbf{x}_0)h = 0, \forall 1 \leq i \leq k. \quad (5-1)$$

Como os vetores normais $\nabla F_i(\mathbf{x}_0)$ a cada hipervariabilidade i são linearmente independentes, tem-se que $T_{\mathbf{x}_0}M$ é um espaço vetorial de dimensão m .

5.1

Aproximação da variedade implícita

Dados $\mathbb{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \in \mathbb{R}^n$, N pontos de uma variedade M de dimensão m , a qual será denotada por M^m . Suponha que também conheçamos o espaço tangente à variedade em cada ponto \mathbf{x}_i , $1 \leq i \leq N$. Queremos obter uma aproximação para as funções F_1, \dots, F_{n-m} , as quais devem ser, pelo menos, de classe C^1 e satisfaçam as hipóteses da definição 5.2, isto é, devemos ter:

- $F_i(\mathbf{x}_j) = 0$, $1 \leq i \leq n - m$ e $1 \leq j \leq N$;
- $\nabla F_1(\mathbf{x}_j), \nabla F_2(\mathbf{x}_j), \dots, \nabla F_{n-m}(\mathbf{x}_j)$ são linearmente independentes para todo $\mathbf{x}_j \in M$, $1 \leq j \leq N$.

Para obter uma aproximação para essas funções utilizaremos polinômios multivariados (n variáveis) de grau d :

$$P^{(d)}(\mathbf{x}) = \sum_{\substack{j_1, j_2, \dots, j_n \geq 0 \\ 0 \leq j_1 + \dots + j_n \leq d}} a_{j_1, \dots, j_n} x_1^{j_1} x_2^{j_2} \dots x_n^{j_n}.$$

O teorema a seguir fornece o número de coeficientes necessários para descrever um polinômio de n variáveis e grau d .

Teorema 5.5. *Considere a notação $\Pi_d(\mathbb{R}^n)$, que denota o espaço de polinômios de n variáveis e grau absoluto no máximo d e \mathbb{N}_0^d , que tem como componentes multi-índices representados por $\alpha = (\alpha_1, \dots, \alpha_d)^T$.*

1. *Os monônimos $x \rightarrow x^\alpha$, $x \in \mathbb{R}^n$, $\alpha \in \mathbb{N}_0^d$, são linearmente independentes.*

$$2. \dim \Pi_d(\mathbb{R}^n) = \binom{n+d}{n} = \frac{(n+d)!}{n!d!}.$$

A demonstração pode ser obtida em Wendland (79).

Utilizaremos uma notação para o polinômio $P^{(d)}(\mathbf{x})$, que generaliza a adotada para curvas algébricas por Tasdizen et al (70) e Arouca (2).

$$P^{(d)}(\mathbf{x}) = \mathbf{v}^t \mathbf{a},$$

onde

- $\mathbf{a} \in \mathbb{R}^l$ é o vetor de coeficientes:

$$(\mathbf{a}_{j_1, j_2, \dots, j_n})_{0 \leq j_1, \dots, j_n; 0 \leq j_1 + \dots + j_n \leq d}, \quad (5-2)$$

- $\mathbf{v} \in \mathbb{R}^l$ é o vetor dos monômios:

$$\left(\prod_{i=1}^n \mathbf{x}_i^{j_i} \right)_{0 \leq j_1, \dots, j_n; 0 \leq j_1 + \dots + j_n \leq d}. \quad (5-3)$$

- A dimensão desses vetores é $l = \frac{(n+d)!}{n!d!}$.

A aproximação de cada uma das funções $F_i(\mathbf{x})$ será feita utilizando polinômios e deverá satisfazer a restrição $F_i(\mathbf{x}_j) = 0$, $1 \leq j \leq N$. Assim, uma maneira simples para alcançarmos tal objetivo será minimizar a distância algébrica sobre o conjunto de pontos dados $\{\mathbf{x}_1, \dots, \mathbf{x}_N\} \in \mathbb{R}^n$ com o critério de mínimos quadrados, isto é, determinar os coeficientes de $(P_i^{(d)})$ de maneira que a soma da distância algébrica de cada um dos N pontos à variedade seja a menor possível. Em outras palavras, buscaremos minimizar:

$$e_{total_i} = \sum_{j=1}^N (P_i^{(d)}(\mathbf{x}_j))^2, \quad \mathbf{x}_j \in \mathbb{R}^n \quad (5-4)$$

Utilizando a notação vetorial de $P_i^{(d)}$ em 5-4 temos:

$$e_{total_i} = \mathbf{a}^t \left(\sum_{j=1}^N \mathbf{v}_j \mathbf{v}_j^t \right) \mathbf{a} \quad (5-5)$$

onde \mathbf{a} e \mathbf{v} são como descritos em 5-2 e 5-3, respectivamente.

Para melhorar a notação, definimos:

- A matriz \mathbf{M} de tamanho $l \times N$:

$$\mathbf{M} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \dots \quad \mathbf{v}_N].$$

- A matriz *dispersão* dos monômios:

$$\mathbf{S} = \mathbf{M} \mathbf{M}^t = \sum_{j=1}^N \mathbf{v}_j \mathbf{v}_j^t \quad (5-6)$$

A fim de evitar a solução trivial para o problema de minimização proposto, uma restrição tal como $\|\mathbf{a}\|^2 = 1$ é imposta e o problema de minimização pode ser reescrito como:

$$\begin{aligned} & \min_{\mathbf{a}} \{\mathbf{a}^t \mathbf{S} \mathbf{a}\}, \\ & \text{sujeito a: } \|\mathbf{a}\|^2 = 1 \end{aligned}$$

A solução para o problema de minimização descrito é dada pelo autovetor unitário de \mathbf{a} associado ao menor autovalor de \mathbf{S} . Tasdizen et al. (70) aponta alguns problemas com o método de mínimos quadrados clássicos, o qual não se mostra útil para representação de formas. Para solucionar tal problema, uma outra aproximação baseada no gradiente do polinômio é apresentada, num algoritmo chamado *Gradiente One Fitting*. A proposta consiste em restringir o gradiente polinomial em cada ponto a ser perpendicular às tangentes locais, além de possuírem norma unitária. Tal modificação faz com que o conjunto de nível zero do polinômio respeite a continuidade local do conjunto de dados. Dessa maneira, dois novos termos são adicionados a 5-5. Como o problema estudado em (70) se referia a curvas no \mathbb{R}^2 , vejamos como fica o problema restrito a esse caso.

Dados:

- $\mathbb{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \in \mathbb{R}^2$; N pontos amostrados de uma curva planar \mathcal{C} ;
- $\mathcal{T} = \{\mathbf{t}_1, \dots, \mathbf{t}_N\} \in \mathbb{R}^2$; conjunto de vetores tangentes unitários à curva \mathcal{C} ;
- $\mathcal{N} = \{\mathbf{n}_1, \dots, \mathbf{n}_N\} \in \mathbb{R}^2$; conjunto de vetores normais unitários à curva \mathcal{C} ;
- $\nabla P^{(d)}(\mathbf{x}) = \begin{pmatrix} \frac{\partial P^{(d)}}{\partial x_1}(\mathbf{x}) \\ \frac{\partial P^{(d)}}{\partial x_2}(\mathbf{x}) \end{pmatrix}$; gradiente de $P^{(d)}$ em \mathbf{x} .

Portanto, a função a ser minimizada fica da seguinte forma:

$$e_{grad} = \sum_{j=1}^N \{(P^{(d)}(\mathbf{x}))^2 + \mu(\mathbf{n}_i^t \nabla P^{(d)}(\mathbf{x}) - 1)^2 + \mu(\mathbf{t}_i^t \nabla P^{(d)}(\mathbf{x}))^2\} \quad (5-7)$$

onde μ é o peso que se dá aos dois novos termos.

Definem-se algumas matrizes e vetores para facilitar a representação. Utilizando a representação vetorial de $P^{(d)}$ em 5-5 tem-se:

- O vetor gradiente $\nabla P^{(d)}$:

$$\nabla P^{(d)} = \nabla(\mathbf{v}_j^t \mathbf{a}) = (\mathbf{D}_j)^t \mathbf{a}. \quad (5-8)$$

onde a matriz \mathbf{D}_j de tamanho $l \times 2$ com $l = \frac{(2+d)!}{2!d!} = \frac{(d+1)(d+2)}{2}$ é dada por:

$$\mathbf{D}_j = \begin{pmatrix} \frac{\partial \mathbf{v}_j}{\partial x_1} & \frac{\partial \mathbf{v}_j}{\partial x_2} \end{pmatrix} \quad (5-9)$$

- A matriz \mathbf{S}_N de tamanho $l \times l$:

$$\mathbf{S}_N = \sum_{j=1}^N \mathbf{D}_j \mathbf{n}_j \mathbf{n}_j^t \mathbf{D}_j^t \quad (5-10)$$

- A matriz \mathbf{S}_T de tamanho $l \times l$:

$$\mathbf{S}_T = \sum_{j=1}^N \mathbf{D}_j \mathbf{t}_j \mathbf{t}_j^t \mathbf{D}_j^t \quad (5-11)$$

- O vetor \mathbf{g}_N de tamanho l :

$$\mathbf{g}_N = \sum_{j=1}^N \mathbf{D}_j \mathbf{n}_j \quad (5-12)$$

Assim, o problema de otimização 5-7 fica:

$$\min_{\mathbf{a}} \{e_{grad}\} = \min_{\mathbf{a}} \{\mathbf{a}^t \mathbf{S} \mathbf{a} + \mu \mathbf{a}^t \mathbf{S}_N \mathbf{a} + \mathbf{a}^t \mathbf{S}_T \mathbf{a} - 2\mu \mathbf{a}^t \mathbf{g}_N\} \quad (5-13)$$

A solução do problema é obtida resolvendo-se o seguinte sistema de equações lineares:

$$(\mathbf{S} + \mu(\mathbf{S}_N + \mathbf{S}_T))\mathbf{a} = \mu \mathbf{g}_N. \quad (5-14)$$

Descrevemos o problema de aproximar uma curva no \mathbb{R}^2 como resolvido em (70). Em (42), é feita uma modificação no método *gradient one fitting* para reconstrução de superfícies no \mathbb{R}^3 . Nessa dissertação, tratamos o problema de representar uma variedade implícita M^m , onde $1 \leq m \leq n - 1$, ou seja, generalizaremos os casos anteriores para lidar com outros tipos de estruturas além de curvas e superfícies e até mesmo com essas estruturas em outros espaços além do \mathbb{R}^2 e \mathbb{R}^3 , respectivamente.

Temos como conjunto de dados de entrada as seguintes informações:

- a dimensionalidade m da variedade M que queremos aproximar;
- conjunto de pontos $\mathbb{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \in M \subset \mathbb{R}^n$;
- o espaço tangente a variedade em cada ponto, conforme dado em 5-1, isto é, $\mathcal{T} = \{T_{\mathbf{x}_1}M, T_{\mathbf{x}_2}M, \dots, T_{\mathbf{x}_N}M\}$, onde $T_{\mathbf{x}_j}M = \{t_{j1}, t_{j2}, \dots, t_{jm}\}$ e cada $t_{jh}, 0 \leq j \leq N, 0 \leq h \leq m$ é um vetor tangente unitário.

Será feita uma modificação no método *Gradient One Fitting* de forma a ajustar às informações disponíveis e satisfazer as restrições em 5.2.

Cada uma das $(n - m)$ funções $F_i(\mathbf{x})$ será obtida pela aproximação de um polinômio em n variáveis e grau d cujos coeficientes serão obtidos de maneira minimizar a soma da distância algébrica de cada um dos N pontos à variedade. Dada a informação das tangentes, também será adicionado o termo que restringe o gradiente do polinômio em cada ponto, para que o mesmo fique perpendicular ao espaço vetorial tangente, em cada ponto. Para que o problema tenha solução não trivial também é adicionada a restrição de que a norma dos coeficientes é 1.

Defina:

$$\mathbf{S}_{\mathbf{T}} = \sum_{j=1}^N \left(\sum_{h=1}^m \mathbf{D}_j \mathbf{t}_{jh} \mathbf{t}_{jh}^t \mathbf{D}_j^t \right) \quad (5-15)$$

onde a matriz \mathbf{D}_j de tamanho $l \times n$, $\left(l = \frac{(n+d)!}{n!d!} \right)$, é dada por:

$$\mathbf{D}_j = \begin{pmatrix} \frac{\partial \mathbf{v}_j}{\partial x_1} & \frac{\partial \mathbf{v}_j}{\partial x_2} & \cdots & \frac{\partial \mathbf{v}_j}{\partial x_n} \end{pmatrix} \quad (5-16)$$

Assim, o problema de minimização fica descrito do seguinte modo:

$$\begin{aligned} \min_{\mathbf{a}} \{ \mathbf{a}^t \mathbf{S} \mathbf{a} + \mu \mathbf{a}^t \mathbf{S}_{\mathbf{T}} \mathbf{a} \}, \\ \text{sujeito a: } \|\mathbf{a}\|^2 = 1 \end{aligned} \quad (5-17)$$

Com o uso do multiplicador de Lagrange λ , o problema de otimização condicionado torna-se:

$$\min_{\mathbf{a}} \{ \mathbf{a}^t \mathbf{S} \mathbf{a} + \mu \mathbf{a}^t \mathbf{S}_{\mathbf{T}} \mathbf{a} + \lambda (\mathbf{a}^t \mathbf{a} - 1) \} \quad (5-18)$$

Chamamos $\mathbf{M} = \mathbf{S} + \mu \mathbf{S}_{\mathbf{T}}$. Assim, o problema fica reescrito como:

$$\min_{\mathbf{a}} \{ \mathbf{a}^t \mathbf{M} \mathbf{a} + \lambda (\mathbf{a}^t \mathbf{a} - 1) \}$$

A solução é dada pelo autovetor unitário de \mathbf{M} associado ao seu menor autovalor.

Para obtermos aproximações distintas das funções $F_i(\mathbf{x}_j)$, também precisamos incluir outra informação e, para isso, devemos considerar a hipótese de que $\nabla F_1(\mathbf{x}_j), \nabla F_2(\mathbf{x}_j), \dots, \nabla F_{n-m}(\mathbf{x}_j)$ são linearmente independentes para todo $\mathbf{x}_j \in M, 1 \leq j \leq N$. Para tentar obedecer a restrição de que os gradientes sejam linearmente independentes, vamos forçar o ajuste de maneira que os gradientes polinomiais sejam ortogonais entre si, isto é, devemos ter:

$$\nabla P_q^{(d)}(\mathbf{x}) \cdot \nabla P_r^{(d)}(\mathbf{x}) = 0, \quad \forall q \neq r; \quad q, r \in \{1, 2, \dots, n - m\}. \quad (5-19)$$

A implementação será feita por meio de um algoritmo sequencial. Descrevemos os passos:

1. Para obter uma aproximação para $F_1(\mathbf{x})$, resolvemos o problema 5-17 e obteremos os coeficientes do polinômio.
2. Para obter uma aproximação para $F_2(\mathbf{x})$, acrescentamos o termo: $\nabla P_1^{(d)}(\mathbf{x}) \cdot \nabla P_2^{(d)}(\mathbf{x})$, onde $P_1^{(d)}$ foi determinado no passo 1. Para descrever o problema, definamos:

- $\mathbf{n}_{1j} = \nabla P_1^{(d)}(\mathbf{x}_j)$ o vetor gradiente unitário do polinômio obtido no passo 1 calculado em \mathbf{x}_j .
- O vetor gradiente $\nabla P_2^{(d)}$ determinado em 5-8:

$$\nabla P_2^{(d)} = \nabla(\mathbf{v}_j^t \mathbf{a}) = (\mathbf{D}_j)^t \mathbf{a},$$

onde D_j é obtido de 5-16.

- A matriz \mathbf{S}_N de tamanho $l \times l$:

$$\mathbf{S}_N = \sum_{j=1}^N \mathbf{D}_j \mathbf{n}_{1j} \mathbf{n}_{1j}^t \mathbf{D}_j^t$$

Assim, o problema fica descrito por:

$$\begin{aligned} \min_{\mathbf{a}} \{ \mathbf{a}^t \mathbf{S} \mathbf{a} + \mu \mathbf{a}^t \mathbf{S}_T \mathbf{a} + \mu \mathbf{a}^t \mathbf{S}_N \mathbf{a} \}, \\ \text{sujeito a: } \|\mathbf{a}\|^2 = 1 \end{aligned} \quad (5-20)$$

3. Para obter uma aproximação para $F_3(\mathbf{x})$, devemos ter $\nabla P_3^{(d)}(\mathbf{x})$ ortogonal a $\nabla P_1^{(d)}(\mathbf{x})$ e $\nabla P_2^{(d)}(\mathbf{x})$, os quais serão denotados por \mathbf{n}_1 e \mathbf{n}_2 , respectivamente. Os coeficientes de $P_1^{(d)}$ e $P_2^{(d)}$ foram obtidos nos passos 1 e 2, respectivamente. O problema de minimização é o mesmo do caso anterior, porém com a matriz \mathbf{S}_N modificada. Denotemos:

$$\mathbf{S}_N = \sum_{i=1}^2 \left(\sum_{j=1}^N \mathbf{D}_j \mathbf{n}_{ij} \mathbf{n}_{ij}^t \mathbf{D}_j^t \right).$$

4. O processo para determinar uma aproximação para $F_4(\mathbf{x}), \dots, F_{n-m}(\mathbf{x})$ é análogo. Vamos obter uma aproximação para $F_{n-m}(\mathbf{x})$. Denotando os vetores gradientes de $1, \dots, n - m - 1$ pelo conjunto $\mathcal{N} = \{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_{n-m-1}\}$, teremos o mesmo problema de minimização, mas, com a matriz \mathbf{S}_N descrita por:

$$\mathbf{S}_N = \sum_{i=1}^{n-m-1} \left(\sum_{j=1}^N \mathbf{D}_j \mathbf{n}_{ij} \mathbf{n}_{ij}^t \mathbf{D}_j^t \right).$$

Algoritmo 2: Aproximação da variedade implícita

Dados:

- a dimensionalidade m da variedade M que queremos aproximar;
- conjunto de pontos $\mathbb{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \in M \subset \mathbb{R}^n$;
- o espaço tangente a variedade em cada ponto, conforme dado em 5-1, isto é, $\mathcal{T} = \{T_{\mathbf{x}_1}M, T_{\mathbf{x}_2}M, \dots, T_{\mathbf{x}_N}M\}$, onde $T_{\mathbf{x}_j}M = \{t_{j1}, t_{j2}, \dots, t_{jm}\}$ e cada $t_{jh}, 0 \leq j \leq N, 0 \leq h \leq m$ é um vetor tangente unitário.
- d : grau do polinômio
- μ : peso dado aos termos \mathbf{S}_T e \mathbf{S}_N

1. Calcule:

- \mathbf{S} como em 5-6;
- \mathbf{S}_T como em 5-15;
- A matriz D_j como em 5-16

2. **for** $i \in [1, n - m]$:

if $i = 1$:

Obtenha os coeficientes \mathbf{a}_1 de $P_1^{(d)}$ resolvendo 5-17.

else:

Calcule:

$$\mathbf{S}_N = \sum_{t=1}^{i-1} \left(\sum_{j=1}^N \mathbf{D}_j \mathbf{n}_{tj} \mathbf{n}_{tj}^t \mathbf{D}_j^t \right).$$

Obtenha os coeficientes \mathbf{a}_i de $P_i^{(d)}$ resolvendo 5-20.

$$P_i^{(d)}(\mathbf{x}) \leftarrow \mathbf{a}_i^t \mathbf{S} \mathbf{a}_i.$$

$\mathbf{n}_i \leftarrow \nabla P_i^{(d)}(\mathbf{x})$ calculado como em 5-8.

Calcule $\mathbf{n}_{ij} \leftarrow \nabla P_i^{(d)}(x_j) / \|\nabla P_i^{(d)}(x_j)\|, \quad \forall 1 \leq j \leq N.$

Para realizar a aproximação descrita, será implementado o método da partição da unidade, com o objetivo de particionar o domínio e obter aproximações locais para a variedade, e, combiná-las a fim de conseguir uma aproximação global. Para a subdivisão do domínio serão apresentadas duas alternativas: uma partição no \mathbb{R}^n chamada de partição 2^n -tree ou malha 2^n -ádica, e uma subdivisão do domínio que se baseia no particionamento binário do espaço (*Binary Space Partitioning* ou *BSP Trees*).

5.2

Partição da Unidade

A partição da unidade é uma ferramenta matemática muito útil que permite particionar o conjunto de dados em vários subconjuntos, tal que os dados de cada subconjunto possam ser aproximados por funções implícitas simples, as quais são combinadas para obtermos uma aproximação global. Propriedades importantes tais como o erro máximo e ordem de convergência podem ser herdadas do comportamento local. A ideia básica da aproximação por partição da unidade consiste em:

1. dividir o domínio em partes;
2. aproximar os dados em cada subdomínio, separadamente;
3. obter uma aproximação global através de uma combinação ponderada das soluções locais, pelo uso de funções suaves não negativas que correspondem aos pesos e somam para um em cada ponto do domínio.

Mais precisamente, vejamos a definição dada por Lima (41).

Definição 5.3. *Seja M uma variedade de classe C^r . Uma partição da unidade de classe C^k ($k \leq r$) em M é uma família de funções $(\varphi_\alpha)_{\alpha \in A}$, de classe C^k , tais que:*

- (i) *Para todos os $p \in M$ e $\alpha \in A$, $\varphi_\alpha(p) \geq 0$; (A é um conjunto de índices)*
- (ii) *A família $\mathcal{C} = (\text{supp}(\varphi_\alpha))_{\alpha \in A}$ é localmente finita em M ;*
- (iii) *Para todo $p \in M$, tem-se $\sum_{\alpha \in A} \varphi_\alpha(p) = 1$.*

Algumas observações:

- Dados um espaço topológico X e uma aplicação $f : X \rightarrow \mathbb{R}^m$, o *suporte* de f é, por definição, o *fecho* do conjunto:

$$\{x \in X | f(x) \neq 0\}.$$

A notação $supp(f)$ indica o suporte de f . Dado $x \in X$, dizer que $x \notin supp(f)$ significa que f se anula em todos os pontos de uma vizinhança de x .

- $0 \leq \varphi_\alpha(p) \leq 1$.
- Toda partição da unidade em uma variedade compacta é finita.

Seja Ω um domínio limitado em um espaço euclidiano e considere um conjunto de funções com suporte compacto não negativo φ_α tal que:

$$\sum_{\alpha} \varphi_{\alpha} \equiv 1 \text{ em } \Omega.$$

Considere um conjunto de funções com suporte compacto \mathcal{F}_α definidas em $supp(\varphi_\alpha)$. Cada função pertencente a \mathcal{F}_α estaria representando uma aproximação local para os pontos do conjunto de entrada que pertencem a $supp(\varphi_\alpha)$.

Uma aproximação global para a função $f : \Omega \rightarrow \mathbb{R}^n$ poderia ser obtida do seguinte modo:

$$f(\mathbf{x}) \simeq \sum \varphi_{\alpha}(\mathbf{x}) f_{\alpha}(\mathbf{x}). \quad (5-21)$$

onde $f_{\alpha} \in \mathcal{F}_{\alpha}$.

Considere um conjunto de funções não-negativas com suporte compacto tal que:

$$\Omega \subset \cup_{\alpha} supp(w_{\alpha}).$$

As funções partição da unidade φ_α podem ser obtidas do seguinte modo:

$$\varphi_{\alpha}(\mathbf{x}) = \frac{w_{\alpha}(\mathbf{x})}{\sum_{\alpha \in A} w_{\alpha}(\mathbf{x})}. \quad (5-22)$$

A aproximação utilizando as equações 5-21 e 5-22 constituem a base da partição da unidade. Essas equações são utilizadas no algoritmo MPU proposto por Oh-take et al em (49). Este algoritmo foi originalmente empregado na construção de uma aproximação para superfícies implícitas a partir de um conjunto de pontos e normais no \mathbb{R}^3 . O método utiliza partição da unidade para obter uma aproximação global para a fronteira de um objeto, combinando aproximações

locais e, emprega uma subdivisão hierárquica do domínio baseada em uma *octree*. O nosso objetivo é efetuar uma partição do domínio baseada em malhas 2^n -ádicas (11), isto é uma generalização para o espaço n -dimensional das *quad-trees* no \mathbb{R}^2 e *octrees* no \mathbb{R}^3 . Ademais, também será utilizada uma variação da *BSP-Tree*(19).

5.2.1

Malha 2^n -ádica

Uma malha 2^n -ádica recursivamente subdivide os dados volumétricos n -dimensionais em 2^n subvolumes e são frequentemente usadas em aproximações multiescala. Considere uma caixa n -dimensional¹ $\Xi \subset \mathbb{R}^n$. A malha 2^n -ádica é uma hierarquia infinita de subdivisões $\mathcal{H} = \mathcal{H}_0 \cup \mathcal{H}_1 \cup \dots$ obtidas dividindo perpendicularmente cada célula em 2^n células filho iguais por n hiperplanos ortogonais a cada um dos eixos coordenados (11). Uma malha 2^n -ádica tem 2^{np} células no nível p . No caso bidimensional a malha 2^n -ádica é conhecida como *quadtree* e no caso tridimensional como *octree*.

Consideremos o método MPU, o qual constrói uma função implícita que globalmente aproxima os pontos de entrada. Daremos uma descrição do método tomando uma malha 2^n -ádica como um esquema hierárquico para guiar a subdivisão do domínio. Inicialmente, os pontos são transladados para tornar o seu centro de massa como sendo a origem do sistema de coordenadas. Em seguida, os pontos são escalonados de tal maneira que a caixa n -dimensional $\Xi = [-1, 1]^n$ contenha todos os pontos. Denotamos por \mathbb{X} o conjunto dos pontos após estas transformações.

A construção da malha 2^n -ádica é realizada através de um procedimento recursivo, onde a subdivisão de cada nó é controlada pelo erro de aproximação local. O critério de refinamento para um determinado nó v consiste em calcular o erro local de aproximação e, caso este erro seja maior que uma dada tolerância, então o nó será subdividido em 2^n novos nós e recursivamente repete-se o mesmo teste para cada um dos nós filhos.

Cada nó v na malha 2^n -ádica está associado a uma função peso ω_v com um suporte compacto que é usado na partição da unidade. O suporte compacto de ω_v é definido como uma bola aberta² de raio r_v e centrada no centro geométrico do nó. Esse raio é escolhido como sendo proporcional ao tamanho da diagonal d_v do cubo n -dimensional. Conforme demonstrado em (39), $c\sqrt{n}$ é o diâmetro do cubo de aresta c em \mathbb{R}^n . Assim, a maior diagonal

¹Uma caixa n -dimensional Ξ em \mathbb{R}^n é o produto cartesiano de n intervalos abertos limitados I_0, I_1, \dots, I_{n-1} e $\Xi = \prod_{i=0}^{n-1} I_i$,

²consideramos a norma euclidiana

do hipercubo mede $c\sqrt{n}$ e o raio será obtido proporcional a este valor, isto é, consideraremos $r_v = 0.75d_v$, onde $d_v = c\sqrt{n}$.

Adotaremos duas funções peso:

1. A função peso gaussiano ω_v adaptada do kernel gaussiano usado por Gingold e Monaghan (20):

$$\omega_v(\mathbf{x}, r) = \alpha e^{-q^2},$$

onde:

$$\alpha = \frac{1}{r\sqrt{\pi}} \text{ (dimensão 1)}, \alpha = \frac{1}{r^2\sqrt{\pi}} \text{ (dimensão 2)} \text{ e } \alpha = \frac{1}{r^3\sqrt[3]{\pi^2}} \text{ (dimensão 3)}$$

$$q = \frac{\|\mathbf{x}-x_i\|}{r}.$$

Consideramos: $\alpha = \frac{1}{r_v^n \sqrt[3]{\pi^2}}$, $q = \frac{3\|\mathbf{x}-c_v\|}{2r_v}$, onde c_v é o centro do hipercubo gerado durante o processo de subdivisão do domínio e r_v é o raio suporte da função peso.

2. A função peso polinomial adaptada do kernel quadrático usado por Jhonson et al. (26):

$$\omega_v = \frac{1}{4}(q-2)^2\beta,$$

$$\text{onde } q = \frac{3\|\mathbf{x}-c_v\|}{2r_v} \text{ e } \beta = \frac{1}{r_v^n \pi^{n/2}}$$

Qualquer que seja o peso adotado, o valor ω_v é zero fora da região suporte, isto é, caso a distância do ponto ao centro do nó v seja maior do que r_v , então teremos $\omega_v(\mathbf{x}) = 0$. Conforme a equação 5-22, a função partição da unidade φ_v associada com a folha v é dada por:

$$\varphi_v(\mathbf{x}) = \frac{w_v(\mathbf{x})}{\sum_{t=1}^L w_t(\mathbf{x})},$$

onde L é o número de nós folhas. Em cada nó um problema de mínimos quadrados é resolvido considerando apenas os pontos \mathbb{X} que pertençam a sua região suporte. Logo, em cada nó da árvore, a aproximação da variedade M^m é obtida através da aproximação de $(n-m)$ funções polinomiais de grau d , cujos coeficientes são determinados utilizando-se os pontos \mathbb{X} que estão no suporte compacto da função φ_v . Esse subconjunto de pontos é denotado por \mathbb{X}_v e denotamos por N_v o número de elementos em \mathbb{X}_v . A aproximação local é feita conforme os passos descritos na subseção 5.1. O erro de aproximação local de cada função F_i no nó v é considerado como sendo a média do quadrado das distâncias algébricas dos pontos de \mathbb{X}_v :

$$\text{erro}\{F_i, v\} = \frac{1}{N_v} \{\mathbf{a}_i^t \mathbf{S}_v \mathbf{a}_i\}. \quad (5-23)$$

Consideramos o erro de aproximação local no nó v como o máximo dos erros calculados para cada uma das funções F_i nesse nó, isto é,

$$\text{erro}_v = \max_{1 \leq i \leq (n-m)} \{\text{erro}\{F_i, v\}\}. \quad (5-24)$$

Estabelecemos uma tolerância ϵ e, se o erro dado por erro_v for maior do que esse valor e, caso a árvore não tenha atingido o seu nível máximo, o nó v deverá ser dividido.

Finalmente, para avaliar cada uma das $(n-m)$ funções implícitas F_i que aproximam a variedade M , basta combinar as aproximações locais F_{iv} com as funções partição da unidade φ_v :

$$F_i(\mathbf{x}) = \frac{\sum_{v=0}^L \omega_v(\mathbf{x}) F_{iv}(\mathbf{x})}{\sum_{t=1}^L \omega_t(\mathbf{x})}, \quad 1 \leq i \leq (n-m). \quad (5-25)$$

5.2.2

Árvores de partição binária do espaço (BSP-Trees) e funções de transição suaves

O método BSP-Tree conforme trabalho de Lage et al. em (31), combina a flexibilidade de árvores de partição binária do espaço com a robustez estatística das árvores de regressão com transição suave. Em aprendizagem estatística, métodos baseados em árvores geram modelos para classificação e regressão que são simples e poderosos. O método baseado em árvores mais popular é o modelo CART (*Classification and Regression Tree*) (10). O modelo STR-Tree, proposto por Rosa et al. (15), combina aspectos do CART e STR (*Smooth Transition Regression*) e tem sido aplicado para análise de séries temporais com mudanças de regimes. Na tese de Kubrusly (30), é apresentado um novo método baseado em árvores, no qual o domínio é particionado por regiões mais flexíveis definidas implicitamente. Vejamos como funciona a partição do espaço com o modelo BSP-Tree e o uso das funções de transição na aproximação implícita de uma variedade e deixemos a parte de regressão para o próximo capítulo.

A técnica de partição binária do espaço é um método simples e eficiente para subdividir um dado domínio inicial n -dimensional $\mathcal{S} \subset \mathbb{R}^n$ em conjuntos convexos, de maneira a combinar a geometria de um dado conjunto de pontos de entrada. A construção da árvore BSP é um processo que considera o espaço inicial \mathcal{S} e o divide em dois novos subespaços por um hiperplano que intercepta o seu interior. Estes dois novos subespaços, por sua vez, podem ser

particionados por outros hiperplanos e o processo recursivamente continua até que algum critério de parada seja alcançado.

Considere o domínio inicial $[0, 1]^n$. O modelo BSP-Tree particiona o espaço em L regiões convexas $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_L$. O processo de subdivisão recai em dois critérios, a saber, como encontrar o melhor hiperplano separante e quando parar o processo de subdivisão. Os critérios de parada do processo serão abordados posteriormente. Com relação ao hiperplano separante, um hiperplano em um espaço de dimensão n significa um subespaço de dimensão $(n - 1)$ que divide o espaço original em dois semi-espacos. No método apresentado, o hiperplano se baseia nos pontos de entrada e é determinado como sendo perpendicular a primeira componente principal que passa pelo centro de massa dos pontos de entrada contidos na região a ser dividida. Assim, em cada região será realizada uma análise dos componentes principais de forma a obter a primeira componente principal, tomaremos o hiperplano separante como sendo perpendicular a este vetor, o qual deverá passar pelo centro de massa dos pontos contidos na região.

Após a partição do domínio, cada uma das funções locais, ajustadas em cada uma das regiões \mathcal{R}_k , podem ser combinadas para fornecer uma aproximação global para o conjunto de dados de entrada, usando funções de transição associadas com todas as informações geométricas que têm sido utilizadas para particionar o espaço de dados. Em Li et al. (38), é apresentado um método construtivo para ajustar tanto uma curva quanto uma superfície, implícita e explícita, a um conjunto de pontos esparsos usando funções de transição e, nos basearemos neste trabalho para construir variedades implícitas usando funções de transição.

Seja \mathcal{D} uma região no espaço euclidiano real \mathbb{R}^n e, considere:

$$\chi_{\mathcal{D}}(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \mathcal{D}, \\ 0, & \mathbf{x} \notin \mathcal{D}, \end{cases}$$

a função característica de uma região \mathcal{D} .

Considere $f(\mathbf{x})$ uma função definida em \mathbb{R}^n . Suponha que o espaço de dados está particionado em n regiões disjuntas $\mathcal{D}_0, \mathcal{D}_1, \dots, \mathcal{D}_{n-1}$, isto é, $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$, $i \neq j$ e $\mathbb{R}^n = \cup_{i=0}^{n-1} \mathcal{D}_i$.

Portanto, se a forma geométrica sobre a região \mathcal{D}_v está representada implicitamente como $F_v(\mathbf{x}) = \mathbf{0}$, onde $F_v(\mathbf{x}) = (F_{1v}(\mathbf{x}), F_{2v}(\mathbf{x}), \dots, F_{(n-m)v}(\mathbf{x})) = \mathbf{0}$ então a forma geométrica global sobre \mathbb{R}^n pode ser escrita como:

$$F_j(\mathbf{x}) = \sum_{v=0}^{n-1} \chi_{\mathcal{D}_v}(\mathbf{x}) F_{jv}(\mathbf{x}), \quad \forall 1 \leq j \leq (n - m).$$

O problema com a função $\chi_{\mathcal{D}_v}(\mathbf{x})$ é que a mesma não é contínua. A forma combinada representada implicitamente por $F(\mathbf{x}) = \mathbf{0}$ pode ser fragmentada. Para superar esta dificuldade, funções de transição suaves podem ser construídas. Uma função de transição suave é definida como uma aproximação suave para a função característica de um subconjunto de \mathbb{R}^n . Em (38), são apresentadas funções degrau unitária (também chamadas de funções degrau *Heaviside*) e suas aproximações suaves e, em seguida, é exibida a construção de funções de transição suaves gerais, usando funções degrau unitárias combinadas com funções distância.

Definição 5.4. *Um mapeamento $g : \mathbb{R}^n \rightarrow [0, 1]$ é chamado uma função de transição com respeito a uma região \mathcal{D} em um espaço euclidiano \mathbb{R}^n se g assume o valor 1 dentro da região \mathcal{D} e o valor 0 fora dessa região, exceto para aqueles pontos que estão na fronteira da região.*

Definição 5.5. *Seja $\mathcal{G} = \{g_t | t = 0, 1, \dots, n - 1\}$ um conjunto de funções de transição definidas no espaço euclidiano \mathbb{R}^n . \mathcal{G} é dito ser um particionamento de \mathbb{R}^n , se para cada $\mathbf{x} \in \mathbb{R}^n$, temos:*

$$\sum_{t=0}^{n-1} g_t(\mathbf{x}) = 1.$$

Seja $\partial\mathcal{D}$ o bordo de uma região \mathcal{D} e $d_{\mathcal{D}} : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função distância com relação ao bordo, contínua, com sinal e que satisfaz:

$$d_{\mathcal{D}}(\mathbf{x}) \begin{cases} > 0, & \text{se } \mathbf{x} \in \mathcal{D} - \partial\mathcal{D}, \\ = 0, & \text{se } \mathbf{x} \in \partial\mathcal{D}, \\ < 0, & \text{se } \mathbf{x} \in \mathcal{D}^c. \end{cases}$$

A função distância com sinal, acima descrita, pode ser combinada com uma função degrau unitária $s(t)$ para definir uma função $g_{\mathcal{D}}(\mathbf{x})$, correspondendo a região \mathcal{D} . Assim,

$$g_{\mathcal{D}}(\mathbf{x}) = s(d_{\mathcal{D}}(\mathbf{x})) = s \circ d_{\mathcal{D}}(\mathbf{x})$$

Algumas observações importantes incluem:

- Se $d_{\mathcal{D}}(\mathbf{x})$ é uma função distância com sinal então $-d_{\mathcal{D}}(\mathbf{x})$ também é uma função distância com sinal.
- Seja $s(t)$ uma função degrau unitária suave satisfazendo $s(t) + s(-t) = 1$ então tanto $g_{\mathcal{D}}(\mathbf{x}) = s(d_{\mathcal{D}}(\mathbf{x}))$ quanto $g_{\mathcal{D}^c}(\mathbf{x}) = s(d_{\mathcal{D}^c}(\mathbf{x})) = s(-d_{\mathcal{D}}(\mathbf{x})) = 1 - s(d_{\mathcal{D}}(\mathbf{x})) = 1 - g_{\mathcal{D}}(\mathbf{x})$ são funções de transição.

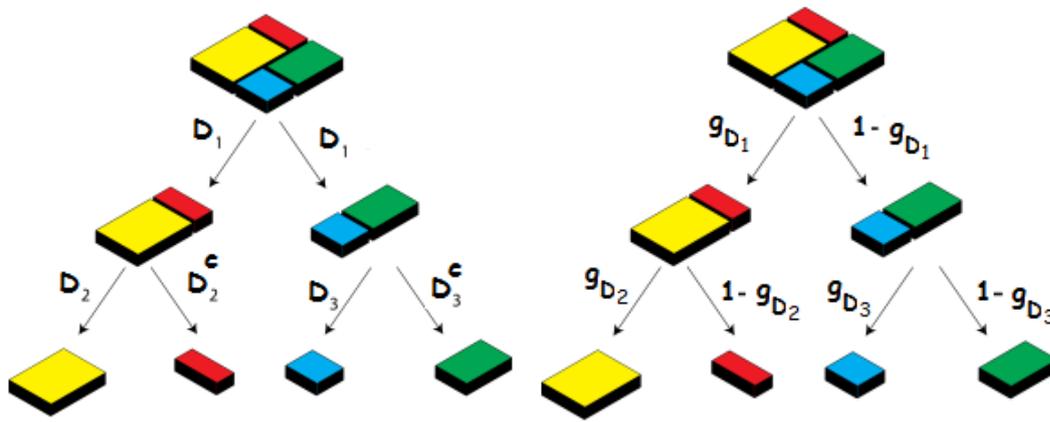


Figura 5.2: Árvore com suas funções de transição em cada região do domínio.

- Sejam g_{D_1} e g_{D_2} funções de transição com relação às regiões $D_1 \subset \mathbb{R}^n$ e $D_2 \subset \mathbb{R}^n$, respectivamente. A função $g_{D_1}g_{D_2}$ é também uma função de transição com relação a região $D_1 \cap D_2$.

As demonstrações dessas observações são simples e podem ser obtidas em (38) e em (30).

As funções de transição podem ser usadas para combinar formas geométricas representadas tanto implicitamente quanto explicitamente. Ao usar funções de transição, um procedimento construtivo pode ser facilmente desenvolvido para construir formas geométricas complicadas. A ideia básica consiste em particionar o espaço de dados em regiões menores com formas geométricas simples até que a forma representada pelos dados em cada sub-região seja simples o suficiente. Para cada subconjunto de dados, a aproximação local da variedade $M^m \subset \mathbb{R}^n$ é obtida através da aproximação de $(n - m)$ funções polinomiais de grau d , conforme os passos descritos na subseção 5.1. Por fim, essas funções locais podem ser combinadas para dar um ajuste global ao conjunto de dados total, usando funções de transição associadas com todas as informações geométricas locais utilizadas na partição do domínio.

Seja \mathcal{R}_k a região definida pela folha k , que é a interseção entre as regiões definidas pelo caminho entre a raiz e a folha em questão. A função de transição B_k , associada à essa região, pode ser definida pelo produto das funções de transição calculadas no caminho da raiz até essa folha. A figura 5.2 apresenta um exemplo de estrutura de árvore e faz sua relação com as funções de transição.

As regiões \mathcal{R}_k definidas na árvore em 5.2 são:

$$\mathcal{R}_1 = \mathcal{D}_1 \cap \mathcal{D}_2 \quad \mathcal{R}_2 = \mathcal{D}_1 \cap \mathcal{D}_2^c \quad \mathcal{R}_3 = \mathcal{D}_1^c \cap \mathcal{D}_3 \quad \mathcal{R}_4 = \mathcal{D}_1^c \cap \mathcal{D}_3^c$$

Cada uma das funções de transição B_k associadas a cada uma das regiões definidas anteriormente são determinadas pelo produto das funções de transição que estão no caminho da raiz até a folha k .

$$\begin{aligned} B_1(\mathbf{x}) &= g_{\mathcal{D}_1} g_{\mathcal{D}_2} \\ B_2(\mathbf{x}) &= g_{\mathcal{D}_1} (1 - g_{\mathcal{D}_2}) \\ B_3(\mathbf{x}) &= (1 - g_{\mathcal{D}_1}) g_{\mathcal{D}_3} \\ B_4(\mathbf{x}) &= (1 - g_{\mathcal{D}_1}) (1 - g_{\mathcal{D}_3}) \end{aligned}$$

A construção das funções de transição naturalmente definem uma partição da unidade, conforme proposição exibida a seguir.

Proposição 5.1. *Seja uma árvore com L folhas. $\sum_{k=1}^L B_k(\mathbf{x}) = 1, \quad \forall \mathbf{x} \in \mathbb{R}^n.$*

A demonstração pode ser obtida em (30).

São propostas duas funções de transição:

- Função de transição gaussiana:

$$g(\mathbf{x}) = \frac{1}{1 + e^{-\lambda d}}$$

- Função de transição polinomial, proposta em (31):

$$g(\mathbf{x}) = \begin{cases} 0 & \text{se } d \leq \frac{-1}{\lambda} \\ 1 & \text{se } d \geq \frac{1}{\lambda} \\ \frac{1}{2} + \frac{15\lambda}{16}d - \frac{5\lambda^3}{8}d^3 + \frac{3\lambda^5}{16}d^5 & \text{caso contrário} \end{cases}$$

As funções apresentadas dependem do parâmetro λ a ser definido pelo usuário e da distância d ao hiperplano separante. Recordemos que o hiperplano separante é perpendicular a primeira componente principal que passa pelo centro de massa dos pontos contidos na região a ser dividida. Se chamamos de \mathbf{w} o vetor da primeira componente principal, temos: $d = \mathbf{w}^t \mathbf{x} - b$, onde b denota o valor por onde passa o hiperplano separante.

Considere a árvore construída com L folhas tal que $\{\mathcal{R}_k\}_{k=1}^L$ é uma partição de \mathbb{R}^n . Seja $B_k(\mathbf{x}), 1 \leq k \leq L$ a função de transição determinada pelo produto das funções de transição que estão no caminho da raiz até a folha

k . A aproximação global de cada uma das funções $F_i(\mathbf{x}) = 0$, $i = 1, \dots, n - m$ que representam implicitamente a variedade M^m será dada por:

$$F_i(\mathbf{x}) = \sum_{k=1}^L B_k(\mathbf{x}) F_{ik}(\mathbf{x}), \quad \forall 1 \leq i \leq n - m \quad (5-26)$$

onde F_{ik} é a aproximação local para a região \mathcal{R}_k do domínio.

A seguir veremos uma proposição importante com relação ao erro na aproximação global para cada função $F_i(\mathbf{x})$, $1 \leq i \leq (n - m)$.

Proposição 5.2. *Seja F_i uma função real definida em \mathbb{R}^n e $\{\mathcal{R}_k\}_{k=1}^L$ ser uma partição de \mathbb{R}^n . Seja \mathbb{X} ser um conjunto de pontos amostrados da forma geométrica definida pela equação $F_i(\mathbf{x}) = 0$ com presença de ruído. Suponha que em cada subconjunto \mathcal{R}_k e em sua vizinhança \mathcal{N}_k , uma equação $F_{ik}(\mathbf{x}) = 0$ possa ser ajustada com erro de ajuste em cada ponto $\mathbf{x} \in \mathcal{R}_k \cup \mathcal{N}_k$ menor que um dado valor ϵ , isto é, $|F_{ik}(\mathbf{x})| \leq \epsilon$. Considere B_1, B_2, \dots, B_k funções de transição correspondentes aos subconjuntos $\{\mathcal{R}_k\}_{k=1}^L$, tais que $B_k = 0$, se $\mathbf{x} \notin \mathcal{R}_k \cup \mathcal{N}_k$ e $\sum_{k=1}^L B_k(\mathbf{x}) = 1$. Então, o erro de ajuste em cada dado para a função*

$$F_i(\mathbf{x}) = \sum_{k=1}^L B_k(\mathbf{x}) F_{ik}(\mathbf{x})$$

também deve ser menor do que ϵ .

Prova. Para qualquer ponto $\mathbf{x} \in \mathbb{X}$ temos:

$$\begin{aligned} |F_i(\mathbf{x})| &= \left| \sum_{k=1}^L B_k(\mathbf{x}) F_{ik}(\mathbf{x}) \right| \\ &= \left| \sum_{\mathbf{x} \in \mathcal{X}_k \cup \mathcal{N}_k} B_k(\mathbf{x}) F_{ik}(\mathbf{x}) \right| \\ &\leq \sum_{\mathbf{x} \in \mathcal{X}_k \cup \mathcal{N}_k} B_k(\mathbf{x}) |F_{ik}(\mathbf{x})| \\ &\leq \sum_{\mathbf{x} \in \mathcal{X}_k \cup \mathcal{N}_k} B_k(\mathbf{x}) \epsilon \leq \epsilon \end{aligned}$$

□

5.2.3

Critérios de parada

Apresentaremos os critérios de parada da árvore obtida tanto pela malha 2^d -ádica quanto por uma árvore com particionamento binário e funções de transição suave.

1º Critério de parada: Profundidade da Árvore (l_{max});

A escolha do nível máximo da árvore influencia no número de subdivisões da mesma. Assim, tal valor deve ser escolhido de maneira que o domínio

tenha subdivisões suficientes para representar bem os dados de entrada. Por sua vez, esse valor também deve levar em consideração o grau do polinômio que deve ser ajustado aos dados, para que se tenha dados suficientes em cada nó de modo que a aproximação possa ser realizada.

2º Critério de parada: Número de Elementos mínimos em cada nó (n_{lim});

Essa escolha depende do grau do polinômio que será utilizado para aproximar os dados em cada nó. Assim, devemos ter $n_{lim} \geq l = \frac{(n+d)!}{n!d!}$, onde d é o grau do polinômio e n representa a dimensão dos dados de entrada.

3º Critério de parada: Cota superior para o erro de aproximação local da variedade (ϵ);

A escolha da cota ϵ determina a precisão do ajuste local desejado e influencia na partição do domínio, pois uma cota muito pequena exigirá uma maior partição do domínio. Em cada nó da árvore, temos que aproximar $(n-m)$ funções $F_i(\mathbf{x}) = 0$ aos dados. Logo, teremos $n-m$ erros locais em cada folha k denotados por $erro\{F_{ik}\}$ e o erro local na folha k será calculado como o máximo dos erros locais das $(n-m)$ funções $F_i(\mathbf{x})$, isto é:

$$Erro_k = \max_{1 \leq i \leq (n-m)} \{erro\{F_{ik}\}\}.$$

4º Critério de parada: Cota para o desvio padrão dos valores de y em cada nó (ϵ).

Esse último critério está relacionado a um dos objetivos desse estudo, isto é, a árvore deve ser construída de tal forma que a aproximação da variedade garanta uma boa previsão das saídas y . Essa aplicação será desenvolvida no próximo capítulo.

5.2.4

Algoritmos: Malha 2^n -ádica e Árvore BSP

Apresentamos um esboço dos algoritmos para os dois métodos de subdivisão de domínio estudados. Os dados a seguir serão utilizados em ambos os métodos, exceto quando virem precedidos do nome do método entre parênteses.

Dados:

- m : dimensionalidade da variedade;
- conjunto de pontos $\mathbb{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \in [0, 1]^n \subset \mathbb{R}^n$;
- conjunto de dados $\mathbb{Y} = \{y_1, \dots, y_N\} \in \mathbb{R}$;
- o espaço tangente a variedade em cada ponto, conforme dado em 5-1, isto é, $\mathcal{T} = \{T_{\mathbf{x}_1}M, T_{\mathbf{x}_2}M, \dots, T_{\mathbf{x}_N}M\}$, onde $T_{\mathbf{x}_j}M = \{t_{j1}, t_{j2}, \dots, t_{jm}\}$ e cada $t_{jh}, 0 \leq j \leq N, 0 \leq h \leq m$ é um vetor tangente unitário;
- d : grau do polinômio;
- μ : peso dado aos termos \mathbf{S}_T e \mathbf{S}_N ;
- l_{max} : nível máximo da árvore;
- l : número de coeficientes do polinômio;
- ϵ : cota do erro de aproximação local;
- (Malha 2^n -ádica) $inf[]$: vetor de tamanho n contendo os ínfimos de cada aresta do hipercubo;
- ε : cota do desvio padrão dos y ;
- (Árvore BSP) $princomp[,]$: matriz de tamanho $(n \times s)$, onde s está relacionado ao número de nós na árvore. Essa matriz armazena em cada coluna o primeiro componente principal de cada nó da árvore;
- (Árvore BSP) $med[]$: vetor que guarda o valor por onde passa o hiperplano separante.

Algoritmo 3: Construção da Malha 2^n -ádica

Passo 1:

if nível $\leq l_{max}$:

- cria um nó;
- $lado = 2^{1-nivel}$;
- $r = (0.75).lado.\sqrt{n}$;
- Calcula \mathcal{I} : conjunto de índices dos pontos do nó que se encontram dentro da bola de raio r e centro $c = inf[t] + lado/2, \forall 1 \leq t \leq n$;

- Vá para o passo 2.

else:

Este nó é uma folha, FIM.

Passo 2: Calcula o número de pontos no nó: tam ;

if $tam < l$:

Este nó é uma folha, FIM.

else:

- Defina A : matriz de tamanho $l \times (n - m)$, cujas colunas armazenarão os coeficientes de $P_i^{(d)}$;
- Calcula \mathbf{a}_i : vetor de coeficientes de $P_i^{(d)}$, obtido resolvendo o algoritmo 2 e armazena o resultado em $A[:, i]$, $\forall 1 \leq i \leq (n - m)$;
- Calcula o erro de aproximação:

$$Erro_aprox = \max_{1 \leq i \leq (n-m)} \{erro\{P_i^{(d)}\}\},$$

onde cada $erro\{P_i^{(d)}\}$ é obtido como em 5-23.

- Vá para o passo 3.

Passo 3: **if** $Erro_aprox > \epsilon$:

Calcula o desvio padrão dos valores de y no nó. Denote $desv_pad$;

if $desv_pad > \epsilon$:

Vá para o passo 4;

else:

Esse nó é uma folha, FIM.

Passo 4: É criado um nó interno. O nó será dividido em 2^n hipercubos, dados lado e $inf[]$;

Considere $inf1$ uma matriz de tamanho $(2^n \times n)$. Essa matriz é construída de maneira que cada linha represente o ínfimo de cada uma das arestas dos 2^n hipercubos. O procedimento é descrito a seguir.

for $col \in [1, n]$:

$row = 0$;

while $row \neq 2^n$:

for $w \in [1, 2^{n-col-1}]$:

$inf1[row, col] = inf[col]$

$row = row + 1$

for $w \in [1, 2^{n-col-1}]$:

$inf1[row, col] = inf[col] + lado$

```

    row = row + 1
  Faça nível=nível+1
  for  $1 \leq t \leq 2^n$  :
     $inf[t] = inf1[t, :]$ ;
    Retorne ao passo 1, isto é, para cada um dos  $2^n$  hipercubos retorne
    ao passo 1.

```

Algoritmo 4: Construção da Árvore de Partição Binária com Funções de Transição Suaves

Passo 1: if nível $\leq l_{max}$:

cria um nó;

Vá para passo 2.

else:

Este nó é uma folha, FIM.

Passo 2: Calcula o número de pontos no nó: tam;

if tam $< l$:

Este nó é uma folha, FIM.

else:

- Defina A : matriz de tamanho $l \times (n-m)$, cujas colunas armazenarão os coeficientes de $P_i^{(d)}$;
- Calcula \mathbf{a}_i : vetor de coeficientes de $P_i^{(d)}$, obtido resolvendo o algoritmo 2 e armazena o resultado em $A[:, i]$, $\forall 1 \leq i \leq (n-m)$;
- Calcula o erro de aproximação:

$$Erro_{aprox} = \max_{1 \leq i \leq (n-m)} \{erro\{P_i^{(d)}\}\},$$

onde cada $erro\{P_i^{(d)}\}$ é obtido como em 5-23.

- Vá para o passo 3.

Passo 3: if $Erro_{aprox} > \epsilon$:

Calcula o desvio padrão dos valores de y no nó. Denote desv_pad;

if desv_pad $> \epsilon$:

Vá para o passo 4;

else:

Esse nó é uma folha, FIM.

Passo 4:

- É criado um nó interno v ;
- Dados os pontos no nó, calcula-se a primeira componente principal e armazena em $princomp[v]$;
- Calcula o valor por onde passa o hiperplano e guarda em $med[v]$;
- Determina o nó filho a esquerda e o nó filho a direita conforme a divisão dada pelo hiperplano;
- Para cada um dos nós filhos, retorna ao passo 1;

Após a construção da árvore, o valor de $princomp[,]$, $med[]$ e o número de folhas serão utilizados para calcular a função de transição e, para obter cada uma das funções $F_i(\mathbf{x})$, conforme fórmula exibida em 5-26. Essas informações serão também utilizadas para determinar uma regressão baseada nesta árvore, conforme será explicado no próximo capítulo.
