



Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Estatística

Dissertação de Mestrado

Determinação do Parâmetro de Suavização Ótimo
na Regressão Geograficamente Ponderada

por

Felipe Franco Mendes

Orientador: Prof. Dr. Alan Ricardo da Silva

Dezembro de 2016

Felipe Franco Mendes

Determinação do Parâmetro de Suavização Ótimo na Regressão Geograficamente Ponderada

Dissertação apresentada ao Departamento de Estatística do Instituto de Ciências Exatas da Universidade de Brasília como requisito à obtenção do título de Mestre em Estatística.

Universidade de Brasília

Brasília, Dezembro de 2016

*“Mesmo as aranhas mais caprichosas
podem deixar um fio frouxo.”*

J.R.R. Tolkien

Agradecimentos

- Agradeço a Deus pela oportunidade de crescimento intelectual e moral;
- Agradeço ao meu professor e orientador Alan Ricardo da Silva, pela oportunidade e as sugestões para os problemas que passamos;
- Agradeço a minha família, especialmente ao meus pais, Elizena e Carlos Fernandes, pela paciência e apoio ao meu estudo;
- Agradeço a minha namorada, Evelyn, por tanto ter me ajudado durante todo esse processo;
- Por fim, agradeço ao pessoal do Departamento de Estatística por estarem sempre dispostos a ajudar e a CAPES pela bolsa concedida para minha dedicação ao mestrado.

Sumário

Agradecimentos	ii
Lista de Figuras	4
Lista de Tabelas	6
Resumo	7
Abstract	8
Introdução	9
1 Regressão Geograficamente Ponderada	15
1.1 Introdução	15
1.2 O modelo de Regressão Geograficamente Ponderada	16
1.3 Estimação do Parâmetro de suavização	19
1.4 Erro Padrão Local	21
1.5 Regressão Poisson Geograficamente Ponderada	23
1.6 Extensões da Regressão Geograficamente Ponderada	24
2 Minimização de funções	26
2.1 Introdução	26
2.2 Método de busca com intervalos iguais (Bisseção)	26
2.3 Otimização por seção áurea (OSA)	28
2.4 Otimização meta-heurística	30
2.5 Algoritmo Busca Harmônica	31
2.6 Algoritmo de Busca Relâmpago	33
2.6.1 Projéteis e propagação da etapa líder	33

2.6.2	Propriedades dos projéteis	34
2.6.3	Modelagem dos projéteis e movimento da etapa líder	35
2.6.3.1	Projéteis de Transição	35
2.6.3.2	Projéteis no espaço	35
2.6.3.3	Projéteis líderes	36
3	Materiais e Métodos	41
3.1	Introdução	41
3.2	Material	41
3.2.1	Dados simulados	43
3.2.2	Casos reais	45
3.3	Método	46
4	Resultados	49
4.1	Exemplos determinísticos	49
4.1.1	Função 1	49
4.1.2	Função 2	52
4.2	Simulação de dados com <i>shape</i> regular	54
4.2.1	Conjunto de dados com $n = 64$	54
4.2.2	Conjunto de dados com $n = 100$	56
4.2.3	Conjunto de dados com $n = 1024$	58
4.3	Simulação de dados com <i>shape</i> aleatório	60
4.3.1	Conjunto de dados com $n = 72$	60
4.3.2	Conjunto de dados com $n = 225$	63
4.4	Estudos de casos reais	65
4.4.1	Dados do estado de Goiás	65
4.4.2	Dados de acidentes de trânsito na cidade de Fortaleza	69
5	Conclusões	72
5.1	Limitações do Trabalho	74
5.2	Sugestões para Trabalhos Futuros	75
	Referências Bibliográficas	76

Lista de Figuras

1	Traço da função f_r	13
2	Função não convexa proveniente do cálculo do Cross-Validation de uma RGP com dados simulados aleatoriamente: (a) $6 \leq h \leq 72$, (b) $20 \leq h \leq 72$. . .	14
1.1	Parâmetro de Suavização.	19
1.2	RGP com parâmetro de suavização w_{ij} ; • pontos dos dados, X pontos de regressão.	19
1.3	RGP com parâmetro de suavização w_{ij} variável; • pontos dos dados, X pontos de regressão.	20
2.1	Método de busca por intervalos iguais	28
3.1	Função f_n , no intervalo $[-1;4]$	42
3.2	Simulação de dados com <i>shape</i> regular de tamanhos: (a) 8×8 , (b) 10×10 , (c) 32×32	43
3.3	Simulação de dados com <i>shape</i> irregular de tamanhos: (a) 72, (b) 225. . . .	44
3.4	Mapa dos municípios dos estados de Pará (a) e São Paulo (b) e seus respectivos centros administrativos (c) e (d).	46
4.1	Etapas do algoritmo OSA aplicado na função f_r nos intervalos: (a) $[9;20]$, (b) $[9;21]$	50
4.2	Divisão da função f_r em três partes.	51
4.3	Divisão da função f_r em quatro partes iguais	53
4.4	Esboço da função CV dos dados simulados com $n = 64$ para os parâmetros de suavização: (a) fixo, (b) adaptável.	54

4.5	Esboço da função CV dos dados simulados com $n = 100$, (a) Fixo: $0 \leq h \leq 7$, (b) Fixo: $0,3 \leq h \leq 7$, (c) Adaptável: $6 \leq h \leq 100$, (d) Adaptável: $30 \leq h \leq 100$	57
4.6	Esboço da função CV dos dados simulados com $n = 1024$, (a) Fixo: $0 \leq h \leq 21$, (b) Fixo: $0,4 \leq h \leq 2$, (c) Adaptável: $6 \leq h \leq 1024$, (d) Adaptável: $1000 \leq h \leq 1024$	59
4.7	Esboço da função CV dos dados simulados aleatoriamente com $n = 72$, (a) Fixo: $0 \leq h \leq 129$, (b) Fixo: $9 \leq h \leq 129$, (c) Adaptável: $6 \leq h \leq 72$, (d) Adaptável: $18 \leq h \leq 72$	62
4.8	Esboço da função CV dos dados simulados aleatoriamente com $n = 225$, (a) Fixo: $0 \leq h \leq 198$, (b) Fixo: $9 \leq h \leq 100$, (c) Adaptável: $6 \leq h \leq 225$, (d) Adaptável: $50 \leq h \leq 225$	64
4.9	Esboço da função CV dos dados de rendimento do estado de Goiás, (a) Fixo: $0 \leq h \leq 888$, (b) Fixo: $8 \leq h \leq 888$, (c) Adaptável: $6 \leq h \leq 242$, (d) Adaptável: $31 \leq h \leq 242$	66
4.10	Estimadores da variável população do modelo RGP com os dados de Goiás, (a) $h = 42$ e (b) $h = 43$, e estimadores significativos com nível de 90%, (c) $h = 42$ e (d) $h = 43$	67
4.11	Estimadores da variável população do modelo RGP com os dados de Goiás, (a) $h = 42$ e (b) $h = 59$, e estimadores significativos com nível de 90%, (c) $h = 42$ e (d) $h = 59$	68
4.12	Esboço da função AICc dos dados de acidentes de trânsito da cidade de Fortaleza, (a) Fixo: $0 \leq h \leq 21000$, (b) Fixo: $200 \leq h \leq 21000$, (c) Fixo: $900 \leq h \leq 1100$, (d) Adaptável: $6 \leq h \leq 126$	70
4.13	<i>Prints</i> dos resultados obtidos através do software GWR4, (a) Definição das variáveis, (b) Resultado da algoritmo OSA, (c) Limites de busca definidos pelo GWR4.	71

Lista de Tabelas

2.1	Algoritmo da Otimização por Seção Áurea	30
2.2	Algoritmo Busca Aleatória	31
2.3	Algoritmo Busca Harmônica	33
2.4	Algoritmo Busca Relâmpago	38
2.5	Continuação do algoritmo Busca Relâmpago	39
2.6	Continuação do algoritmo Busca Relâmpago	40
3.1	Regras para definição dos b 's de cada conjunto.	44
3.2	Detalhes dos conjuntos de dados simulados aleatoriamente de tamanhos $n =$ 72 e $n = 225$	45
3.3	Algoritmo da Otimização por Seção Áurea	47
4.1	Valores da primeira etapa do algoritmo OSA aplicado na função f_r nos intervalos $[9;20]$ e $[9;21]$	50
4.2	Mínimos encontrados pelo algoritmo OSA das 3 divisões da função f_r	51
4.3	Mínimo dos algoritmos OSA, Relâmpago e Harmônico para o intervalo $[9;21]$, tempo de processamento e erro máximo admitido.	52
4.4	Mínimo dos algoritmos OSA, Relâmpago e Harmônico para a função f_n , tempo de processamento e erro máximo admitido.	54
4.5	Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função CV dos dados simulados com $n = 64$, tempo de processamento com parâmetro de suavização fixo e adaptável, e R^2 dos modelos RGP para cada parâmetro. . .	55
4.6	Frequências cruzada das estimativas significativas à 90%, 95%, 99% ou não significativas, dos modelos RGP com $n = 64$ e os parâmetros adaptáveis: $h = 46$ e $h = 47$	56

4.7	Frequências cruzada das estimativas significativas à 90%, 95%, 99% ou não significativas, dos modelos RGP com $n = 64$ e os parâmetros fixos: $h = 1511,42$ e $h = 1512,015$	56
4.8	Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função CV dos dados simulados com $n = 100$, tempo de processamento com parâmetro de suavização fixo e adaptável, e R^2 dos modelos RGP para cada parâmetro.	58
4.9	Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função CV dos dados simulados com $n = 1024$, tempo de processamento com parâmetro de suavização fixo e adaptável, e R^2 dos modelos RGP para cada parâmetro.	60
4.10	Frequências cruzada das estimativas significativas à 90%, 95%, 99% ou não significativas, dos modelos RGP com $n = 1024$ e os parâmetros adaptáveis: $h = 1018$ e $h = 1004$	60
4.11	Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função CV dos dados simulados aleatoriamente com $n = 72$, tempo de processamento com parâmetro de suavização fixo e adaptável, e R^2 dos modelos RGP para cada parâmetro.	61
4.12	Frequências cruzada das estimativas significativas à 90%, 95%, 99% ou não significativas, dos modelos RGP com $n = 72$ e os parâmetros adaptáveis: $h = 63$ e $h = 26$	63
4.13	Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função CV dos dados simulados aleatoriamente com $n = 225$, tempo de processamento com parâmetro de suavização fixo e adaptável, e R^2 dos modelos RGP para cada parâmetro.	65
4.14	Frequências cruzada das estimativas significativas à 90%, 95%, 99% ou não significativas, dos modelos RGP com $n = 225$ e os parâmetros adaptáveis: $h = 71$ e $h = 108$	65
4.15	Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função CV dos dados de Goiás com parâmetro de suavização adaptável e R^2 dos modelos RGP para cada parâmetro.	67
4.16	Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função com critério AICc dos dados dos acidentes de trânsito e tempo de processamento com parâmetro de suavização fixo e adaptável	71

Resumo

Os modelos da Regressão Geograficamente Ponderada são muito sensíveis à escolha do parâmetro de suavização. E essa escolha é tradicionalmente feita através do algoritmo Otimização por Seção Áurea, ou do inglês *Golden Section Search*. Esse algoritmo é aplicado com uma função que quantifica a eficiência do modelo, procurando, portanto, o parâmetro ótimo que resulta no melhor modelo. Neste trabalho, estudou-se o comportamento da função de validação cruzada e verificou-se que ela não é estritamente convexa, o que faz com que o algoritmo Otimização por Seção Áurea possa convergir para mínimos locais. Três algoritmos foram propostos para encontrar o parâmetro de suavização ótimo, sendo eles o algoritmo Relâmpago, o algoritmo Harmônico e uma adaptação do algoritmo Otimização por Seção Áurea. Além disso, foram feitas comparações entre os algoritmos, e foi apresentado a influência da escolha do parâmetro de suavização nos modelos da Regressão Geograficamente Ponderada. Constatou-se que o algoritmo Otimização por Seção Áurea não é o mais adequado nesta situação, pois em mais de um exemplo ele resultou em um parâmetro muito distante do parâmetro de suavização ótimo. Também verificou-se que os modelos com o parâmetro de suavização encontrado incorretamente possuem estimadores com significância muito diferente em relação aos modelos com o parâmetro ótimo. Outro ponto observado foi o tempo demasiado grande que os algoritmos Relâmpago e Harmônico gastam no processamento para encontrar o parâmetro ótimo.

Palavras-chave: parâmetro de suavização; análise espacial; regressão geograficamente ponderada.

Abstract

The Geographically Weighted Regression models are very sensitive to the choice of the bandwidth. And this choice is traditionally made through the Golden Section Search algorithm. This algorithm is applied with a function that quantifies the efficiency of the model, therefore, looking for the optimal parameter that results in the best model. In this work, the behavior of the Cross-Validation function was studied and it was verified that it is not strictly convex, which makes the Golden Section Search algorithm converges to local minimums. Three algorithms were proposed to find the optimal bandwidth: the Lightning algorithm, the Harmonic algorithm and an adaptation of the Golden Section Search algorithm. In addition, comparisons were made between the algorithms, and a set of influence of the choice of the bandwidth in the Geographically Weighted Regression models. It was found that the Golden Section Search algorithm is not the most adequate in this situation, because in more than one example it resulted in a value too far from the optimal bandwidth. It was also verified that the models with the bandwidth found incorrectly have estimators with very different significance in relation to the models with the optimal parameter. Another point observed was the large processing time that the Lightning and Harmonic algorithms spend to find the optimal bandwidth.

Keywords: bandwidth; spatial analysis; geographically weighted regression.

Introdução

Nas mais diversas áreas da ciência o objeto de estudo constitui-se nos fenômenos naturais ou sociais. Entender esses fenômenos pode ser uma tarefa que compara e relaciona características, variáveis ou parâmetros, e busca, essencialmente, trazer benefícios à sociedade.

Inumeráveis caminhos podem ser considerados para encontrar essas relações, e uma estratégia que pode ser adotada é incluir uma estrutura espacial para entender como se comportam no estudo os objetos ou indivíduos mais próximos. Para compreender como a proximidade geográfica pode ser relevante em um estudo, considere um possível caso em que deseja-se definir a demanda de transporte coletivo em cada região de uma cidade.

O número de indivíduos que utilizarão o transporte coletivo poderá ser relacionado com algumas características sociais, por exemplo, média da renda familiar em cada região, a média do número de pessoas das famílias das regiões, população total em cada região, se a região é residencial ou comercial, e muitas outras características podem ser cogitadas.

Relacionar essas características pode resultar em boas estimativas para a demanda. No entanto, possivelmente, pode ocorrer das regiões vizinhas influenciarem na demanda, e ao não ponderar essa influencia as estimativas tornam-se pouco confiáveis. Então o ideal seria incluir uma estrutura espacial nesse relacionamento para obter estimativas mais próximas possíveis do ideal.

Entretanto, apresenta-se um novo problema, pois, se irá considerar as localizações geográficas das regiões, deverá definir como cada região irá atuar nas outras. Presumi-se, no entanto, que as regiões mais próximas terão relacionamentos maiores. Sendo assim, voltando ao contexto do estudo, a tarefa agora é encontrar o critério que defini quais e quantos são os bairros mais próximos que se influenciarão.

Dado que, normalmente, os bairros possuem dimensões diferentes, sendo alguns responsáveis por enormes áreas na cidade e outros com apenas meia dúzia de quadras, um

“círculo de inclusão”, com o centro em pontos estratégicos das linhas do transporte público, mostraria resultados mais confiáveis, pois um bairro com dimensão muito grande pode ter influência de várias regiões de maneiras diferentes. Portanto, considerando um “círculo” que irá definir as regiões próximas, deve-se refletir a respeito do tamanho dele.

Atenta-se, portanto, à importância da escolha perfeita do diâmetro do “círculo” dado que a estimação errônea acarretará em uma estimativa alta ou baixa da demanda do transporte coletivo em uma região, podendo por exemplo, em uma determinada linha de ônibus receber uma frota regular menor que a necessária, diminuindo a frequência que os ônibus passam nos pontos e assim acarretando em superlotações, por outro lado, em outra linha os ônibus circularão vazios.

Vê-se, então, a importância de um estudo que relaciona variáveis ou características, e também o cuidado que deve-se ter com cada fase desse estudo. Um método muito conhecido que realiza essa tarefa é o modelo de regressão linear, que segundo Neter *et al.* (1996), é a forma mais comum para descrever a relação existente entre uma variável resposta ou dependente e um conjunto de variáveis explicativas ou independentes. Esse modelo geralmente possui a forma:

$$y_i = \beta_0 + \sum_{j=1}^k \beta_j x_{ij} + \varepsilon_i \quad (1)$$

em que y_i é a i -ésima observação da variável resposta, $i = 1, \dots, n$, x_{ij} é o valor da j -ésima variável explicativa para a i -ésima observação, $j = 1, \dots, k$, e ε_i é o erro relacionado à i -ésima observação. Supõe-se nesse modelo que os erros são independentes e normalmente distribuídos, com média 0 e variância σ^2 ($\varepsilon_i \sim N(0, \sigma^2)$).

Essa relação é medida através de dados coletados e, geralmente, possui os objetivos: prever a variável independente, y , a partir de variáveis explicativas, x_i ; selecionar variáveis explicativas que mais afetam y ; estimar parâmetros e realizar inferências sobre estes parâmetros. Estas características fizeram essa técnica ser bastante difundida entre todas as áreas da ciência.

O uso constante das regressões nos mais diversos trabalhos foi importante para encontrar outras maneiras de relacionar a variável independente com variáveis dependentes. Alguns exemplos dessas variações são a regressão logística (Hosmer e Lemeshow, 2000), a regressão quantílica (Koenker, 2005), a regressão espacial (Anselin, 1988; Fotheringham *et al.*, 2002) entre outras. Sendo a primeira delas utilizada para respostas com dados categóricos, já a regressão quantílica é utilizada para observar as relações nos quantis dos dados

e não apenas focado na média. A regressão espacial global é utilizada com um parâmetro a mais, que inclui dependência espacial aos dados e portanto, essa técnica pode ser combinada com outras técnicas de regressão, como as regressões logística e quantílica. Contudo, este trabalho foca apenas na regressão espacial, mais especificamente, na Regressão Geograficamente Ponderada, uma área que trabalha dentro da regressão espacial analisando os dados localmente.

A regressão espacial é uma técnica que possibilita um entendimento mais aprimorado em situações estudadas pelo homem e esta técnica possui tamanha flexibilidade que tornou a sua aplicação recorrente em diversas áreas, como geografia, economia, epidemiologia, ciências ambientais e ciências políticas. A estrutura, ou dependência, espacial dos dados possibilita a criação de modelos que levam em consideração a semelhança, mais precisamente a proximidade dos objetos em estudo. Com foco no aspecto físico da espacialidade, sabe-se que a relação entre duas ou mais variáveis em estudos socioeconômicos não mantém-se as mesmas em diferentes regiões, pelo fato de existir diferenças culturais nas diversas regiões do planeta, dos países, e até mesmo dentro de uma cidade, portanto, incluir a espacialidade nos estudos os tornam mais eficazes para melhorar a medição dessas relações de cada estudo.

Nota-se, no entanto, que os modelos globais não conseguem captar essas particularidades, sendo mais úteis em casos onde pode-se constatar homogeneidade entre as regiões, o que raramente pode ser observado, e Fotheringham *et al.* (2000) reforçam essa constatação afirmando que os modelos de regressões globais estimam os parâmetros sem conseguir captar as variações no espaço. Sendo assim, os modelos globais foram atualizados para incorporar a espacialidade, como os modelos da regressão espacial global e a regressão espacial local ou, como dito anteriormente, a Regressão Geograficamente Ponderada (RGP), ou do inglês *Geographically Weighted Regression*.

A principal diferença entre a RGP e a regressão espacial global é que a primeira diferencia-se por fazer uma análise específica de cada região. Neste sentido, dentro da regressão espacial, a regressão espacial global tem um foco maior em encontrar similaridades no espaço, enquanto que a regressão espacial local procura encontrar as diferenças no espaço (Fotheringham *et al.*, 2002).

Uma vantagem da RGP é a possibilidade dela encontrar resultados que podem ser mapeados e analisados dentro de um Sistema de Informações Geográficas (SIG), o que torna possível o entendimento das relações entre os parâmetros no espaço, e fazer investigações

em busca de padrões. Através dessa possibilidade, pode ser possível observar os resultados locais alterarem uma suposição feita a partir de estatísticas globais.

Essa técnica mais refinada, no entanto simples, explora o ajuste de processos não estacionários, calibrando localmente a variação espacial dos coeficientes do modelo de regressão, o que traz um resultado final mais representativo. Para isso, pressupõe-se que as regiões mais próximas influenciem mais nas estimativas de uma regressão do que regiões mais distantes, seguindo o raciocínio da Primeira Lei da Geografia enunciado por Tobler (1979), “todas as coisas são parecidas, mas coisas mais próximas se parecem mais que coisas mais distantes”.

Para Brunson *et al.* (1998), essencialmente, o problema a ser resolvido na técnica RGP é estimar um B_j para cada variável j e cada localidade geográfica i , além disso, deve-se definir uma maneira de trabalhar com a modelagem local: definir as regiões mais próximas de forma discreta ou contínua. Em qualquer um dos casos, uma etapa relevante na RGP é a definição do número de regiões que serão consideradas mais importantes. Se o critério de inclusão considerar muitas regiões, então quase todos os dados serão incluídos na modelagem, e assim, os resultados se aproximariam da modelagem global. E se considerar poucas regiões, poucas, também, serão as observações, resultando em estimativas com grandes erros padrões.

No caso contínuo, a definição das regiões se passa pelo parâmetro de suavização (*bandwidth*), que através de uma função *Kernel*, irá definir um círculo de inclusão dos dados espaciais, além de funcionar, muitas vezes, como um fator de variabilidade da curva da função *Kernel*. Devido essas duas características, encontrar o parâmetro de suavização ótimo torna-se um ponto chave da RGP, pois uma pequena variação do seu valor pode alterar significativamente o resultado final.

Contudo, a estimação do parâmetro de suavização é feita a partir de otimização numérica, com técnicas de minimização de funções, como Otimização por Seção Áurea ou *Golden Section Search*, que é amplamente utilizada em RGP. Esse ponto da Regressão Geograficamente Ponderada será aprofundado nesse trabalho, pois a otimização por Seção Áurea, mesmo que bastante difundida nesta área, pode encontrar mínimos locais, dependendo das entradas do algoritmo, como por exemplo, os valores dos extremos do intervalo de busca. A escolha dos limites do espaço de busca muitas vezes pode influenciar no mínimo encontrado pelo algoritmo, veja por exemplo o caso da função da Equação (2) que está representada na Figura 1 para o $r > 9$. Notavelmente, o mínimo global da função encontra-se próximo

ao $r = 10$, todavia, um mínimo local também é observado próximo ao $r = 16$. Então, utilizando o algoritmo de Otimização por Seção Áurea, definindo os limites de busca em $[9, 20]$, o mínimo encontrado é, possivelmente, o mínimo global $r = 10,01901$. No entanto, ao alterar os limites de busca para $[9, 21]$, o mínimo encontrado passa a ser o mínimo local $r = 16,31579$.

$$f_r(r) = \left(\sin(r) - \frac{\sin(2r)}{2} + \frac{\sin(3r)}{3} - \frac{\sin(4r)}{4} + 4 \right) \frac{r^2}{r+1}. \quad (2)$$

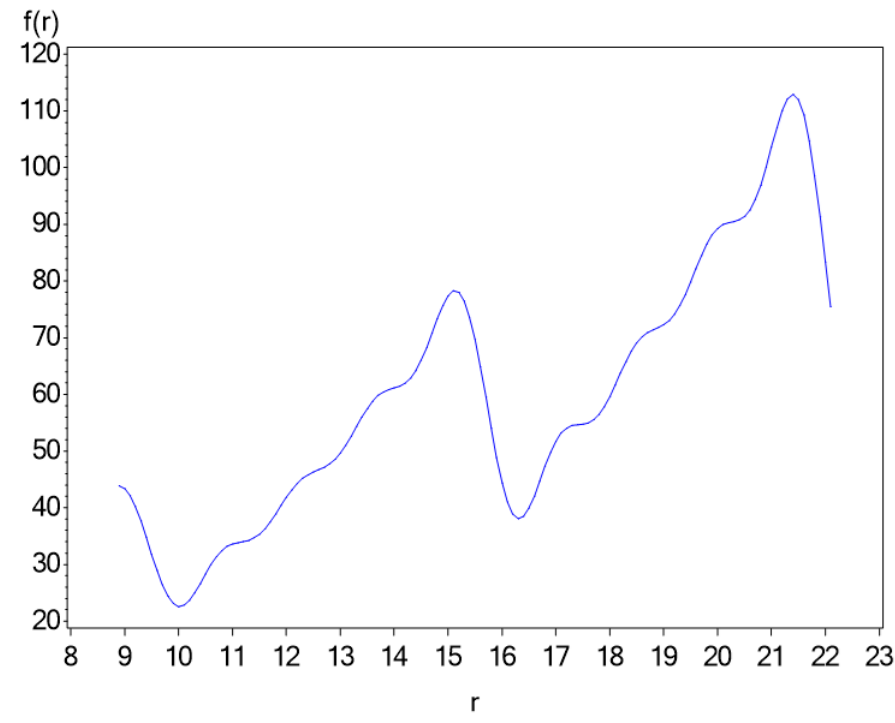


Figura 1: Traço da função f_r

Todavia, outro caso onde o algoritmo pode não encontrar o mínimo global é visto na função Cross-Validation, que estima o parâmetro de suavização, como no exemplo esboçado na Figura 2. O mínimo global da função é $h = 69$, visto claramente na Figura 2(b), no entanto no intervalo de busca entre 6 e 72, mostrado na Figura 2(a), o mínimo encontrado é $h = 23$.

Diante deste contexto, algumas questões podem ser levadas em consideração: Como evitar o mínimo local? A Otimização por Seção Áurea é a melhor técnica na estimação do parâmetro de suavização? Outros algoritmos podem ser considerados para este fim? Encontrar o mínimo global interfere na inferência dos parâmetros? Ou seja, vale a pena o esforço para encontrar o mínimo global?

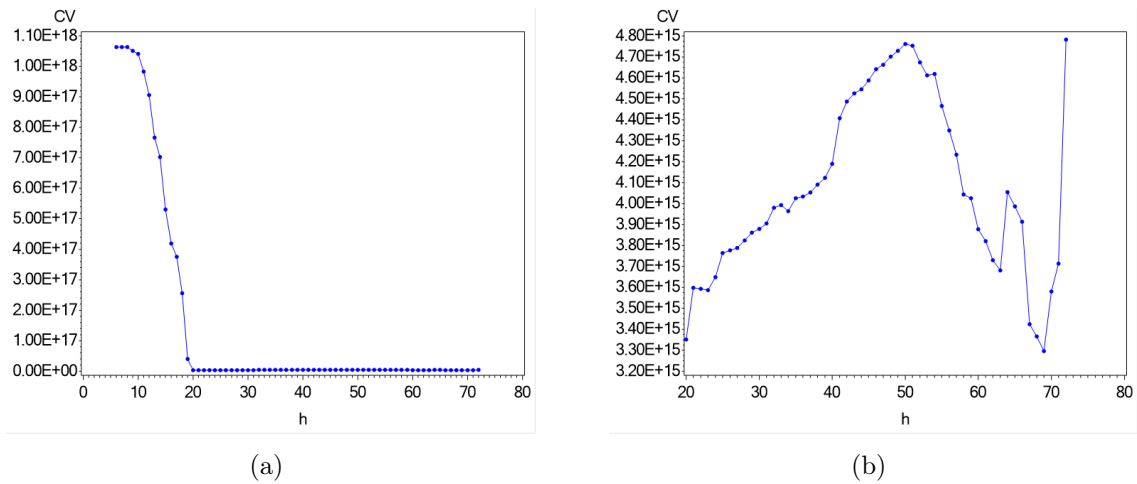


Figura 2: Função não convexa proveniente do cálculo do Cross-Validation de uma RGP com dados simulados aleatoriamente: (a) $6 \leq h \leq 72$, (b) $20 \leq h \leq 72$

Discussões sobre cada uma dessas perguntas são tratadas nesta dissertação, e, claro, na tentativa de encontrar respostas para elas. O primeiro capítulo aborda o conteúdo referente a regressão geograficamente ponderada, modelagem, estimação, parâmetro de suavização e o erro padrão. No Capítulo 2 estão alguns algoritmos de otimização, tais como, a Otimização por Seção Áurea, que ainda é o padrão “ouro” em RGP, Bisseção, Busca Aleatória, algoritmo Busca Harmônica e Algoritmo de Busca Relâmpago. O Capítulo 3 apresenta a metodologia de estudo e no Capítulo 4 está a análise dos resultados. Por fim o Capítulo 5 apresenta as conclusões, limitações do trabalho e recomendações para trabalhos futuros.

Capítulo 1

Regressão Geograficamente Ponderada

1.1 Introdução

A Regressão Geograficamente Ponderada (RGP), proposta por Fotheringham *et al.* (1996), é uma atualização da regressão linear, com acréscimo da estrutura espacial dos dados. A principal característica dessa técnica é analisar as relações das variáveis de forma específica para cada unidade do estudo. Atuando, como sugerem Fotheringham *et al.* (2002), como um microscópio para aumentar o conhecimento sobre áreas menores.

Dessa forma, a RGP é definida como uma técnica não paramétrica descritiva para explorar a não estacionariedade dos dados, com a suposição de que as regiões mais próximas possuem maior influência nas estimativas dos coeficientes da regressão do que as regiões mais afastadas. E devido a não estacionariedade em dados regionais, comum em estudos sociais, os modelos globais não são adequados, sendo a RGP uma técnica construída e eficiente para suprir essa lacuna.

Neste capítulo serão tratados os conceitos básicos da técnica de Regressão Geograficamente Ponderada, mostrando a equação do modelo, os estimadores dos parâmetros, função *Kernel*, o parâmetro de suavização, a técnica para estimá-lo e o erro padrão das estimativas.

1.2 O modelo de Regressão Geograficamente Ponderada

Como o modelo RGP é uma extensão do modelo do linear, como da Equação (1), então podemos escrevê-lo conforme a Equação (1.1).

$$y_i = \beta_0(u_i, v_i) + \sum_k \beta_k(u_i, v_i)x_{ik} + \varepsilon_i, \quad (1.1)$$

em que (u_i, v_i) corresponde às coordenadas do i -ésimo ponto no espaço e $\beta_k(u_i, v_i)$ é o resultado da função contínua $\beta_k(u, v)$ no ponto i , isto é, encontra-se uma superfície contínua dos valores dos parâmetros que refletem a variabilidade espacial (Fotheringham *et al.*, 2002). Por ser uma extensão do modelo linear, o modelo da Equação (1) é um caso especial da Equação (1.1) quando os parâmetros são espacialmente invariante. O termo do erro ε_i continua seguindo as mesmas propriedades, no entanto, os problemas de autocorrelação e heterocedasticidade são menores, por conta da possibilidade da variação espacial para os parâmetros.

Na RGP, uma observação j é ponderada de acordo com sua proximidade à localidade i , sendo que esse peso não se mantém constante para determinada observação, e sim depende da localidade i com que está sendo ponderada. A forma matricial da Equação (1.1) é dada por

$$\mathbf{y} = (\boldsymbol{\beta} \otimes \mathbf{X})\mathbf{1} + \boldsymbol{\varepsilon}, \quad (1.2)$$

em que \otimes é o operador multiplicador lógico, que denota a multiplicação elemento a elemento, e faz com que cada elemento de $\boldsymbol{\beta}$ seja multiplicado pelo correspondente de \mathbf{X} . Se existem n dados e k variáveis explicativas, ambos $\boldsymbol{\beta}$ e \mathbf{X} terão dimensões $n \times (k + 1)$ e $\mathbf{1}$ é um vetor $(k + 1) \times 1$ de 1's. A matriz $\boldsymbol{\beta}$ consiste agora de n parâmetros locais e tem a estrutura da Equação (1.3).

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0(u_1, v_1) & \beta_1(u_1, v_1) & \dots & \beta_k(u_1, v_1) \\ \beta_0(u_2, v_2) & \beta_1(u_2, v_2) & \dots & \beta_k(u_2, v_2) \\ \vdots & \vdots & \ddots & \vdots \\ \beta_0(u_n, v_n) & \beta_1(u_n, v_n) & \dots & \beta_k(u_n, v_n) \end{bmatrix}. \quad (1.3)$$

Matematicamente, $\beta_k(u_i, v_i)$ é estimado na forma matricial por:

$$\hat{\beta}(u_i, v_i) = (\mathbf{X}^T \mathbf{W}(u_i, v_i) \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W}(u_i, v_i) \mathbf{y} \quad (1.4)$$

em que $\hat{\beta}$ representa uma estimativa para β , e $\mathbf{W}(u_i, v_i)$ é uma matriz $n \times n$ com elementos fora da diagonal iguais a zero e os da diagonal representando o peso geográfico de cada observação ao ponto i . Resumidamente, e chamando (u_i, v_i) por (i) os parâmetros em cada linha da matriz da Equação (1.3) são estimados por:

$$\hat{\beta}(i) = (\mathbf{X}^T \mathbf{W}(i) \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W}(i) \mathbf{y}, \quad (1.5)$$

em que i representa a linha da matriz da Equação (1.3) e $\mathbf{W}(i)$ é uma matriz diagonal de pesos espaciais $n \times n$ da forma:

$$\mathbf{W}(i) = \begin{bmatrix} w_{i1} & 0 & \dots & 0 \\ 0 & w_{i2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_{in} \end{bmatrix}, \quad (1.6)$$

em que w_{in} é o peso dado ao ponto n na calibração do modelo para o ponto i .

O estimador da Equação (1.5) é um estimador de mínimos quadrados ponderados, no entanto não utiliza uma matriz pesos constantes. Os pesos em RGP, os valores da matriz de ponderação \mathbf{W} , são calculados para cada localidade i . Dessa forma, cada localidade recebe um peso diferente na estimação em i , ou seja, é feita uma calibragem para cada ponto de interesse. Neste sentido, a ideia é que os pesos sejam uma medida de proximidade da observação ao ponto de estimação i .

Portanto o ponto chave dessa técnica é a definição do “círculo de inclusão” de observações ao redor do ponto i , ou mais genericamente, da estrutura espacial. O círculo especificado tem raio de tamanho r , no entanto qual r deve-se considerar? Se r for muito grande, então quase todos os dados serão incluídos na estimação de $\beta_k(u_i, v_i)$, fazendo com que as estimativas se tornem bem próximas da obtida pelo modelo global (Equação (1)). Se r for muito pequeno, poucas observações serão incluídas na calibração, resultando em estimativas $\beta_k(u_i, v_i)$ com grandes erros padrões. Encontrar o melhor tamanho de r é, então, extremamente importante para achar o melhor ajuste da RGP.

Todavia a consideração da característica do peso também é relevante no ajuste, pois ele pode ser feito de forma discreta ou contínua, como comentam Brunson *et al.* (1998). No caso discreto, para realizar a calibração alguns pontos são excluídos segundo algum critério, podendo-se utilizar por exemplo um círculo de inclusão com raio r , ou seja, para uma dada localidade i , o peso w_{ik} dada à localidade k pode ser:

$$w_{ik} = \begin{cases} 1, & \text{se } d_{ik} < r, \\ 0, & \text{caso contrário.} \end{cases} \quad (1.7)$$

em que d_{ik} é a distância entre i e k . Ou outra possibilidade:

$$w_{ik} = \begin{cases} 1, & \text{se } k \text{ é um dos } N \text{ vizinhos mais próximos de } i, \\ 0, & \text{caso contrário.} \end{cases} \quad (1.8)$$

O caso contínuo considera que as localidades k mais próximas da localidade i têm mais peso na estimação do que localidades mais distantes, além disso a forma contínua pode seguir diversas distribuições. No caso Gaussiano, o peso w_{ik} pode ser representado por:

$$w_{ik} = \exp\left(\frac{-d_{ik}^2}{2h^2}\right). \quad (1.9)$$

Nesta situação, o valor do peso decai gradualmente com a distância e pode ser escrito, também:

$$w_{ik} = \begin{cases} [1 - (d_{ik}/h)^2]^2, & \text{se } d_{ik} < h, \\ 0, & \text{caso contrário.} \end{cases} \quad (1.10)$$

Essas funções, como da Equação (1.10), são conhecidas como “funções *Kernel*” ou *Kernels* e são denotadas pela letra K tal como: $w_{ik} = K(d_{ik})$. Observe que agora é o h que se torna o responsável pelo “círculo de inclusão”, como o raio r discutido anteriormente, e mais do que isso, o h também defini o grau de influência dos pesos de cada observação. O h possui também a vantagem do grau do peso decair com a distância até atingir o valor zero.

O problema, então, passa-se a ser estimar a constante h , algumas vezes referenciada como *Kernel bandwidth* ou parâmetro de suavização, que funciona, também, como um fator de variabilidade da curva dos pesos, tal como apresentado na Figura 1.1.

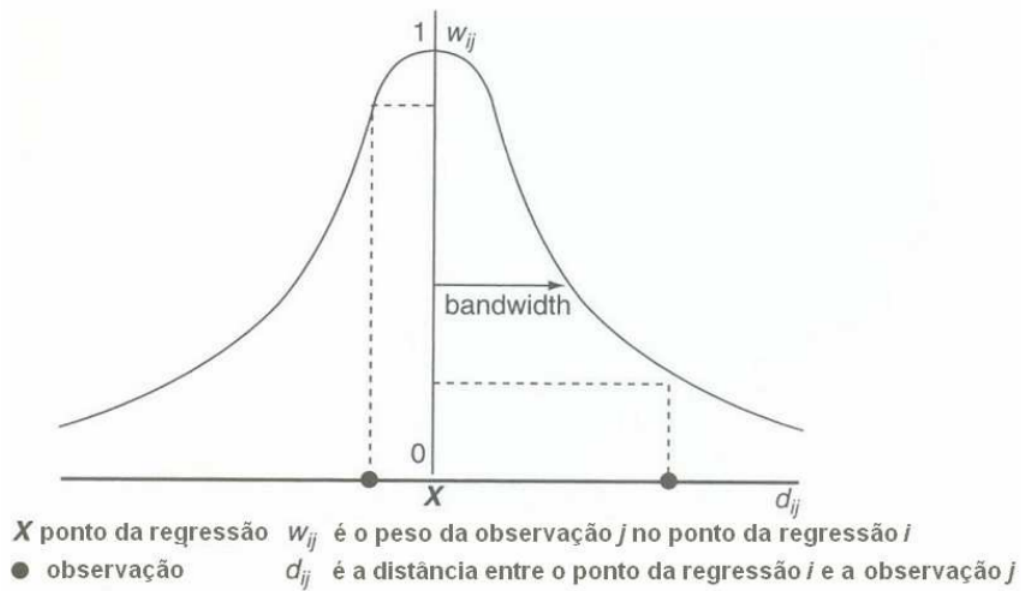


Figura 1.1: Parâmetro de Suavização.
 Fonte: (Fotheringham *et al.*, 2002)

1.3 Estimação do Parâmetro de suavização

Sabe-se que os resultados da RGP são relativamente indiferentes à escolha da função *Kernel*, mas são altamente sensíveis ao parâmetro de suavização de uma função *Kernel* utilizada (Fotheringham *et al.*, 2002). Portanto, a definição do parâmetro de suavização ótimo é um ponto crucial para o ajuste de um bom modelo RGP. E no caso mais geral, um parâmetro de suavização constante para todos os pontos se mostra eficiente se os pontos são igualmente espaçados, como está representado na região mais a esquerda da Figura 1.2.

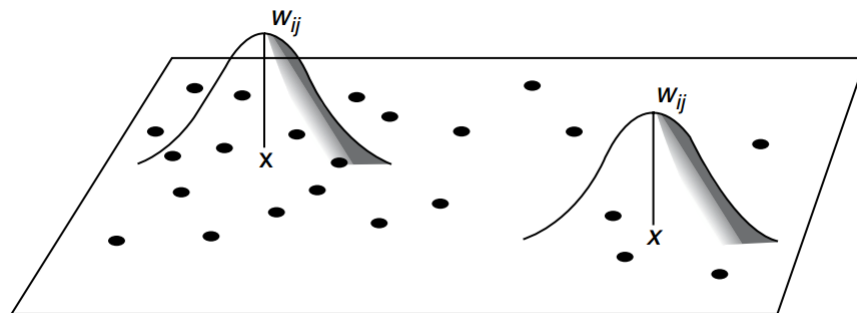


Figura 1.2: RGP com parâmetro de suavização w_{ij} ; ● pontos dos dados, X pontos de regressão.
 Fonte: (Fotheringham *et al.*, 2002)

No caso em que os dados não são igualmente espaçados, que é o caso quando olhado toda a Figura 1.2 (especialmente dispersos ou quando as áreas têm tamanhos diferentes), um parâmetro de suavização constante poderia se mostrar adequado para algumas localidades, no entanto, isso pode não ocorrer em outras localidades. Isso porque, nesse último, os parâmetros estimados poderiam ter grandes erros padrões devido a poucos pontos utilizados na calibração, ou em casos extremos, a estimação não seria possível devido à falta de variabilidade. Assim, para reduzir esses problemas, é possível utilizar um parâmetro de suavização variável (Figura 1.3), o que permitiria um grande parâmetro de suavização, onde os dados são dispersos, e um pequeno parâmetro de suavização, onde os dados são mais abundantes.

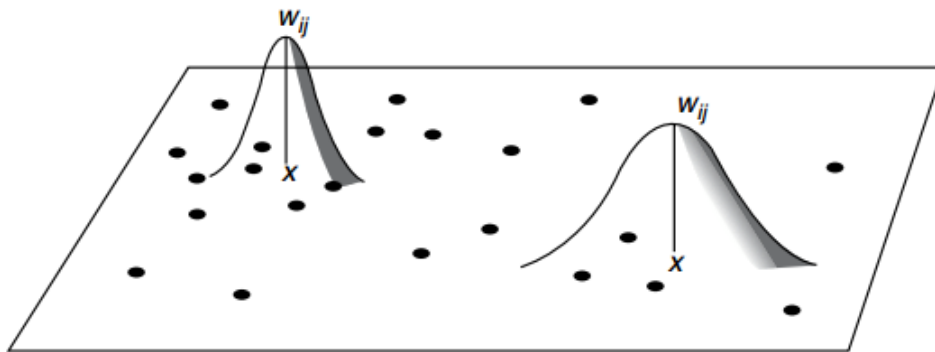


Figura 1.3: RGP com parâmetro de suavização w_{ij} variável; • pontos dos dados, X pontos de regressão.

Fonte: (Fotheringham *et al.*, 2002)

Uma solução para determinar o parâmetro de suavização é a Validação Cruzada (do inglês, *Cross-Validation* - CV), que foi sugerida por Cleveland (1979), para a regressão local da forma:

$$CV = \sum_{i=1}^n [y_i - \hat{y}_{\neq i}(b)]^2, \quad (1.11)$$

em que $\hat{y}_{\neq i}(b)$ é o valor ajustado para o ponto y_i , omitindo-se a própria observação i desse ajuste. Esta abordagem tem a propriedade desejável de contabilizar o “efeito ao redor”, já que quando b se torna o menor possível, o modelo é calibrado apenas em amostras perto de i e não do i em si. Portanto o valor que minimiza a Equação (1.11) é o parâmetro de suavização ótimo do método de validação cruzada.

Usualmente o parâmetro de suavização é estimado através do algoritmo de Otimização por seção áurea (*Golden Section Search*), o qual utiliza as premissas do algoritmo de biseção, no entanto, utiliza a seção áurea que diminui o esforço computacional, sendo a seção

áurea baseada no valor da Razão Áurea ($\phi = 1,61803$). Portanto, substituindo a equação da Validação Cruzada na função $f(\cdot)$ da Otimização por seção áurea, com o parâmetro de suavização o argumento a ser otimizado, obtêm-se o estimador deste parâmetro. Mais detalhes do algoritmo de Otimização por Seção Áurea é tratado na Seção 2.3.

1.4 Erro Padrão Local

Assim como nos modelos globais, calcular os erros padrões das estimativas locais é um passo essencial para verificar a variabilidade e a validade estatística dos estimadores dos parâmetros da regressão local. Considere que o estimador para as estimativas locais possa ser reescrito conforme a Equação (1.12).

$$\hat{\boldsymbol{\beta}}(u_i, v_i) = \mathbf{C}\mathbf{y}, \quad (1.12)$$

com

$$\mathbf{C} = (\mathbf{X}^T \mathbf{W}(u_i, v_i) \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W}(u_i, v_i). \quad (1.13)$$

A variância das estimativas dos parâmetros é dada por:

$$\text{Var}[\hat{\boldsymbol{\beta}}(u_i, v_i)] = \mathbf{C}\mathbf{C}^T \sigma^2, \quad (1.14)$$

em que σ^2 é a soma de quadrados de resíduos normalizado para regressão local, definido como

$$\sigma^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 / (n - 2\nu_1 + \nu_2), \quad (1.15)$$

em que $\nu_1 = \text{tr}(\mathbf{S})$ e $\nu_2 = \text{tr}(\mathbf{S}^T \mathbf{S})$. Cada linha da matriz \mathbf{S} , \mathbf{r}_i , por sua vez, é dada por (Fotheringham *et al.*, 2002):

$$\mathbf{r}_i = \mathbf{X}_i (\mathbf{X}^T \mathbf{W}(u_i, v_i) \mathbf{X})^{-1} \mathbf{X}_i^T \mathbf{W}(u_i, v_i). \quad (1.16)$$

O termo $n - 2\nu_1 + \nu_2$ é considerado o efetivo grau de liberdade do resíduo. O termo $2\nu_1 - \nu_2$ é equivalente ao número de parâmetros em um modelo de regressão linear global, e pode ser denominado o efetivo número de parâmetros em um modelo local RGP. Fotheringham *et al.* (2002) comentam que, devido ao traço de \mathbf{S} e $\mathbf{S}^T \mathbf{S}$ serem geralmente similares, o efetivo número de parâmetros em uma regressão local pode ser aproximado por ν_1 , o

que permite um ganho computacional por não ser necessário calcular $tr(\mathbf{S}^T \mathbf{S})$. Assim, os desvios padrões são obtidos de (1.14) por:

$$\mathbf{SE}(\hat{\beta}_i) = \sqrt{Var(\hat{\beta}_i)}. \quad (1.17)$$

em que $\hat{\beta}_i$ é uma notação reduzida para $\hat{\beta}(u_i, v_i)$. De posse dos desvios padrões é possível realizar o teste t , não no sentido formal, mas apenas de forma exploratória para destacar as partes do mapa onde os relacionamentos têm mais ou menos intensidade.

Para realizar o teste t , Silva e Fotheringham (2016) propuseram uma correção para o teste de significância das estimativas do modelo RGP. De maneira que baseia-se na relação entre o número efetivo das estimativas dos parâmetros no modelo local e o número de parâmetros no modelo global. Incluindo, então, o número de testes múltiplos e o parâmetro de suavização ótimo no modelo RGP. Dessa maneira, foi sugerido o seguinte α para o teste múltiplo:

$$\alpha = \frac{\xi_m}{\frac{p_e}{p}}, \quad (1.18)$$

em que α é o erro tipo I ou erro de significância, m é o número de testes, p_e é o número efetivo de parâmetros estimados, ou seja, $p_e = 2tr(S) - tr(S'S)$, p é o número de parâmetros do modelo e ξ_m é definido por:

$$\xi_m = 1 - (1 - \alpha)[1 - \alpha(1 - D^2)]^{m-1}, \quad (1.19)$$

sendo $D = \frac{p}{p_e}$.

Visto isso, deve-se refletir a respeito do erro padrão no modelo RGP, pois, como usual, ele é o responsável pelas inferências em relação ao modelo. Tal reflexão refere-se ao fato do erro padrão depender da matriz de pesos \mathbf{W} , Equações (1.13) e (1.14), e que por sua vez possui forte dependência do parâmetro de suavização, onde vê-se, portanto, a crucial importância desse parâmetro em toda a teoria da RGP.

Devido a significativa relevância do parâmetro de suavização na modelagem, a sua estimação deve ser feita em um processo minucioso e rigoroso, a fim de obter-se o melhor valor possível. E neste momento que torna-se essencial que o algoritmos de otimização responsáveis por esta tarefa sejam certos, o que não ocorre sempre com o algoritmo de Otimização por seção áurea, que pode encontrar um ótimo local na busca, podendo assim

não encontrar o modelo com melhor ajuste.

1.5 Regressão Poisson Geograficamente Ponderada

O modelo de Regressão Poisson Geograficamente ponderado (RPGP) é uma extensão do modelo RGP para tratar de casos os quais possuam a variável resposta como um processo de contagem. Os coeficientes de RPGP podem ser estimados através da calibração de um modelo de Regressão de Poisson onde a probabilidade é ponderada geograficamente, sendo os pesos uma função Kernel centrada em um vetor de coordenadas (u_i, v_i) (Nakaya *et al.*, 2005).

Para estimar os coeficientes faz-se inicialmente a função de verossimilhança da distribuição de Poisson a partir da variável resposta:

$$L = (\beta) = \prod_{i=1}^n \frac{\exp(-\mu(x_i, \beta)) (\mu(x_i, \beta))^{y_i}}{y_i!}, \quad (1.20)$$

com $\mu(x_i, \beta) = \exp(x_i', \beta)$. Maximiza-se, então, a log-verossimilhança da Equação (1.20):

$$\ln L(\beta) = - \sum_{i=1}^n \exp(x_i, \beta) + \sum_{i=1}^n y_i(x_i, \beta) - \sum_{i=1}^n \ln y_i!. \quad (1.21)$$

De acordo com Fotheringham *et al.* (2002) as observações em RPGP são ponderadas de acordo com a proximidade do i -ésimo local. Pode ser expressado então:

$$\ln * L(\beta(u_i, v_i)) = \left(- \sum_{i=1}^n \exp(x_i, \beta(u_i, v_i)) + \sum_{i=1}^n y_i(x_i, \beta(u_i, v_i)) - \sum_{i=1}^n \ln y_i! \right) w_{ij}(u_i, v_i). \quad (1.22)$$

Fazendo a derivada parcial da Equação (1.22) em relação aos parâmetros $\beta'(u_i, v_i)$ e igualando a zero:

$$\frac{\ln * L[\beta(u_1, v_i)]}{\beta(u_i, v_i)} = \left[- \sum_{i=1}^n x_i \exp(x_i \beta(u_i, v_i)) + \sum_{i=1}^n x_i y_i \right] w_{ij}(u_i, v_i) = 0. \quad (1.23)$$

Para maximizar a função da Equação (1.23) utiliza-se os Mínimos Quadrados Iterativamente Ponderados. Portanto tem-se:

$$\beta^{(l+1)}(u_i, v_i) = (X'W(u_i, v_i)A(u_i, v_i)^{(l)}X)^{-1}X'W(u_i, v_i)A(u_i, v_i)^{(l)}z(u_i, v_i)^{(l)}, \quad (1.24)$$

em que $(l + 1)$ indica o número de iterações. $W(u_i, v_i)$ é a matriz de pesos, $A(u_i, v_i)$ representa a matriz da variância dos pesos e é definida por:

$$A(u_i, v_i)^{(l)} = \text{diag}[\hat{y}_1\beta^{(l)}(u_i, v_i), \hat{y}_1\beta^{(l)}(u_i, v_i), \dots, \hat{y}_n\beta^{(l)}(u_i, v_i)], \quad (1.25)$$

e $z(u_i, v_i)^{(l)}$ é o vetor da variável dependente ajustada, definida como:

$$z(u_i, v_i)^{(l)} = [z_1(u_i, v_i)^{(l)}, z_2(u_i, v_i)^{(l)}, \dots, z_n(u_i, v_i)^{(l)}]. \quad (1.26)$$

1.6 Extensões da Regressão Geograficamente Ponderada

Além dos modelos propostos por Fotheringham *et al.* (1996) e Nakaya *et al.* (2005), conforme Fotheringham *et al.* (2002) algumas extensões também já foram desenvolvidas, como a abordagem bayesiana proposta por LeSage (2004), que utiliza técnica de Monte Carlo via Cadeia de Markov. Repara-se, porém, que essa abordagem ainda envolve normalidade aos dados das variáveis respostas. Seguindo outros caminhos têm-se os trabalhos de Atkinson *et al.* (2003), Silva e Rodrigues (2014) e Lima (2015), que apresentam o modelo de Regressão Geograficamente Ponderada para distribuições Logística, Binomial Negativa e Beta, respectivamente. Têm-se também outros trabalhos que estendem o modelo RGP, como é o caso de Yang (2014) que desenvolve um estudo do modelo RGP com o parâmetro de suavização flexível. Já Huang *et al.* (2010) trabalha o conceito de estudo geográfico-temporal, incluindo a não estacionariedade temporal no modelo RGP, todavia esse assunto também é tratado em Fotheringham *et al.* (2015).

Contudo, a mesma dependência que o modelo RGP tem do parâmetro de suavização em toda sua teoria, encontra-se também nessas extensões citadas acima, pelo fato de serem adaptações da RGP, tendo similarmente nas estimações dos parâmetros e do erro padrão o uso da matriz de pesos \mathbf{W} . Portanto mudanças na forma de estimação do parâmetro de suavização podem implicar em alterações nos trabalhos que utilizam os modelos citados nessa seção, além do modelo RGP proposto inicialmente por Fotheringham *et al.* (1996), e ainda, sabendo que todos os trabalhos na área utilizam a Otimização por seção áurea, um estudo sobre este algoritmo é adequado tanto quanto o estudo da estimação do parâmetro de suavização em si, já que convencionalmente esses dois assuntos estão intimamente ligados

no contexto da RGP. Por isso, no Capítulo 2 são abordados algoritmos de otimização, incluindo a Otimização por seção áurea e algoritmos baseados em processos aleatórios, com intuito de procurar alternativas para encontrar o parâmetro de suavização ótimo global.

Capítulo 2

Minimização de funções

2.1 Introdução

Para encontrar o mínimo (ou máximo) de funções muitas vezes os algoritmos computacionais são utilizados para esse fim. Atualmente, encontrar mínimos computacionalmente é amplamente difundido, sendo responsável por resolver problemas em diversas áreas do conhecimento. No entanto, encontrar o mínimo global da função não é uma tarefa fácil, visto que em alguns algoritmos param nos mínimos locais, devido a natureza da função ou dos intervalos de busca.

Encontrar o mínimo global é sempre o principal objetivo dos algoritmos de otimização, por isso, muitos deles foram desenvolvidos utilizando as mais variadas estratégias, se inspirando muitas vezes em fenômenos e comportamentos da natureza. Com o foco nos algoritmos, neste capítulo apresentar-se-á métodos de busca de mínimo, através de técnicas computacionais: Busca com intervalos iguais (Bisseção), Otimização por Seção Áurea ou *Golden Section Search* e outros métodos de otimização meta-heurística que já foram desenvolvidos.

2.2 Método de busca com intervalos iguais (Bisseção)

Quando se tem uma função f que dependa de uma ou mais variáveis, constantemente busca-se encontrar o máximo ou o mínimo desta função, por variados motivos. As soluções

dos casos de minimização linear é, geralmente, obtido por procedimentos passo a passo (Greig, 1980). E utilizar algoritmos computacionais podem ser utilizados para esse fim. Atualmente, algoritmos de minimização têm sido amplamente utilizados para resolver problemas de otimização complexos nas diversas áreas da ciência, devido à sua facilidade de uso, ampla aplicabilidade, e perspectiva global (Shareef *et al.*, 2015).

Sabe-se, porém, que nos algoritmos é desejado que: sejam rápidos, tenham baixo custo computacional e utilizem pouca memória. No entanto nos casos de otimização (encontrar o máximo ou mínimo) deve incluir mais um ponto de interesse, calcular f o mínimo de vezes possíveis (Press *et al.*, 2007).

Sendo assim, existe uma estrutura fundamental para quase todo algoritmo de minimização: começa-se com um ponto inicial; determina, de acordo com regras fixas, a direção do movimento; e então move para a direção do mínimo da função (Luenberger, 1984). No entanto em uma função de otimização pode-se encontrar ou o extremo global ou extremo local. Encontrar o extremo global é uma tarefa árdua. Geralmente a escolha dos limites do intervalo de busca podem influenciar no extremo encontrado, podendo encontrar um extremo local para cada limite escolhido (Press *et al.*, 2007). Portanto, o principal problema com a minimização de funções é ter certeza que o método escolhido encontrou o mínimo global e não tenha caído em um mínimo local (Fotheringham *et al.*, 2002).

O método de busca de mínimo com intervalos iguais é um dos mais simples para este propósito. Ele possui o foco de encontrar o mínimo de uma função f dentro do intervalo $[a, b]$ (Press *et al.*, 2007). Como mostra a Figura 2.1 tem-se o intervalo definido onde assume-se que o mínimo ocorre e escolhe-se um valor ε . Calcula-se então $f_1 = f((a + b)/2 + \varepsilon/2)$ e $f_2 = f((a + b)/2 - \varepsilon/2)$. Se $f_1 < f_2$, então o intervalo onde encontra-se o mínimo da função é alterado para $[(a + b)/2 + \varepsilon/2, b]$, caso contrário o intervalo será $[a, (a + b)/2 - \varepsilon/2]$. Desta maneira reduz-se o intervalo até encontrar o mínimo. Contudo este método não é o mais eficiente, pois a escolha do valor central do intervalo $((a + b)/2)$ não é a melhor escolha.

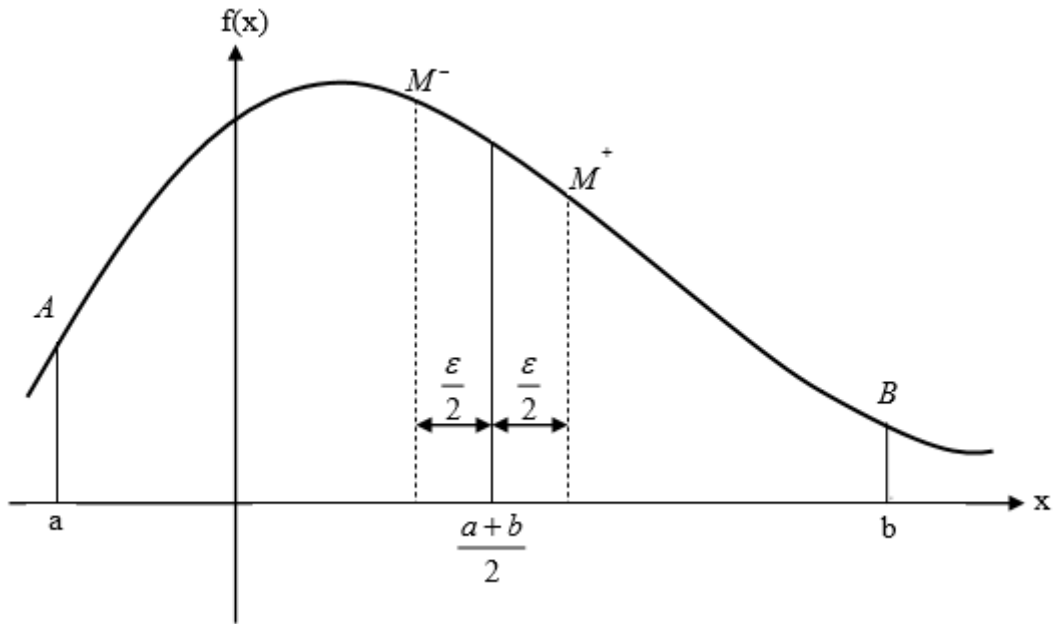


Figura 2.1: Método de busca por intervalos iguais

2.3 Otimização por seção áurea (OSA)

A escolha do valor intermediário do intervalo de busca de mínimo é intuitivo e é capaz de decrescer o intervalo até encontrar o mínimo da função. No entanto a escolha do ponto do meio do intervalo pode não ser o meio ótimo para reduzir os intervalos da maneira mais eficiente. Segundo Luenberger (1984) um método muito popular para resolver problemas de buscas lineares é o método baseado na sequência de Fibonacci descrito nessa seção.

Analogamente ao método da biseção, escolhe um novo ponto x , ou entre a e b ou entre b e c . Se $f(b) < f(x)$, então os novos pontos serão (a, x, b) , caso contrário, se $f(b) > f(x)$, então serão (b, x, c) . Em qualquer dos casos, o intervalo é reduzido até que os pontos estejam próximos o suficiente. Geralmente esse pequeno intervalo é definido como $(1 - \epsilon)b < b < (1 + \epsilon)b$, em que ϵ é o ponto flutuante de precisão do computador, com 10^{-7} para *float* ou 2×10^{-16} para *double* (Press *et al.*, 2007).

Supondo que b seja o ponto ótimo da divisão do intervalo de busca, então w será como:

$$w = \frac{b - a}{c - a} \quad (2.1)$$

e

$$1 - w = \frac{c - b}{c - a}. \quad (2.2)$$

Supondo também que no passo seguinte o ponto x é uma fração adicional de z além de b , ou seja:

$$z = \frac{x - b}{c - a}. \quad (2.3)$$

Então a próxima divisão do intervalo terá segmentos que tenham tamanhos relativos ou a $w + z$, ou a $1 - w$. E se procurar minimizar o pior caso possível, então escolhe-se o z tal que:

$$z = 1 - 2w. \quad (2.4)$$

Com o novo ponto simétrico em relação a b no intervalo original, tem-se $|b - a| = |x - c|$. Isto implica que o ponto x localiza-se no maior dos dois segmentos. Portanto x precisa ser a fração da mesma forma de b para c como b foi de a para c , ou seja:

$$\frac{x - b}{c - b} = \frac{b - a}{c - a} \Rightarrow \frac{z}{1 - w} = w. \quad (2.5)$$

Das equações (2.4) e (2.5) tem-se a equação quadrática:

$$w^2 - 3w + 1 = 0, \quad (2.6)$$

com a raiz de interesse $w = \frac{3 - \sqrt{5}}{2} \approx 0,38197$.

Por fim, tem-se o intervalo ótimo (a, b, c) com ponto do meio b a distância fracionada $0,38197$ partindo do a , e $0,61803$ do b até o c . Estas frações são chamadas de seções áureas, que possuem ligações com a Razão Áurea ($\phi = 1,61803 \Rightarrow \frac{1}{\phi} = 0,61803$), tal que segundo Luenberger (1984) foi considerado na Grécia Antiga como o valor mais estético de dois lados adjacentes de um retângulo, além dessa razão ser encontrada em muitos elementos da natureza, inclusive no corpo humano. Sendo, portanto, chamado de Otimização por seção áurea, ou do inglês, *Golden Section Search* (Greig, 1980).

Para exemplificar, suponha o intervalo $[a, b, c]$ com $a = 0$ e $c = 1$, dessa forma, teríamos que calcular x_1 e x_2 para identificar o valor de b e assim reduzir o intervalo até um valor ϵ . Então tem-se:

$$x_1 = c - (c - a)r = 1 - 0,618 = 0,382 \quad (2.7)$$

e

$$x_2 = a + (c - a)r = 0 + 0,618 = 0,618, \quad (2.8)$$

em que r representa o fração da seção áurea. Caso $f(x_1) < f(x_2)$ o novo intervalo será $[0; 0,618]$. Sendo assim, no próximo passo os novos valores de x_1 e x_2 serão:

$$x_1 = 0,618 - 0,618 \times 0,618 = 0,236 \quad (2.9)$$

e

$$x_2 = 0 + 0,618 \times 0,618 = 0,382, \quad (2.10)$$

logo em cada iteração apenas um valor deve ser alterado, mantendo o outro, poupando assim tempo e recurso computacional. O mesmo aconteceria se $f(x_1) > f(x_2)$, apenas trocando os pontos escolhidos. Uma rotina completa da Otimização por Seção Áurea pode ser encontrada em Press *et al.* (2007), no entanto um esboço do algoritmo encontra-se na Tabela 2.1.

Tabela 2.1: Algoritmo da Otimização por Seção Áurea

$$r = (\sqrt{5} - 1)/2$$

Enquanto $|b - a| > \varepsilon$ **faça**

$$x_1 = b - (b - a) \times r$$
$$x_2 = a + (b - a) \times r$$
$$f_1 = f(x_1)$$
$$f_2 = f(x_2)$$

Se $f_1 < f_2$ **faça**

$$b = x_2$$

Se $f_1 > f_2$ **faça**

$$a = x_1$$

Fonte:(Press *et al.*, 2007)

2.4 Otimização meta-heurística

A otimização meta-heurística é considerada um subcampo da otimização estocástica, que são algoritmos que aplicam grau de aleatoriedade para encontrar soluções ótimas em problemas difíceis (Luke, 2013). Em Glover e Kochenberger (2003) meta-heurística é definida como métodos de soluções que orquestram interações entre busca locais e estratégias de alto nível capaz de escapar do mínimo local, e realizar uma busca robusta no espaço de

soluções.

Posteriormente, nestes métodos também foram incluídos estratégias para evitar o mínimo local em espaços de buscas de soluções complexas, especialmente em procedimentos que utilizam uma ou mais estruturas de vizinhança como maneira de definir movimentos possíveis para transitar de uma solução a outra (Glover e Kochenberger, 2003).

Utilizar técnicas meta-heurísticas é, geralmente, aplicado para resolver problemas com poucas informações, por exemplo, em casos que não se sabe o comportamento do problema para saber a solução ótima, com pouca informação heurística ou não é possível fazer uma busca exaustiva, por ter um espaço de solução grande demais.

E hoje sabe-se que ferramentas e mecanismos baseados em métodos meta-heurísticos mostraram ser extraordinariamente eficazes. À medida que métodos meta-heurísticos não são capazes de certificar a otimização das soluções encontradas, no entanto os procedimentos exatos (que fornecem certa certificação) muitas vezes são incapazes de encontrar soluções melhores que as encontradas pelos métodos meta-heurísticos, particularmente para problemas reais, que habitualmente possuem nível mais elevados de complexidade (Glover e Kochenberger, 2003).

Contudo, existem muitos caminhos para construir um algoritmo de busca do mínimo global. Provavelmente, o algoritmo de Busca Aleatória é o mais simples (Luke, 2013). O esboço do algoritmo pode ser visto na Tabela 2.2. Entretanto esse algoritmo pode ser adaptado, mesclando-o com outros métodos de busca, tornando-o, assim, mais eficaz.

Tabela 2.2: Algoritmo Busca Aleatória

$Best \leftarrow$ valor inicial aleatório
Repita
 $S \leftarrow$ valor aleatório
 Se $f(S) < f(Best)$ **faça**
 $Best \leftarrow S$
Até $Best$ é a solução ideal ou máximo de iteração
Retorna $Best$

Fonte: (Luke, 2013)

2.5 Algoritmo Busca Harmônica

O algoritmo Busca Harmônica (*Harmony Search* - HS) foi desenvolvido por Geem e Kim (2001) para resolver problemas de otimização através de uma técnica que imita a

improvisação dos músicos. Como a maioria dos algoritmos meta-heurísticos, este também foi desenvolvido na tentativa de encontrar um processo capaz de encontrar o mínimo global e com menos gasto de energia e tempo.

Baseado em um fenômeno artificial, esse algoritmo se apoia na estética da harmonia musical gerada pelo trabalho humano. Por definição, harmonia musical é a combinação de sons considerados agradáveis do ponto de vista estético. Sendo assim, para atingir essa harmonia os músicos buscam o melhor resultado através de treino. Semelhante a esse pensamento, o HS busca encontrar o melhor valor (mínimo global) através de iterações, através do mínimo esforço possível.

Para encontrar a melhor harmonia, portanto, em cada treino os músicos improvisam em certos pontos da música, sempre adaptando-a para obter um melhor resultado. Consequentemente, o HS inicia com um vetor de soluções dentro do espaço de busca e avalia a função com os valores iniciais. Na sequência um valor aleatório de uma distribuição Uniforme $[0, 1]$ é gerado (ε) e comparado com a taxa de consideração da Memória Harmônica (TCMH), que é previamente definida. Caso o ε seja maior que o TCMH, um valor aleatório é gerado dentro do espaço de busca definido, esse passo, portanto, no contexto da música seria uma improvisação aleatória sem considerar as notas musicais atuais. Se o ε for menor que o TCMH, duas situações podem ocorrer. Novamente, compara-se um ε , agora com a Taxa de Ajuste de Afastamento (TAA), que é uma maneira de improvisar dado o histórico harmônico. Se o ε for menor que a TAA então atualiza-se o vetor de soluções com um valor próximo do antigo, com a proximidade definida pelo próprio TAA. Por fim, se $TCHM < \varepsilon < TAA$ o vetor de soluções mantêm-se com o mesmo valor. Esse passo se repete para cada posição do vetor. E finaliza o algoritmo quando o critério de parada é satisfeito. Detalhes desse algoritmo encontra-se na Tabela 2.3.

Tabela 2.3: Algoritmo Busca Harmônica

```

 $X_S \leftarrow$  vetor inicial aleatório
Enquanto critério de parada não satisfeito faça
  Para  $d = 1 : D$  faça //  $D =$  tamanho do vetor
    Se  $\varepsilon_1 < \text{TCHM}$  faça // consideração da memória
       $x_{new}(d) = x_{old}(d)$ 
      Se  $\varepsilon_1 < \text{TAA}$  faça // ajuste aproximado
         $x_{new}(d) = x_{old}(d) + bw(2\varepsilon_2 - 1)$  //  $bw < \text{TAA}$ 
      fimse
    else // seleção aleatória
       $x_{new}(d) = x_{min}(d) + \varepsilon_3 \times [x_{max}(d) - x_{min}(d)]$ 
    fimse
  fimpara
  Se  $f(X_{new}) < f(X_S)$  faça  $X_S = X_{new}$ 
fimenquanto
Retorna  $X_S$ 

```

Fonte: (Wang e Guo, 2013)

2.6 Algoritmo de Busca Relâmpago

Além de fascinante o relâmpago possui uma natureza probabilística que o torna objeto de estudo em muitos trabalhos (Dul'zon *et al.*, 1999). Inspirado no comportamento natural do relâmpago, Shareef *et al.* (2015) propuseram o Algoritmo de Busca Relâmpago (do inglês, *Lightning Search Algorithm - LSA*) para encontrar mínimos globais de funções. Recentemente, Islam *et al.* (2016) desenvolveram uma extensão do LSA, que foi definido como uma variante binária desse algoritmo: BLSA. Esse algoritmo foi desenvolvido para cobrir problemas com espaço de busca binário. No entanto, por conta do foco deste trabalho, abordar-se-á apenas o algoritmo LSA.

O algoritmo de otimização referido é uma generalização da hipótese apresentada em Berkopec (2012) como um mecanismo de propagação de etapa líder. Que considera o envolvimento de partículas rápidas conhecidas como projéteis na formação de uma estrutura em árvore binária com uma etapa líder.

2.6.1 Projéteis e propagação da etapa líder

Átomos de hidrogênio, nitrogênio e oxigênio podem ser encontrados perto de regiões com nuvens carregadas com eletricidade. Durante o congelamento intenso de moléculas de água dentro de uma nuvem de tempestade, uma parte das moléculas de água que não

conseguem encaixar-se na estrutura de gelo são espremidas para fora da formação de gelo em velocidades intensas. Os átomos de oxigênio e o hidrogênio são separados durante o processo e ejetados numa direção aleatória como projéteis. Depois de ser expulso da célula de trovão, o projétil viaja através da atmosfera e fornece um caminho de ionização inicial ou canal através de colisão e transição para a etapa líder.

No LSA, cada projétil, das células dos trovões assumem a criação de uma etapa líder e um canal. Em outras palavras, o projétil representa o tamanho inicial da população. Os projéteis sugerem soluções aleatórias para problemas a serem resolvidos pelo LSA. A solução refere-se à ponta da energia da etapa líder atual E_c .

2.6.2 Propriedades dos projéteis

Um projétil que viaja através da atmosfera sob condições normais perde a sua energia cinética durante colisões elásticas com moléculas e átomos no ar. A velocidade do projétil é dada por:

$$v_p = \left[1 - \left(1/\sqrt{1 - (v_0/c)^2} - sF_i/mc^2 \right)^{-2} \right]^{-1/2} \quad (2.11)$$

em que v_p e v_0 são as velocidades final e inicial, respectivamente, do projétil; c é a velocidade da luz; F_i é a constante da taxa de ionização; m é a massa do projétil; e s é a distância do caminho percorrido.

A Equação (2.11) mostra que a velocidade é uma função da posição da ponta líder e a massa projétil. Quando a massa é pequena ou quando o caminho percorrido é longo, o projétil tem pouco potencial para ionizar ou explorar um grande espaço. Ele só pode ionizar ou explorar o espaço nas proximidades. Portanto, as habilidades de prospecção e exploração do algoritmo pode ser controlado usando as energias relativas das etapas líderes.

Outra propriedade importante de uma etapa líder é a bifurcação, em que dois ramos simultâneos e simétricos emergem. Este fenômeno raramente ocorre devido a colisão de núcleos. Qualquer canal adicional criado durante bifurcação aumenta o número de projéteis por um e, conseqüentemente, o tamanho da população. No LSA, a bifurcação é realizada de duas formas. Em primeiro lugar, canais simétricos são criados porque a colisão de núcleos do projétil é realizada usando o número oposto como na Equação (2.12).

$$\bar{p}_i = a + b - p_i \quad (2.12)$$

em que \bar{p}_i e p_i são os projéteis oposto e original, respectivamente, num sistema unidimensional; a e b são os limites de contorno. Esta adaptação pode melhorar algumas das soluções ruins na população. Se a bifurcação não melhorar a propagação do canal na LSA, um dos canais no ponto de bifurcação é eliminado para manter o tamanho da população.

No segundo tipo de bifurcação, um canal assume uma etapa líder bem-sucedida devido à redistribuição de energia do líder mais mal sucedido depois de várias tentativas de propagação. A pior etapa líder pode ser redistribuída para definir o número máximo permitido de ensaios como o tempo de canal. Neste caso, o tamanho da população das etapas líderes não aumenta.

2.6.3 Modelagem dos projéteis e movimento da etapa líder

Três tipos de projéteis são desenvolvidos para representar os projéteis de transição que criam a primeira etapa líder da população de tamanho N , os projéteis no espaço que tentam alcançar a melhor posição de líder, e os projéteis líderes que representam a melhor posição entre os N números de etapas líderes.

2.6.3.1 Projéteis de Transição

Como mencionado anteriormente, a ponta líder é formada por conta do estágio anterior que ejeta um projétil de uma molécula de trovão em uma direção aleatória. Portanto, isso pode ser modelado como um número aleatório x^T de uma distribuição Uniforme $[a, b]$, sendo a e b os limites do espaço de solução, e x^T é o número aleatório que pode ser considerado a solução ou a energia inicial E_{sl_i} da etapa líder sl_i . Para uma população com N etapas líderes $SL = [sl_1, sl_2, sl_3, \dots, sl_N]$, são necessários N projéteis aleatórios $P^T = [p_1^T, p_2^T, p_3^T, \dots, p_N^T]$ que satisfazem as dimensões da solução.

2.6.3.2 Projéteis no espaço

Uma vez que as N etapas líderes evoluem, elas precisam se deslocar com projéteis energéticos por uma seção ionizada nas imediações da antiga etapa líder na próxima etapa *etapa* + 1. A posição do projétil no espaço $P^S = [P_1^S, P_2^S, P_3^S, \dots, P_N^S]$ na *etapa* + 1 pode ser parcialmente modelada como um número aleatório x^S gerado a partir da distribuição Exponencial com parâmetro μ . Ou seja, a posição do projétil no espaço é controlada pelo parâmetro μ . No LSA, μ_i de um específico projétil no espaço p_i^S é tomada como a distância

entre o projétil líder p^L e o projétil no espaço p_i^S sob consideração. Com esta definição, a posição de P_i^S na *etapa* + 1 pode ser escrita como:

$$p_{i_{new}}^S = p_i^S \pm \text{exprand}(\mu_i) \quad (2.13)$$

em que *exprand* é um número aleatório da distribuição Exponencial com parâmetro μ_i . Se p_i^S é negativo, então o número gerado deve ser subtraído, pois a distribuição Exponencial fornece apenas valores positivos.

No entanto, a nova posição $p_{i_{new}}^S$ não garante a propagação de uma etapa líder ou formação de canais, a menos que a energia do projétil $E_{p_i}^S$ é maior do que a etapa líder E_{sl_i} para estender o canal ou até que uma boa solução seja encontrada. Se $p_{i_{new}}^S$ fornece uma boa solução na *etapa* + 1, então a correspondente etapa líder sl_i é estendida para uma nova posição $sl_{i_{new}}$, e p_i^S é atualizado para $P_{i_{new}}^S$. Caso contrário, eles permanecem inalterados até a etapa seguinte. Se $P_{i_{new}}^S$ estende o $sl_{i_{new}}$ além do maior líder deste processo, então torna-se o projétil líder.

2.6.3.3 Projéteis líderes

Presumivelmente, a etapa líder que percorreu mais próxima para o chão e o projétil associado a ele não terá um potencial suficiente para ionizar grandes seções em frente da etapa líder. Portanto, o projétil líder pode ser modelado como um número aleatório x^L tirado a partir da distribuição Normal com os parâmetros μ e σ . Com isso, o valor gerado possibilita o projétil líder ir em qualquer direção, já que a distribuição Normal pode gerar valores negativos e positivos. Este projétil também tem uma capacidade de exploração definido pelo parâmetro σ .

No LSA, μ_L para o projétil líder p^L é tomado como P^L , e o parâmetro de escala σ_L diminui exponencialmente à medida que progride em direção à Terra ou quando ele encontra a solução ótima. Com esta definição, a posição do P^L na *etapa* + 1 pode ser escrita como:

$$p_{new}^L = p^L + \text{normrand}(\mu_L, \sigma_L) \quad (2.14)$$

em que *normrand* é o valor gerado da distribuição normal. Da mesma forma, a nova posição de liderança do projétil p_{new}^L não garante a propagação da etapa líder, a menos que a energia projétil líder $E_{p_i}^L$ seja maior do que a etapa líder E_{sl_i} para estender a uma solução satisfatória. Se p_{new}^L fornece uma boa solução na *etapa* + 1, então a correspondente

etapa líder sl_i é estendida para uma nova posição $sl_{L_{new}}$, e P^L é atualizado para p_{new}^L . Caso contrário, eles permanecem inalteradas até à fase seguinte, como no caso do projétil no espaço. O algoritmo Relâmpago está apresentado nas Tabelas 2.4, 2.5 e 2.6.

Tabela 2.4: Algoritmo Busca Relâmpago

```

Função evaluateF(X)
N=nrow(X)
dim=ncol(X)
fitness=j(N,1,0) vetor com N linhas, 1 coluna de com valores 0
Para i=1:N faça
    L=X[i,]
    fitness[i]=cv(L)
fimpara
retorna fitness
fimfunção

Função LSAMain(Limite inferior, Limite Superior, Número de observações)
low=Limite Inferior
up=Limite Superior
dim=Número de observações
N = 50
DD = dim
TT = 500
Se ncol(up)=1 faça
    UB = j(DD,1,1)*up
    LB = j(DD,1,1)*low
fimse
Se ncol(up)>1 faça
    UB = up
    LB = low
fimse
Dpoint=j(N,DD,0)
uni=j(N,1,0)
Para d=1:DD faça
    Dpoint[,d]= uniform(uni)*(UB[d]-LB[d])+LB[d]
fimpara
chtime = 0
maxchtime = 10
fitold = 1010*j(N,1,1)
xx = j(dim,1,0)
random(xx,'UNIFORM',-1,1) número aleatório uniforme de -1 a 1
direct = sinal(xx)
fitness=j(TT,1,0)
Para t=1:TT faça
    fit = evaluateF(Dpoint)
    Ec = fit
    chtime = chtime+1
Se chtime >= maxchtime faça

```

Fonte: (Shareef *et al.*, 2015) com adaptações.

Tabela 2.5: Continuação do algoritmo Busca Relâmpago

```

Ms=Ec
sort(Ms,1) ordenar Ms
sortndx(ds, Ec, 1) ds igual a Ec ordenado
Dpoint[ds[N],] = Dpoint[ds[1],]
Ec[ds[N]] = Ec[ds[1]]
chtime = 0
fimse
Ms=Ec
sort(Ms,1)
sortndx(ds, Ec, 1)
best = Ec[ds[1]]
worst = Ec[ds[N]]
Energy = 2.05 - 2*exp(-5*(TT-t)/TT)
Para d=1:DD faça
    Dpointtest = Dpoint[ds[1],]
    Dpointtest[d] = Dpointtest[d]+direct[d]*0.005*(UB[d]-LB[d])
    fvtest = evaluateF(Dpointtest)
    Se fvtest < best faça direct[d] = direct[d]
    senão direct[d] = -1*direct[d]
fimpara
Para i=1:N faça
    dist=Dpoint[i,]-Dpoint[ds[1],]
    Dpointtemp=j(DD,1,0)
    Para d=1:DD faça
        xx = j(1,1,0)
        random(xx,'NORMAL',0,Energy) número aleatório normal com  $\mu = 0$  e  $\sigma = Energy$ 
        yy = j(1,1,0)
        Se dist[d]=0 faça dist[d]=0.0000001
        senão dist[d]=dist[d]
        random(yy,'EXPONENTIAL',abs(dist[d]))
        Se Dpoint[i,]=Dpoint[ds[1],] faça
            Dpointtemp[d] = Dpoint[i,d]+direct[d]*abs(xx)
        senão faça
            if dist[d]<0 then Dpointtemp[d] = Dpoint[i,d]+yy
            else Dpointtemp[d] = Dpoint[i,d]-yy
        fimse
        Se (Dpointtemp[d]>UB[d]) | (Dpointtemp[d]<LB[d]) faça
            Dpointtemp[d] = ranuni(1)*(UB[d]-LB[d])+LB[d]
        fimse
    fimpara

```

Fonte: (Shareef *et al.*, 2015) com adaptações.

Tabela 2.6: Continuação do algoritmo Busca Relâmpago

```

fv = evaluateF(Dpointtemp')
fv=fv[,1]
diff=abs(fv-Ec[i])
Se fv < Ec[i] then faça
    Dpoint[i,] = Dpointtemp'
    Ec[i] = fv[1]
    Se ranuni(1) < 0.01 faça
        Dpointfock=j(DD,1,0)
        Para d=1:DD
            Dpointfock[d] = UB[d]+LB[d]-Dpointtemp[d]
        fimpara
        fockfit = evaluateF(Dpointfock')
        fockfit=fockfit[,1]
        Se fockfit < Ec[i] faça
            Dpoint[i,] = Dpointfock'
            Ec[i] = fockfit
        fimse
    fimse
fimse
fimse diff<0.00001 faça
    i=N
    t=TT
fimse
fimpara
Se best = worst faça t=TT
    fitness[t] = min(fit)
fimpara
fitness2=fitness[loc(fitness[,1]>0),1] localização da condição
grandmin = min(fitness[loc(fitness[,1]>0),1])
min=grandmin||Dpointtemp
retorna (min)
fimfunção

```

Fonte: (Shareef *et al.*, 2015) com adaptações.

Capítulo 3

Materiais e Métodos

3.1 Introdução

No Capítulo 1 apresentou-se o conceito geral da Regressão Geograficamente Ponderada, atentando-se a importância do parâmetro de suavização, que para ser estimado geralmente é feito por um algoritmo de otimização, do qual trata o Capítulo 2, detalhando o algoritmo mais comum utilizado na RPG, Otimização por Seção Áurea, assim como outros algoritmos que buscam encontrar o mínimo global por caminhos diferentes. Este trabalho, portanto, verificará possíveis maneiras de estimar o parâmetro de suavização, através de simulações e estudos de casos reais. Para desenvolvimento dos algoritmos e o processamento das regressões são utilizados os *softwares* SAS[®] Studio (predominantemente), R-Project versão 3.2.2, MATLAB[®] versão 9.0.0.341360 e o GWR4.

3.2 Material

Inicialmente será comparada a eficiência e rapidez dos algoritmos Otimização por Seção Áurea (OSA), Relâmpago, Harmônico e uma variação do OSA, que foram apresentados no Capítulo 2. O algoritmo OSA é frequentemente utilizado na RGP, o que o torna o algoritmo referência, todavia a escolha do algoritmo Relâmpago é feita devido ao artigo de Shareef *et al.* (2015), que apresentou o melhor desempenho entre outros algoritmos de busca mais conhecidos. No entanto, o algoritmo Harmônico não é comparado no artigo de Shareef *et al.* (2015), portanto ele foi escolhido para obter mais um parâmetro dos resultados obtidos.

Para essa comparação serão utilizadas duas funções: a primeira delas está representada

na função f_r da Equação (2), que possui um comportamento cíclico, e dentro do intervalo $9 \leq r \leq 21$ a função possui dois mínimos. A Figura 1 apresenta os mínimos local e global, e esta característica torna a finalidade dos algoritmos de buscas por mínimos globais mais difíceis.

Com esse mesmo intuito, os algoritmos também serão comparados utilizando a função f_n da Equação (3.1), que não possui comportamento cíclico e controlado, e no intervalo $[-1,4]$ possui 2 mínimos locais e um global, podendo assim fazer uma comparação distinta. Visualmente, pela Figura 3.1, sabe-se que o mínimo global da função f_n é igual 3, então espera-se que os algoritmos sejam capazes de encontrar um valor próximo.

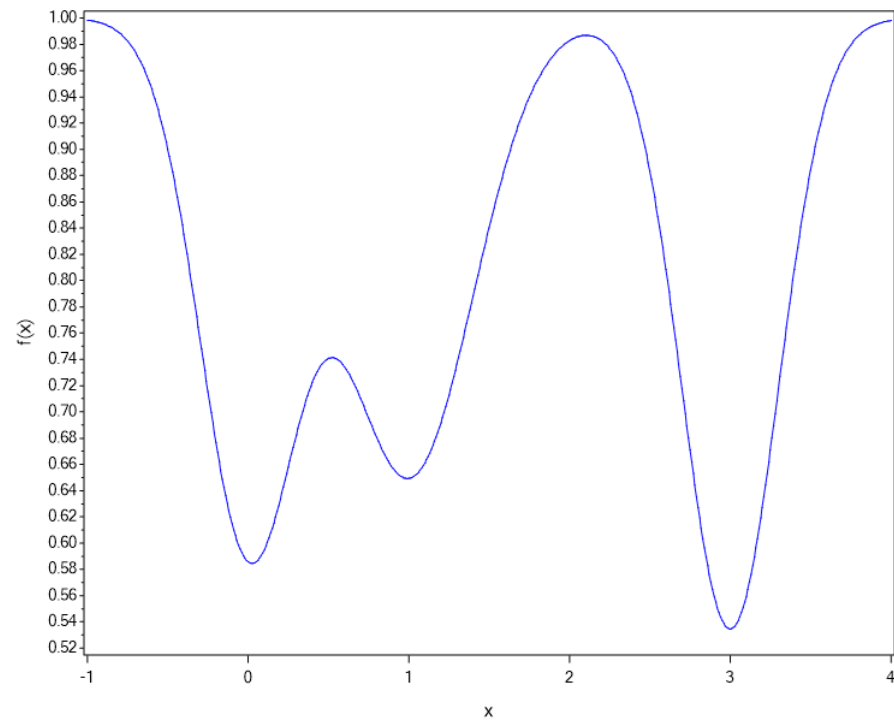


Figura 3.1: Função f_n , no intervalo $[-1;4]$.

$$f_n(x) = 1 - 0.3 \times \frac{1}{0.3\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-0}{0.3}\right)^2} + 0.35 \times \frac{1}{0.4\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-1}{0.4}\right)^2} + 0.35 \times \frac{1}{0.3\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-3}{0.3}\right)^2} \quad (3.1)$$

3.2.1 Dados simulados

O primeiro caso com dados simulados segue um padrão homogêneo, ou seja, possui um *shape* regular, como está representado na Figura 3.2. Desta maneira, foram simulados 3 conjuntos de dados de tamanhos 64 (8×8), 100 (10×10) e 1024 (32×32), todos regulares, no entanto com detalhes distintos. As coordenadas são mais espaçadas no primeiro conjunto, enquanto que os conjuntos com 100 e 1024 pontos distanciam-se em apenas 0,5 horizontal e verticalmente.

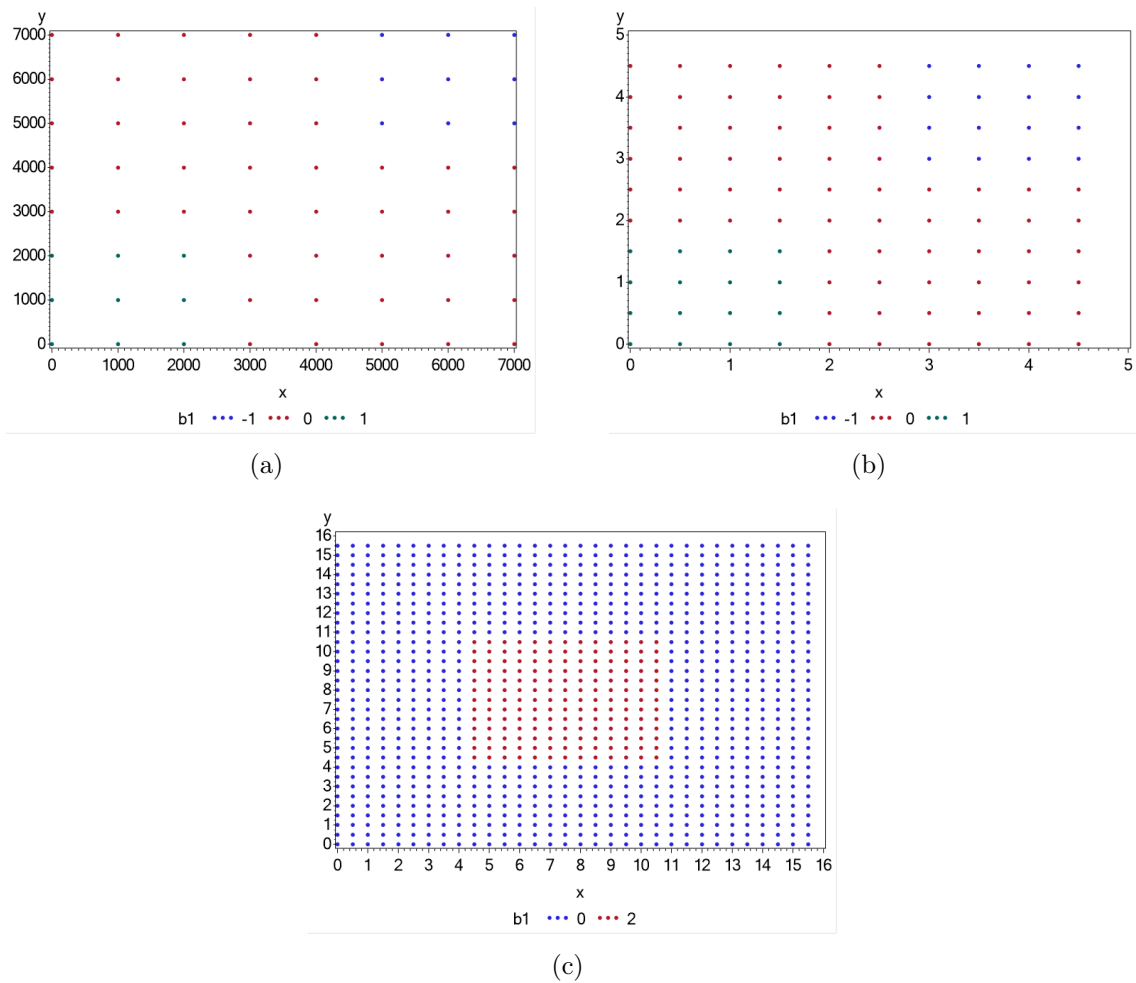


Figura 3.2: Simulação de dados com *shape* regular de tamanhos: (a) 8×8 , (b) 10×10 , (c) 32×32 .

São definidas três variáveis e um erro, com as variáveis $varx$ e $varx2$ recebendo um valor aleatório de uma distribuição Uniforme $[0, 1]$. O erro possui distribuição Normal $(0, 1)$ e $vary = b_0 + b_1 varx_1 + b_2 varx_2$, em que $b_0 = 5$, e b_1 e b_2 recebem inicialmente o valor 0. Contudo, como observa-se na Figura 3.2, algumas regiões possuem b 's distintos, que para

cada conjunto a regra é um pouco diferente, o qual estão apresentadas na Tabela 3.1. Os detalhes do código utilizado para gerar esses conjuntos encontram-se no Apêndice A.

Tabela 3.1: Regras para definição dos b 's de cada conjunto.

Conjunto	Regra	b_1	b_2
Dados 64	$x < 3000$ e $y < 3000$	1	5
Dados 64	$x > 4000$ e $y > 4000$	-1	-5
Dados 100	$x < 2$ e $y < 2$	1	5
Dados 100	$x > 2.5$ e $y > 2.5$	-1	-5
Dados 1024	$x > 4$ e $x < 11$ e $y > 4$ e $y < 11$	2	8

Como o *shape* regular não reflete os casos que são estudados na prática, simulações com coordenadas aleatórias serão também criadas, no entanto, com um comportamento desejado, o qual muitos pontos se concentram em uma pequena região e alguns poucos pontos são mais esparsos, indicado na Figura 3.3.

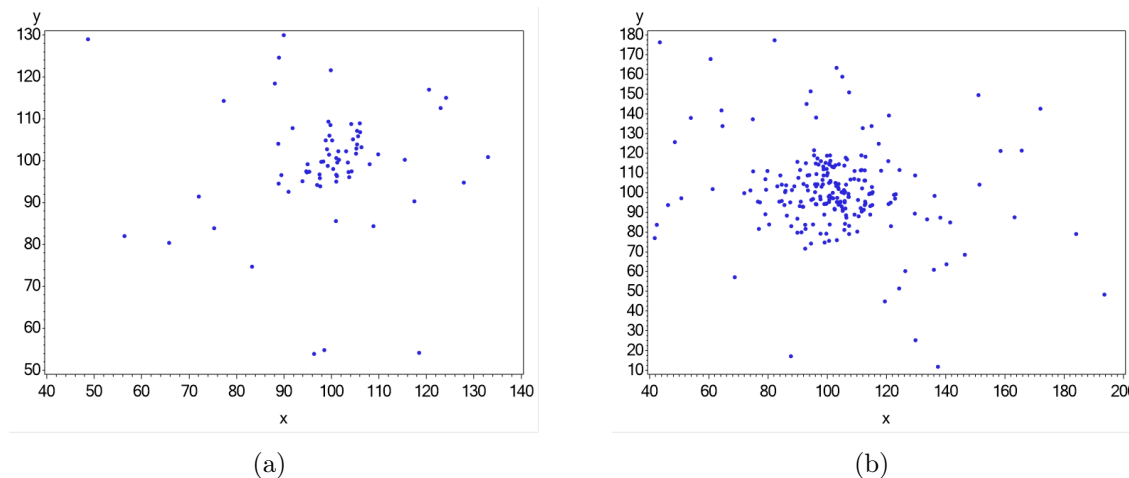


Figura 3.3: Simulação de dados com *shape* irregular de tamanhos: (a) 72, (b) 225.

Além das coordenadas aleatórias, os valores das variáveis $vary$, $varx$, e $varx_2$ também diferem da simulação do *shape* regular, pois no primeiro caso os valores foram gerados aleatoriamente sem levar em consideração a posição das coordenadas de cada observação, assim como os valores b_0 , b_1 e b_2 que foram definidos deterministicamente. Para o *shape* aleatório, leva-se em consideração a posição (x, y) de cada observação, para gerar aleatoriamente $vary$, $varx_1$, e $varx_2$, e definir b_0 , b_1 e b_2 baseado nas coordenadas. A simulação dos dois conjuntos de simulação aleatória também estão detalhados no Apêndice A.

A simulação neste caso segue com métodos semelhantes para os dois conjuntos. Diferenciando no número de observações, a construção dos valores dos b 's, que está detalhado na

Tabela 3.2, e o desvio padrão das coordenadas x e y . De maneira que 2/3 das observações possuem desvio padrão menor, $\sigma = (5, 10)$, respectivamente do primeiro e o segundo conjunto. E 1/3 possui desvio padrão um pouco maior, $\sigma = (25, 25)$. Portanto as coordenadas x e y são geradas aleatoriamente por uma distribuição Normal $(100, \sigma)$.

Tabela 3.2: Detalhes dos conjuntos de dados simulados aleatoriamente de tamanhos $n = 72$ e $n = 225$

$n = 72$	$n = 225$
$b_0 = 0,000005 \times x^3 + 0,000005 \times y^3$	$b_0 = 0,000005 \times x^3 + 0,000005 \times y^3$
$b_1 = \sqrt{x^2 \times y^2}$	$b_1 = \sqrt{\frac{x^2}{y^2}}$
$b_2 = 2 \times \left(- \left(\frac{x}{5} \right)^2 - \left(\frac{y}{5} \right)^2 + 0.13 \right)$	$b_2 = 2 \times \left(- \left(\frac{x}{5} \right)^2 - \left(\frac{y}{5} \right)^2 + 0.13 \right) / 1000$
$varx \sim Normal(b_1, 2)$	$varx \sim Normal(b_1, 2)$
$varx_2 \sim Exponencial(1/ b_2)$	$varx_2 \sim Exponencial(b_2)$
$erro = Normal(0, 5)$	$erro = Normal(0, 5)$
$vary = b_0 + b_1varx_1 + b_2varx_2 + erro$	$vary = b_0 + b_1varx_1 + b_2varx_2 + erro$

Esses *shapes* aleatórios podem ser considerados uma projeção de dados reais, pois nota-se semelhanças com os mapas reais, como os mapas de São Paulo e Pará apresentado na Figura 3.4. Nos mapas dos estados (Figuras 3.4(a) e 3.4(c)) estão representados também os centros administrativos de cada município. Mesmo que sejam os mapas que apresentam os limites dos municípios, para calcular a distância entre municípios geralmente são utilizados pontos do municípios como o centro administrativos ou o centróide. Ao retirar os limites dos municípios e manter apenas os centros administrativos, como nas Figuras 3.4(b) e 3.4(d), a nuvem de pontos torna-se muito semelhante aos *shapes* gerados aleatoriamente da Figura 3.3.

3.2.2 Casos reais

A análise continuará com dados reais, observando novamente o tempo gasto em cada algoritmo e a eficiência nos casos do parâmetro de suavização fixo e o adaptável, e também será visto o comportamento da função CV através do seu esboço. Esses dados reais são relacionados ao estado de Goiás, com o rendimento médio da população como variável resposta e o tamanho da população dos municípios como a variável explicativa. E a partir do mapa do estado serão comparados as intensidades dos estimadores da variável população do modelo RGP, para diferentes valores do parâmetro de suavização.

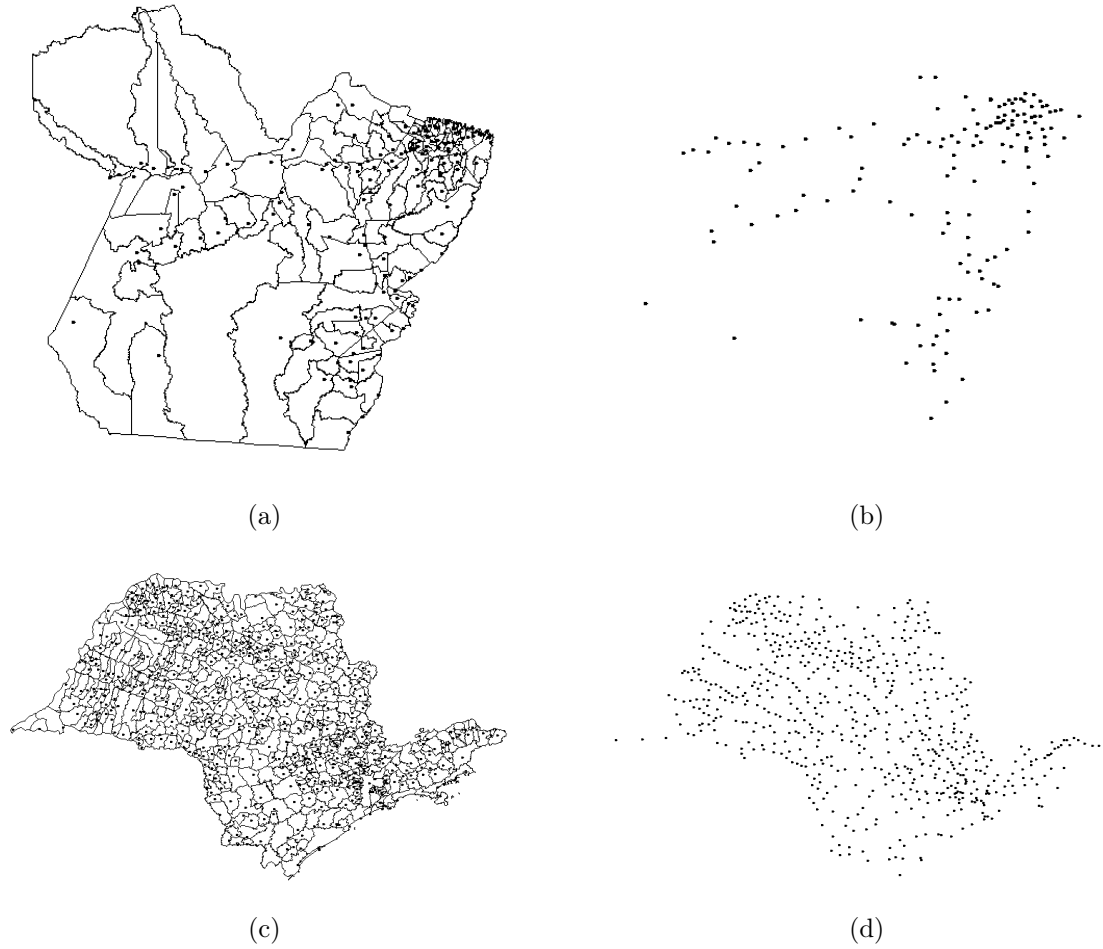


Figura 3.4: Mapa dos municípios dos estados de Pará (a) e São Paulo (b) e seus respectivos centros administrativos (c) e (d).

O segundo caso analisado será o número de acidentes ocorridos na zonas de tráfego na cidade de Fortaleza/CE, o qual será um análise diferente das demais, dado que a variável resposta neste caso é um processo de contagem. Portanto, o modelo que será ajustado para esse conjunto de dados será definido pela Regressão Poisson Geograficamente Ponderada. Estes dados foram gentilmente cedidos pelo SIATFOR - Sistema de Informações de Acidentes de Trânsito de Fortaleza.

3.3 Método

Dado que as funções f_r e f_n não são estritamente convexas, espera-se que o algoritmo OSA não seja adequado, por isso uma adaptação será proposta, de maneira que o intervalo de busca seja dividido em três partes, e cada divisão dessa seja aplicado o algoritmo OSA.

As divisões são definidas semelhantemente ao primeiro passo do algoritmo OSA, ou seja, a primeira divisão se dará no ponto $x_1 = b - (b - a) \times r$ e a segunda $x_2 = a + (b - a) \times r$. O esboço do algoritmo proposto encontra-se na Tabela 3.3.

Tabela 3.3: Algoritmo da Otimização por Seção Áurea

```

 $r = (\sqrt{5} - 1)/2$ 
 $inf = (a, (1 - r) \times b, r \times b)$ 
 $sup = ((1 - r) \times b, r \times b, b)$ 
Para  $j = 1 : 3$  faça
   $a1 = inf[j]$ 
   $b1 = sup[j]$ 
  Enquanto  $|b - a| > \varepsilon$  faça
     $x_1 = b_1 - (b_1 - a_1) \times r$ 
     $x_2 = a_1 + (b_1 - a_1) \times r$ 
     $f_1 = f(x_1)$ 
     $f_2 = f(x_2)$ 
    Se  $f_1 < f_2$  faça
       $b_1 = x_2$ 
       $X_{min} = b_1$ 
    Se  $f_1 > f_2$  faça
       $a_1 = x_1$ 
       $X_{min} = a_1$ 
    fimenquanto
     $X_{local}[j] = X_{min}$ 
  fimpara
Retorna  $\rightarrow$  Menor ( $X_{local}$ )

```

Serão feitas então comparações entre os algoritmos OSA, OSA+divisão, Relâmpago e Harmônico, e a partir da comparação inicial com as funções f_r e f_n , serão selecionados os algoritmos que encontram com maior rapidez o mínimo global, para assim retirar opções que gastam muito tempo em funções simples. Transcorrer-se-á desta maneira, pois, a função CV, que é o meio para encontrar o parâmetro de suavização, possui maior complexidade e exige algoritmos mais ágeis. Seguindo, serão calculados o parâmetro de suavização, com os tamanhos diferentes de população. Os tamanhos diferentes trarão respostas quanto a comportamento de cada algoritmo, podendo verificar se algum deles tem um aumento do tempo maior quando os dados são maiores.

Para cada conjunto comparar-se-á os algoritmos analisando o tempo gasto de cada um para encontrar mínimo da função CV, e portanto o parâmetro de suavização ótimo. Em cada caso, também serão apresentados os resultados para o parâmetro de suavização fixo

e o adaptável, que possuem comportamentos diferentes, sendo portanto, outro meio para entender qual algoritmo mais adequado para cada situação. Para todos os casos, o intervalo de busca será sempre começado do 0 e o limite superior será o inteiro imediatamente superior da maior distância observada dos pontos do conjunto de dados, para o parâmetro de suavização fixo. Para o adaptável, o intervalo de busca compreenderá de 6 vizinhos até o número máximo de observações. E para entender os comportamentos de cada caso, acrescentar-se-á gráficos que mostram o esboço da função CV nas condições estudadas.

Por fim, buscar-se-á respostas para a última pergunta levantada na Introdução, ou seja, se os resultados encontrados pelos algoritmos deste estudo alterarão as inferências encontradas anteriormente. E será feito tanto para o caso do parâmetro de suavização fixo quanto para o adaptável. Para verificar a significância dos estimadores do modelo RGP será adotado o critério do teste múltiplo com a correção definida por Silva e Fotheringham (2016). Além disso serão comparados resultados com o *software* mais utilizado para problemas de regressão geograficamente ponderada, GWR4, indicando possíveis alterações que podem ser incorporadas nesse *software*, tais como um algoritmo mais eficiente para cada problema.

Capítulo 4

Resultados

4.1 Exemplos determinísticos

4.1.1 Função 1

A Equação (2), função f_r , apresentada na Introdução possui uma característica cíclica, e ao examinar apenas o intervalo com r entre 9 e 21, constata-se uma função com dois mínimos, um mínimo local igual a $r = 16,31579$ e um mínimo global igual a $r = 10,01901$, tendo uma situação ideal para avaliar a eficiência dos algoritmos apresentados no Capítulo 2. Como já mostrado na Introdução o algoritmo Otimização por Seção Áurea (OSA) é capaz de encontrar o mínimo global, no entanto, a escolha do intervalo pode influenciar no mínimo encontrado. Por exemplo, aplicando o algoritmo no intervalo $[9;20]$ o mínimo global é encontrado, porém alterando o intervalo para $[9;21]$ o mínimo encontrado passa a ser o local. Destaca-se, contudo, a rapidez deste algoritmo, que gasta apenas 0,03 segundos para encontrar o mínimo global na busca do intervalo $[9;20]$, com um erro menor que 0,0001.

Através da Figura 4.1 é possível entender o motivo do algoritmo OSA encontrar o mínimo local, no intervalo $[9;21]$. Atenta-se na Figura 4.1(b) que na primeira etapa do algoritmo compara-se dois valores intermediários, o primeiro definido por $x_1 = b - (b - a)\frac{1}{\phi}$ e o segundo por $x_2 = a + (b - a)\frac{1}{\phi}$, sendo $\phi = 1,61803$. Para esta função x_1 assume 13,58359 e x_2 é igual a 16,41641. Portanto, pode-se observar através da Tabela 4.1 que $f_r(x_1) = 58,62218$ e $f_r(x_2) = 38,6247$. e o novo intervalo, da segunda etapa do algoritmo, será $[13,58359;21]$, excluindo assim a possibilidade de encontrar o mínimo global ($r = 10,01901$). As etapas para esse intervalo estão ilustradas na Figura 4.1(b).

Já para o intervalo $[9;20]$, apresentado na Figura 4.1(a), a primeira etapa do algoritmo OSA seleciona o intervalo $[9;15,79837]$, pois x_1 e x_2 assumem os valores 13,20163 e 15,79837, respectivamente, e conseqüentemente $f_r(x_1) = 52,63829$ e $f_r(x_2) = 54,113$, como pode ser visto na Tabela 4.1. Verifica-se que a diferença entre $f_r(x_1)$ e $f_r(x_2)$ é bem menor neste caso, e quase indistinguível ao observar a Figura 4.1(a), no entanto o ponto importante a atentar-se entre estes dois casos detalhados é que a diferença de apenas uma unidade no limite superior da busca alterou o mínimo encontrado.

Tabela 4.1: Valores da primeira etapa do algoritmo OSA aplicado na função f_r nos intervalos $[9;20]$ e $[9;21]$.

Intervalo	x_1	x_2	$f_r(x_1)$	$f_r(x_2)$
$[9, 20]$	13,20163	15,79837	52,63829	54,113
$[9, 21]$	13,58359	16,41641	58,62218	38,6247

(a)

(b)

Figura 4.1: Etapas do algoritmo OSA aplicado na função f_r nos intervalos: (a) $[9;20]$, (b) $[9;21]$.

Fica claro que em caso de uma função não estritamente convexa o algoritmo OSA não é adequado. Portanto é necessário uma alternativa em casos onde possa existir mais de um mínimo na função. E pensando na enorme rapidez com que o algoritmo OSA encontra o resultado, um passo interessante para eliminar a chance do resultado ser um mínimo local seria dividir o intervalo em seções definidas estrategicamente, e deste modo aplicar o algoritmo OSA em cada intervalo. Assim, ter-se-á um mínimo para cada intervalo, podendo então detectar os mínimos locais e o mínimo global. Claro que em uma função com milhares de mínimos locais esse processo também não seria adequado, todavia, em funções mais “suaves” é completamente aplicável, sendo um exemplo a função da Figura 1.

Seja, então, o intervalo de busca $[9,21]$ na função f_r , dividido em 3 partes, como apresentado na Figura 4.3. Em cada novo intervalo é aplicado o algoritmo OSA, desta maneira tem-se agora três mínimos para ser comparados, que estão expostos na Tabela 4.2. O mínimo da função fora encontrado ao utilizar a divisão, podendo enfim definir qual o mínimo global, ou seja, para o r igual 10,019016.

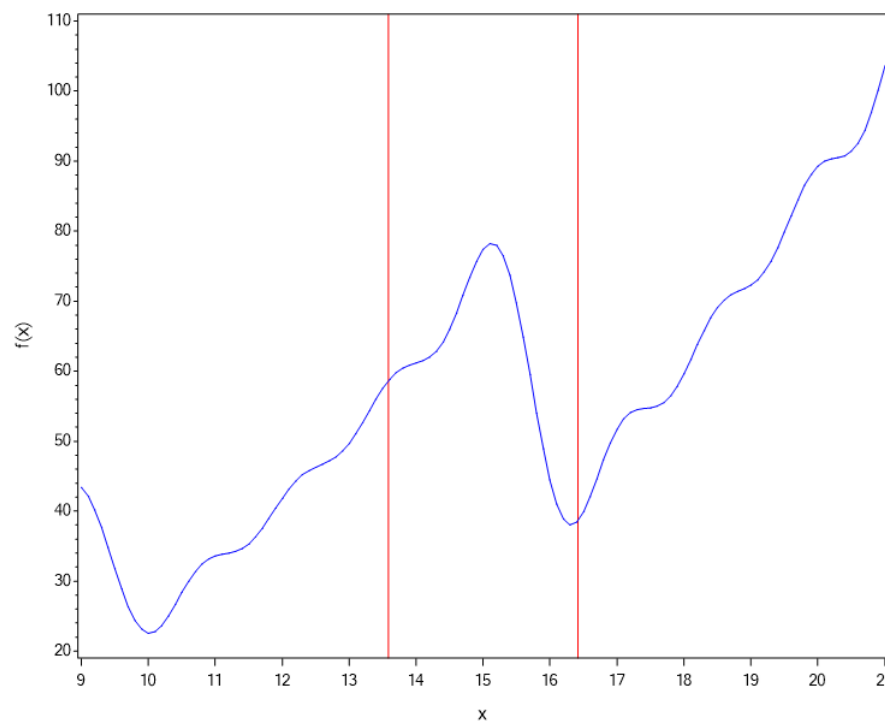


Figura 4.2: Divisão da função f_r em três partes.

Tabela 4.2: Mínimos encontrados pelo algoritmo OSA das 3 divisões da função f_r .

r	$f_r(r)$
10,019016	22,567409
13,583592	58,622175
16,416414	38,624768

Contudo, pode-se encontrar o mínimo global com os algoritmos meta-heurísticos apresentados no Capítulo 2. Com o mesmo espaço de busca, $[9;21]$, e utilizando os algoritmos Relâmpago e Harmônico na função f_r , o mínimos encontrados, respectivamente, foram $r_R = 10.019249$ e $r_H = 10.018984$. Na Tabela 4.3 são dados o mínimos encontrados por cada algoritmo citado, e também seus respectivos valores da função f_r . Exceto pelo OSA clássico, todos os algoritmos apresentam resultados muito próximos, diferenciando apenas na terceira casa decimal, ainda assim a diferença da função f_r para cada valor encontrado

é apenas na quinta casa decimal, que foi o erro máximo admitido em cada algoritmo.

Tem-se, portanto, três algoritmos eficazes, que encontram o mínimo global, no entanto o tempo de processamento de cada algoritmo não são próximos, principalmente o algoritmo Harmônico que levou um tempo muito maior que os outros algoritmos, conforme apresentado na Tabela 4.3. O algoritmo levou 11,65 segundos para encontrar o mínimo, enquanto que os outros algoritmos tiveram um tempo na casa dos centésimos de segundo.

Concentrando nos algoritmos mais rápidos, é importante notar que mesmo aplicando o algoritmo OSA nos três intervalos, a diferença de tempo entre o OSA comum e o OSA+divisão é de apenas 0,01. E o algoritmo Relâmpago em termos absolutos também tem um tempo bem próximo, diferenciando em 0,04 do OSA comum e 0,03 do OSA+divisão. No entanto em termos relativos, o algoritmo Relâmpago é 133% mais lento que o OSA comum.

Tabela 4.3: Mínimo dos algoritmos OSA, Relâmpago e Harmônico para o intervalo [9;21], tempo de processamento e erro máximo admitido.

Algoritmo	Mínimo	$f(r)$	Tempo(s)	Erro
OSA comum	16,315797	38,039611	0,03	0,00001
OSA+divisão	10,019016	22,567409	0,04	0,00001
Relâmpago	10,019249	22,567411	0,07	0,00001
Harmônico	10,018984	22,567409	11,65	0,00001

4.1.2 Função 2

A função f_r possui um comportamento cíclico, e o intervalo escolhido possui apenas um mínimo local e um mínimo global. Já a função f_n , Equação (3.1), não possui comportamento cíclico e controlado, e no intervalo [-1,4] possui 2 mínimos locais e um global, podendo assim fazer uma nova comparação do algoritmo OSA comum com o OSA+divisão, e também com os algoritmos Relâmpago e Harmônico.

Visualmente, pela Figura 3.1, sabe-se que o mínimo global da função f_n é para $x = 3$, então espera-se que os algoritmos sejam capazes de encontrar um valor próximo, novamente com o erro máximo admitido igual a 0,00001. Ao processar, então, o algoritmo OAS comum no intervalo [-1;4], o mínimo encontrado para a função f_n é $x = 0,0246363$, tendo mais uma vez encontrado um mínimo local, o que não era desejado, mas não é uma surpresa, devido a não convexidade da função.

A etapa seguinte, portanto, passa a ser o algoritmo OSA+divisão, o qual possui suas

divisões da função f_n representadas na Figura 4.3. Os três mínimos da função ficaram em divisões diferentes, o que tornará possível encontrar cada um, para comparar os resultados do algoritmo OSA de cada parte.

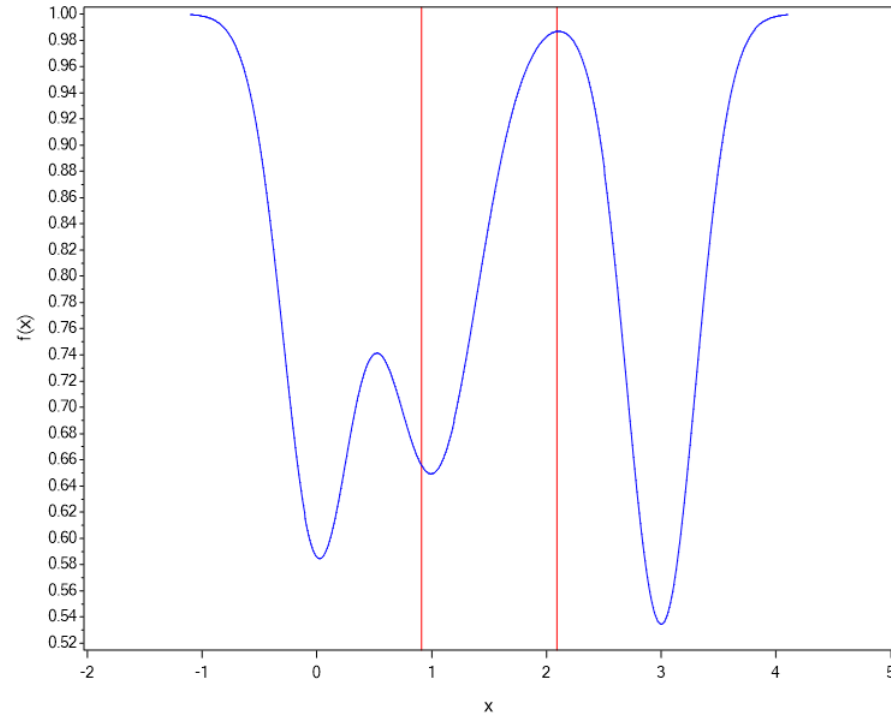


Figura 4.3: Divisão da função f_r em quatro partes iguais

O melhor resultado do algoritmo, ou seja, o menor f_n encontrado foi 0,534566 para $x = 2,9999984$, que é o mínimo global, e gastou apenas 0,07 segundos para chegar ao resultado, conforme a Tabela 4.4. Diferente da função f_r , que tem uma diferença relativa do tempo de processamento pequena entre o OSA comum e o OSA+divisão, a diferença relativa do tempo gasto desses algoritmos na função f_n é bem maior, já que o OSA comum gastou apenas 0,03 segundos. Mesmo assim o tempo ainda é ainda muito baixo, e o OSA+divisão tem a vantagem de encontrar o menor valor possível. O algoritmo Relâmpago também possui um tempo maior de processamento, 0,14 segundos, contudo ele também encontra o mínimo global, $x = 3,0032915$. Enquanto que o algoritmo Harmônico leva muito mais tempo para achar o mínimo, 48,41 segundos. O valor da f_n encontrado pelo Harmônico é menor que o Relâmpago, no entanto apenas a partir da quarta casa decimal, portanto considerar-se-á apenas o algoritmo OSA+divisão e o Relâmpago para o estudo realizado com dados simulados, devido principalmente a enorme diferença que o algoritmo Harmônico gasta a mais para encontrar o mínimo.

Tabela 4.4: Mínimo dos algoritmos OSA, Relâmpago e Harmônico para a função f_n , tempo de processamento e erro máximo admitido.

Algoritmo	Mínimo	$f(r)$	Tempo(s)	Erro
OSA comum	0,0246363	0,5845442	0,03	0,00001
OSA+divisão	2,9999984	0,534566	0,04	0,00001
Relâmpago	3,0032915	0,5345891	0,14	0,00001
Harmônico	2,9999957	0,534566	48,41	0,00001

4.2 Simulação de dados com *shape* regular

4.2.1 Conjunto de dados com $n = 64$

O primeiro conjunto simulado, representado na Figura 3.2(a), possui intervalo de busca entre 0 e 9900. O esboço da função CV para o parâmetro de suavização fixo (Figura 4.4(a)) exibe indícios de uma função estritamente convexa. Portanto, espera-se que os algoritmos sejam capazes de encontrar o mínimo global. O que de fato ocorre, haja visto os resultados da Tabela 4.5, ou seja, tanto o algoritmo Relâmpago quanto os algoritmos OSA e OSA+divisão resultaram em valores próximos a 1511.

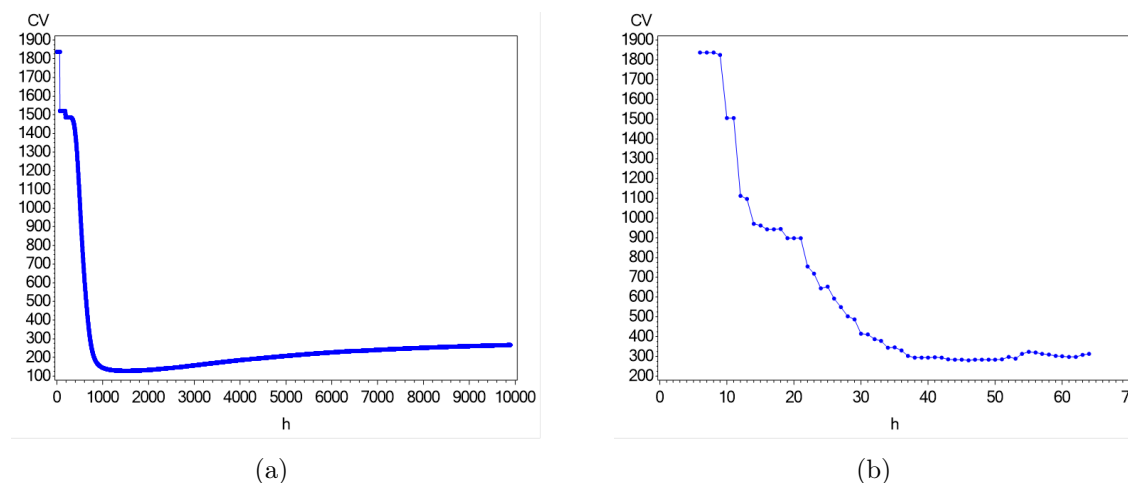


Figura 4.4: Esboço da função CV dos dados simulados com $n = 64$ para os parâmetros de suavização: (a) fixo, (b) adaptável.

Os valores de OSA e OSA+divisão mostraram-se idênticos, devido principalmente de se tratar de um algoritmo adaptado do OSA. Já o algoritmo Relâmpago diferencia-se no mínimo encontrado em 0,595, entretanto, a diferença do CV desse algoritmo para os OSA's é apenas na quinta casa decimal, que é um fato sensato dado que as coordenadas x e y estão regularmente bem espaçadas, portanto, para este caso, pequenas alterações no parâmetro

de suavização produzem mínimas diferenças na função CV.

Do mesmo modo, para o parâmetro de suavização adaptável, os algoritmos não encontram o mesmo resultado, mas continuam próximos. O algoritmo OSA+divisão foi o único a encontrar o mínimo global $h = 46$, enquanto que os outros dois resultaram em 1 número de vizinhos a mais, $h = 47$. Apesar dessa pequena diferença, o CV do OSA+divisão difere em 3,32 em relação aos outros dois, e mesmo assim, o CV com o parâmetro fixo é menor. Portanto, neste caso, o modelo com parâmetro de suavização fixo pode ter melhor ajuste aos dados.

Tabela 4.5: Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função CV dos dados simulados com $n = 64$, tempo de processamento com parâmetro de suavização fixo e adaptável, e R^2 dos modelos RGP para cada parâmetro.

Parâmetro	Algoritmo	h mínimo	$CV(h)$	Tempo(s)	R^2
Fixo	OSA comum	1511,42	128,13467	2,05	0,8519
	OSA+divisão	1511,42	128,13467	5,82	0,8519
	Relâmpago	1512,015	128,13468	8,77	0,8518
Adaptável	OSA comum	47	282,07989	1,47	0,5617
	OSA+divisão	46	278,7555	3,99	0,5691
	Relâmpago	47	282,07989	1,6	0,5617

Principalmente ao observar o grau de ajuste do modelo, o qual no modelo global possui $R^2 = 0,0809$ e modelo com o parâmetro fixo possui $R^2 = 0,851$. Enquanto que o modelo com o parâmetro adaptável, tem R^2 igual a 0,5617, no caso de $h = 46$, conforme Tabela 4.5.

Contudo, essas pequenas diferenças nos parâmetros fixo e do adaptável não produzem alterações nos estimadores do modelo RGP. Pois, com um vizinho a mais não há estimadores que mudam a significância, conforme Tabela 4.6. Assim como no caso fixo, onde o Algoritmo Relâmpago encontrou um h que difere em 0,595, entretanto os estimadores também não alteram sua significância (Tabela 4.7).

Quanto ao tempo de processamento de cada algoritmo, para o caso do parâmetro fixo, há uma disparidade grande entre eles, pois o OSA comum gastou apenas 2,05 segundos, enquanto que o OSA+divisão demorou 5,82 segundos ou 2,84 vezes mais tempo. O algoritmo Relâmpago teve um tempo ainda maior de processamento, com 4,27 vezes o tempo do OSA comum. Deve-se atentar que esse caso possui apenas 64 pontos de observação e a diferença é grande apenas relativamente, mas caso o banco de dados seja maior a diferença absoluta pode ser ainda maior.

Tabela 4.6: Frequências cruzada das estimativas significativas à 90%, 95%, 99% ou não significativas, dos modelos RGP com $n = 64$ e os parâmetros adaptáveis: $h = 46$ e $h = 47$

		Intercepto ($h = 46$)				Varx ($h = 46$)				Varx2 ($h = 46$)							
		NS	S90	S95	S99					NS	S90	S95	S99				
Intercepto*	NS	0	0	0	0	Varx*	NS	62	0	0	0	Varx2*	NS	45	0	0	0
	S90	0	0	0	0		S90	0	2	0	0		S90	0	1	0	0
	S95	0	0	0	0		S95	0	0	0	0		S95	0	0	4	0
	S99	0	0	0	64		S99	0	0	0	0		S99	0	0	0	14

Nota: * $h = 47$; NS - não significativo; S90, S95 e S99 - significativo com 90%, 95% e 99%.

Tabela 4.7: Frequências cruzada das estimativas significativas à 90%, 95%, 99% ou não significativas, dos modelos RGP com $n = 64$ e os parâmetros fixos: $h = 1511, 42$ e $h = 1512, 015$

		Intercepto ($h = 1511, 42$)				Varx ($h = 1511, 42$)				Varx2 ($h = 1511, 42$)							
		NS	S90	S95	S99					NS	S90	S95	S99				
Intercepto*	NS	1	0	0	0	Varx*	NS	61	0	0	0	Varx2*	NS	49	0	0	0
	S90	0	2	0	0		S90	0	2	0	0		S90	0	0	0	0
	S95	0	0	13	0		S95	0	0	1	0		S95	0	0	4	0
	S99	0	0	0	48		S99	0	0	0	0		S99	0	0	0	11

Nota: * $h = 1512, 015$; NS - não significativo; S90, S95 e S99 - significativo com 90%, 95% e 99%.

Observa-se, também, na Tabela 4.5 que o algoritmo OSA+divisão é o mais lento entre os três, para o parâmetro adaptável, diferindo do caso fixo. No entanto, o tempo de processamento é menor que no caso fixo, muito devido ao intervalo de busca menor. Vê-se, ainda, que a diferença relativa entre o OSA comum e o Relâmpago é irrelevante, porém ao comparar com o OSA+divisão a diferença continua semelhante ao encontrado para o parâmetro fixo.

4.2.2 Conjunto de dados com $n = 100$

Os dados com *shape* regular de tamanhos $n = 100$, no intervalo $[0, 7]$, possui um esboço da função CV para h fixo que não indica ser estritamente convexa, como observa-se na Figura 4.5(a). No entanto os valores da função para $h < 0,3$ são muito superiores ao restante do intervalo, deste modo o algoritmo OSA não é afetado pelo mínimo local próximo do 0. Eliminado os pontos para $h < 0,3$ obtém-se uma curva mais suave e com um mínimo evidentemente próximo de 1,1 (Figura 4.5(b)).

Os mínimos encontrados em cada algoritmo foram muito próximos, com diferença apenas na quarta casa decimal, e ainda, uma diferença nula nos resultados da função CV. Com o parâmetro adaptável, também encontram o mesmo mínimo, $h = 89$, portanto o mesmo valor na função CV. O esboço da função CV com o parâmetro adaptável, apresen-

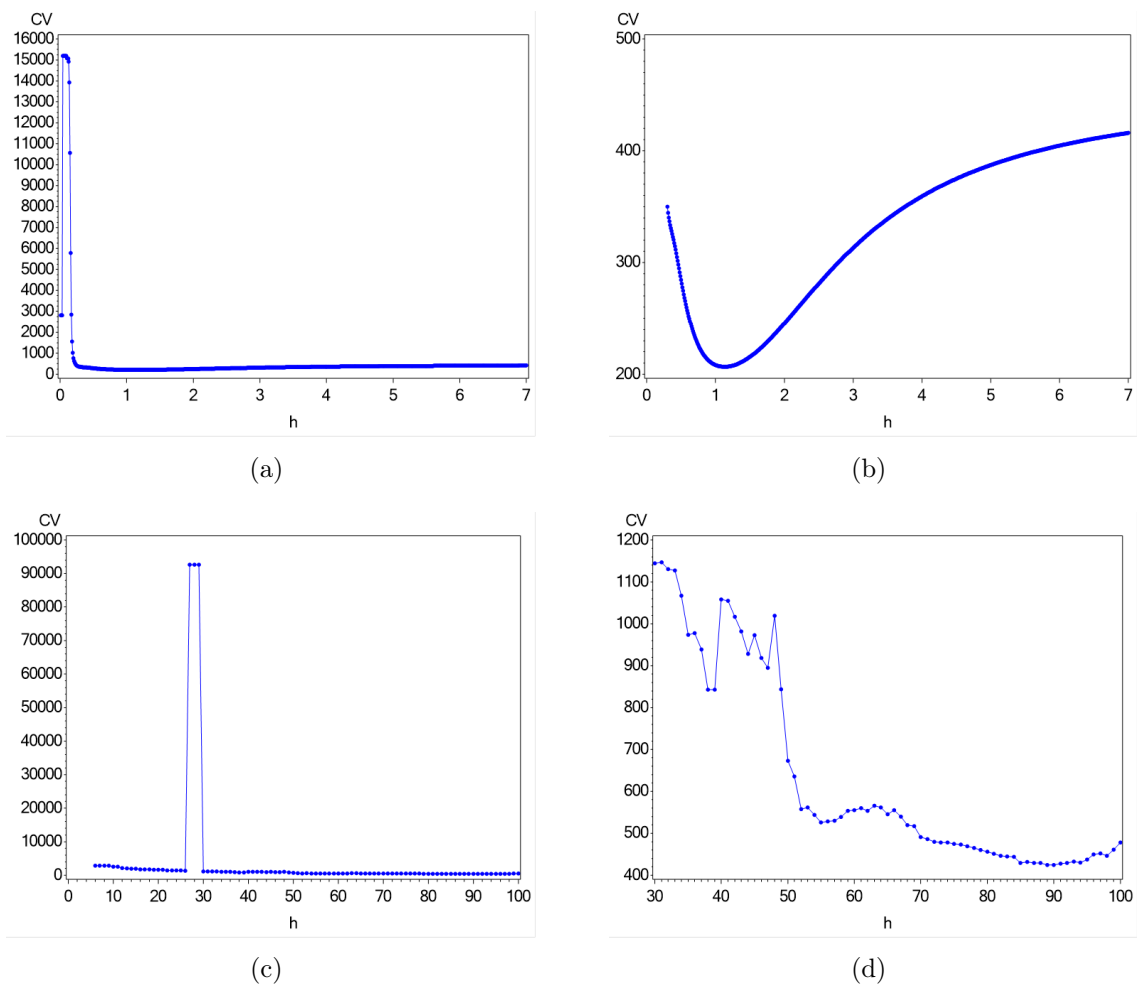


Figura 4.5: Esboço da função CV dos dados simulados com $n = 100$, (a) Fixo: $0 \leq h \leq 7$, (b) Fixo: $0,3 \leq h \leq 7$, (c) Adaptável: $6 \leq h \leq 100$, (d) Adaptável: $30 \leq h \leq 100$.

tado na Figura 4.5(c), possui três pontos distantes dos demais. Ao alterar o intervalo de busca para $30 \leq h \leq 100$, como na Figura 4.5(d), evidencia-se o mínimo global o qual fora encontrado pelos três algoritmos. E devido a igualdade dos resultados encontrados pelos algoritmos, também não existe diferença no resultado dos estimadores do modelo RGP para os parâmetros de suavização encontrados.

Seguindo essa tendência o R^2 também não diferenciou entre os modelos com os parâmetros encontrados. No entanto vê-se novamente superioridade no ajuste do modelo para o parâmetro fixo, com $R^2 = 0,7409$ e o modelo com parâmetro adaptável com $R^2 = 0,4417$, que estão apresentados na Tabela 4.8. No entanto, o R^2 do modelo global, ainda foi muito abaixo do modelo espacial, sendo igual a 0,0162.

Contudo, o tempo de processamento possui detalhes intrigantes, dado que para o parâmetro fixo, o algoritmo Relâmpago levou 26,22 segundos para encontrar o mínimo global,

Tabela 4.8: Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função CV dos dados simulados com $n = 100$, tempo de processamento com parâmetro de suavização fixo e adaptável, e R^2 dos modelos RGP para cada parâmetro.

Parâmetro	Algoritmo	h mínimo	$CV(h)$	Tempo(s)	R^2
Fixo	OSA comum	1,13382	206,62524	3,15	0,7409
	OSA+divisão	1,1338144	206,62524	8,81	0,7409
	Relâmpago	1,1337332	206,62524	26,22	0,7409
Adaptável	OSA comum	89	424,00363	3,73	0,4417
	OSA+divisão	89	424,00363	9,66	0,4417
	Relâmpago	89	424,00363	4,58	0,4417

ou seja, 8,32 vezes mais lento que o OSA. Esse resultado difere muito do visto para o conjunto de dados com $n = 64$, principalmente, porque os outros tempos mantiveram-se equivalentes nos dois conjuntos de dados, $n = 64$ e $n = 100$.

4.2.3 Conjunto de dados com $n = 1024$

Seguindo ainda com os conjuntos de dados com *shape* regular, o terceiro com essa característica possui $n = 1024$, e possui o intuito principal de testar o tempo que cada algoritmo gasta para encontrar o mínimo da função CV. No entanto, observa-se na Figura 4.6(d) que a função CV para o parâmetro adaptável possui mais de um mínimo local, sendo, portanto, um bom exemplo para verificar a capacidade dos algoritmos para encontrar o mínimo global.

Para o parâmetro fixo a curva CV aparenta-se perfeitamente convexa, conforme Figura 4.6(a), que é uma situação ótima para o algoritmo OSA. Neste caso, pode-se notar na Figura 4.6(b) que o mínimo está bem próximo de 0,9. O mínimo global exato é $h = 0,9281317$ (Tabela 4.9), que resulta em um CV igual a 1972,9806. O mesmo mínimo é encontrado no três algoritmos e, conseqüentemente, possui os mesmos estimadores das variáveis explicativas do modelo RGP.

Na Figura 4.6(d) o $h = 1018$ possui menor valor do CV, ou seja, o parâmetro de suavização adaptável ótimo para este caso é $h = 1018$. Este valor foi encontrado tanto pelo OSA quanto pelo OSA+divisão, já o algoritmo Relâmpago encontrou $h = 1004$, que resulta em um CV igual a 5655,0405, quase dez unidades maior que o parâmetro encontrado pelo OSA.

Essa diferença se reflete na significância dos estimadores da variável *varx*, como vê-se na Tabela 4.10, pois 58 estimadores tornam-se significativos quando utiliza-se o $h = 1004$.

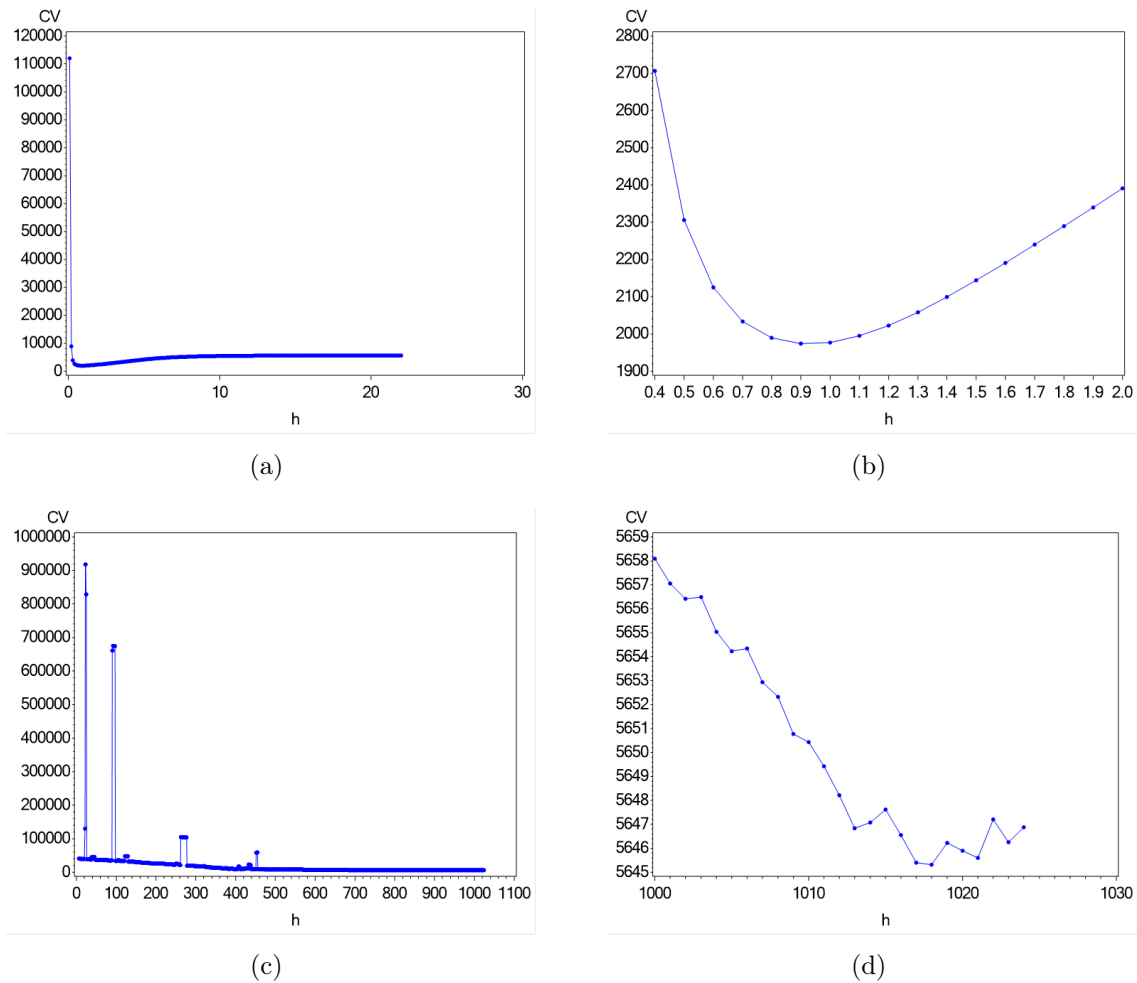


Figura 4.6: Esboço da função CV dos dados simulados com $n = 1024$, (a) Fixo: $0 \leq h \leq 21$, (b) Fixo: $0,4 \leq h \leq 2$, (c) Adaptável: $6 \leq h \leq 1024$, (d) Adaptável: $1000 \leq h \leq 1024$.

O nível de significância também muda para 81 estimadores ao trocar o parâmetro de suavização ótimo, $h = 1018$ para $h = 1004$.

Novamente o ajuste do modelo RGP com parâmetro fixo é melhor que o modelo com o parâmetro adaptável. Contudo, a diferença neste caso é muito maior, com R^2 do modelo de parâmetro adaptável igual a 0,1156 ($h = 1018$) ou 0,1313 ($h = 1004$), conforme mostrado na Tabela 4.9. Esses valores se aproximam muito mais do R^2 do modelo global, igual a 0,0471, do que o modelo com parâmetro fixo, $R^2 = 0,818$. Nota-se que o parâmetro encontrado pelo algoritmo Relâmpago resulta em um CV maior, no entanto quanto aplicado no modelo RGP o R^2 é melhor.

Os resultados da Tabela 4.9 mostram que, para $n = 1024$, o tempo de processamento do algoritmo Relâmpago foi 14,89 vezes o tempo do OSA, demorando mais de 1 hora e 23

Tabela 4.9: Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função CV dos dados simulados com $n = 1024$, tempo de processamento com parâmetro de suavização fixo e adaptável, e R^2 dos modelos RGP para cada parâmetro.

Parâmetro	Algoritmo	h mínimo	$CV(h)$	Tempo(s)	R^2
Fixo	OSA comum	0,9281317	1972,9806	335,12	0,8181
	OSA+divisão	0,9281352	1972,9806	955,02	0,8181
	Relâmpago	0,928513	1972,9807	4990,04	0,8180
Adaptável	OSA comum	1018	5645,3125	426,94	0,1156
	OSA+divisão	1018	5645,3125	1154,52	0,1156
	Relâmpago	1004	5655,0405	369,5	0,1313

minutos para encontrar o mínimo global do parâmetro de suavização fixo, enquanto que o OSA gastou pouco mais de 5 minutos. O algoritmo OSA+divisão continua com a diferença relativa ao OSA semelhante aos encontrados nos outros conjuntos.

Enquanto no caso fixo o algoritmo Relâmpago apresenta tempos cada vez maiores, para o parâmetro adaptável ele apresenta tempo melhores, pois para $n = 1024$ ele converge mais rapidamente que o OSA. Deve-se atentar, no entanto, que o resultado encontrado não foi o mínimo global.

Tabela 4.10: Frequências cruzada das estimativas significativas à 90%, 95%, 99% ou não significativas, dos modelos RGP com $n = 1024$ e os parâmetros adaptáveis: $h = 1018$ e $h = 1004$

		Intercepto ($h = 1018$)				Varx ($h = 1018$)				Varx2 ($h = 1018$)							
		NS	S90	S95	S99					NS	S90	S95	S99				
Intercepto*	NS	0	0	0	0	Varx*	NS	146	0	0	0	Varx2*	NS	0	0	0	0
	S90	0	0	0	0		S90	58	134	0	0		S90	0	0	0	0
	S95	0	0	0	0		S95	0	69	382	0		S95	0	0	0	0
	S99	0	0	0	1024		S99	0	0	12	221		S99	0	0	0	1024

Nota: * $h = 1004$; NS - não significativo; S90, S95 e S99 - significativo com 90%, 95% e 99%.

4.3 Simulação de dados com *shape* aleatório

4.3.1 Conjunto de dados com $n = 72$

A simulação de dados com um *shape* regular apresentou informações importantes, principalmente quanto ao tempo de processamento dos algoritmos, no entanto, um conjunto com essa característica não é possível encontrar em situações reais. O seguinte conjunto de dados dessa subseção possui um *shape* gerado aleatoriamente, assemelhando-se aos casos reais, com $n = 72$ apresentado na Figura 3.3(a).

Imediatamente observa-se padrão semelhante ao *shape* regular no esboço da função CV com o parâmetro de suavização fixo, mostrado nas Figuras 4.7(a) e 4.7(b), pois possui convexidade clara e um ponto de mínimo entre 9 e 10. A semelhança também se observa para o parâmetro adaptável, pois o esboço da função CV indica que não é uma função estritamente convexa, como vê-se na Figura 4.7(d). Além disso, tem-se o mínimo global do h adaptável igual 63, no entanto um mínimo local é visto no intervalo entre 20 e 30.

No intervalo de busca entre 0 e 129, os três algoritmos encontraram o mínimo global, como já esperado devido a convexidade da função com o parâmetro fixo. O mínimo global é definido para $h = 9, 10612$ e a função CV resulta em $1, 8245 \times 10^{15}$ (Tabela 4.11), que é um valor muito menor que o encontrado pelo método adaptável, pois os mínimos encontrados da função CV para o parâmetro adaptável são quase 5 vezes maiores que os encontrados com o h fixo (Tabela 4.11).

E para o h adaptável os mínimos encontrados pelos algoritmos não são iguais, diferenciando-se em 37 vizinhos entre o algoritmo OSA e os outros dois algoritmos. O algoritmo OSA apresentou um mínimo igual a 26, que como observado anteriormente na Figura 4.6(d) é um mínimo local, portanto, este exemplo mostra a vulnerabilidade do algoritmo OSA na busca do parâmetro de suavização ótimo. O algoritmo proposto para evitar esse problema encontrou $h = 63$, assim como o algoritmo Relâmpago, portanto ambos algoritmos convergiram para o mínimo global.

O modelo global possui $R^2 = 0, 9314$, que já é um valor muito alto, porém o modelo RGP resultou em um ajuste quase perfeito com R^2 superior a 0,99, conforme Tabela 4.11. O mais interessante a se notar é o ajuste do modelo com parâmetro adaptável igual a 26, que é um pouco melhor que o modelo com $h = 63$, interessante pois o $h = 26$ não representa o mínimo global da função CV. E ainda, esse modelo possui mesmo R^2 do modelo com parâmetro de suavização fixo.

Tabela 4.11: Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função CV dos dados simulados aleatoriamente com $n = 72$, tempo de processamento com parâmetro de suavização fixo e adaptável, e R^2 dos modelos RGP para cada parâmetro.

Parâmetro	Algoritmo	h mínimo	$CV(h)$	Tempo(s)	R^2
Fixo	OSA comum	9,10612	$1, 8245 \times 10^{15}$	2,08	0,9998
	OSA+divisão	9,10612	$1, 8245 \times 10^{15}$	5,73	0,9998
	Relâmpago	9,10612	$1, 8245 \times 10^{15}$	1223,48	0,9998
Adaptável	OSA comum	26	$9, 1107 \times 10^{15}$	1,77	0,9998
	OSA+divisões	63	$8, 8117 \times 10^{15}$	5,27	0,9975
	Relâmpago	63	8.8117×10^{15}	1,49	0,9975

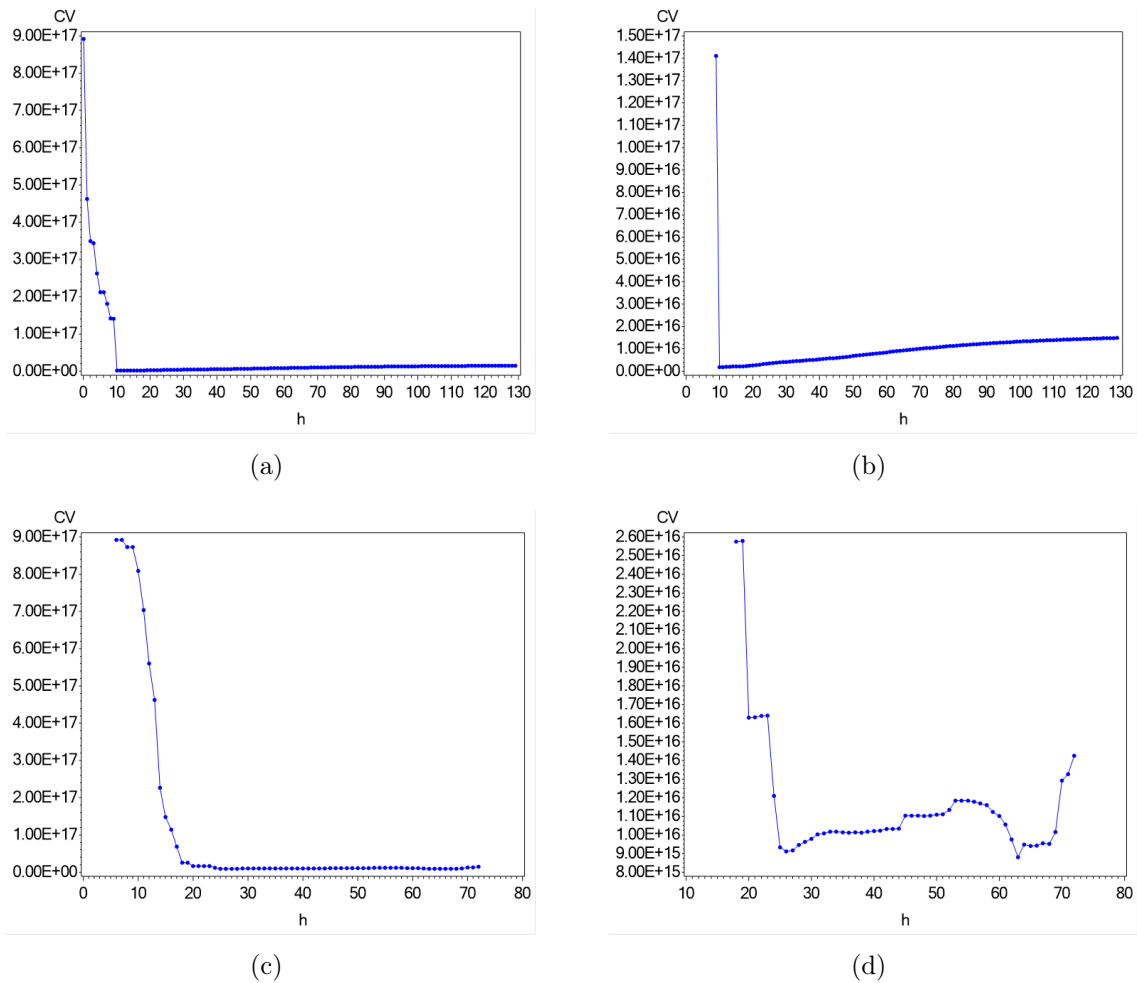


Figura 4.7: Esboço da função CV dos dados simulados aleatoriamente com $n = 72$, (a) Fixo: $0 \leq h \leq 129$, (b) Fixo: $9 \leq h \leq 129$, (c) Adaptável: $6 \leq h \leq 72$, (d) Adaptável: $18 \leq h \leq 72$.

Essa diferença afeta também os estimadores dos modelos RGP, pois com $h = 26$, na variável $varx2$ 4 estimadores são considerados não significativos que são significativos com o parâmetro de suavização ótimo, $h = 63$. E um estimador é definido significativo no modelo com $h = 26$, mas não é significativo no modelo com $h = 63$, conforme Tabela 4.12.

Importante frisar o tempo de processamento do algoritmo Relâmpago para o parâmetro fixo, o qual gastou 588 vezes mais tempo para encontrar o mínimo do que o OSA. Um tempo muito maior que os já mostrados nos outros conjuntos. Os outros tempos seguem com o mesmo padrão de diferença relativa que os dados com *shape* regular apresentou.

Tabela 4.12: Frequências cruzada das estimativas significativas à 90%, 95%, 99% ou não significativas, dos modelos RGP com $n = 72$ e os parâmetros adaptáveis: $h = 63$ e $h = 26$

		Intercepto ($h = 63$)				Varx ($h = 63$)				Varx2 ($h = 63$)							
		NS	S90	S95	S99					NS	S90	S95	S99				
Intercepto*	NS	0	0	0	0	Varx*	NS	0	0	0	0	Varx2*	NS	65	1	2	1
	S90	0	0	0	0		S90	0	0	0	0		S90	1	0	0	0
	S95	0	0	0	0		S95	0	0	0	0		S95	0	0	0	1
	S99	0	0	0	72		S99	0	0	0	72		S99	0	0	0	1

Nota: * $h = 26$; NS - não significativo; S90, S95 e S99 - significativo com 90%, 95% e 99%.

4.3.2 Conjunto de dados com $n = 225$

O último conjunto de dados simulados, representado na Figura 4.8, não apresenta nos resultados grandes diferenças aos outros conjuntos mostrados nesse trabalho. O mais importante nesse caso, com coordenadas aleatórias e $n = 225$, é o h adaptável encontrado pelo algoritmo OSA ser novamente de um mínimo local. Na Figura 4.8(d) é possível notar o mínimo global em h igual a 71, no entanto o algoritmo OSA convergiu para 108, como vê-se na Tabela 4.13. Um h muito distante do que produz o menor valor do CV, além disso, esse valor do CV é 13,99% maior que o CV para $h = 71$.

Para o parâmetro fixo, o esboço da função não aparenta ter a característica de ser estritamente convexa, mas como aconteceu para o conjunto de dados com $n = 100$ (Figura 4.5(a)) os valores da função para $h < 9$ são muito maiores que o restante do intervalo, então não influencia no algoritmo OSA.

Vale notar que novamente o algoritmo OSA é o mais rápido e em caso de função convexa, ele sempre encontra o mínimo global. O algoritmo Relâmpago apresentou nesse caso tempo de processamento quase 14 vezes mais lento que o OSA, mas também encontrou o mínimo global, para o h fixo. Já para o parâmetro adaptável, a diferença relativa é menor, apenas 3,65 vezes o tempo dele em relação ao OSA, e ainda convergiu para o mínimo global, enquanto que o OSA retornou um mínimo local.

Nesse caso, a escolha inadequada do parâmetro, ou seja, $h = 108$ que não representa o mínimo global, resultaria em 57 estimadores da variável $varx$ não significativos, mas que são significativos quando utilizado o parâmetro de suavização global, como pode ser visto na Tabela 4.14. Esse valor representa 25,33% do total, portanto apenas em uma variável do modelo RGP, a escolha do h errado, já teria uma consequência grave no resultado final. E ainda pior é a situação dos estimadores da variável $varx2$, porque 111 estimadores são definidos erroneamente como não significativos, conforme Tabela 4.14, isto é quase 50%

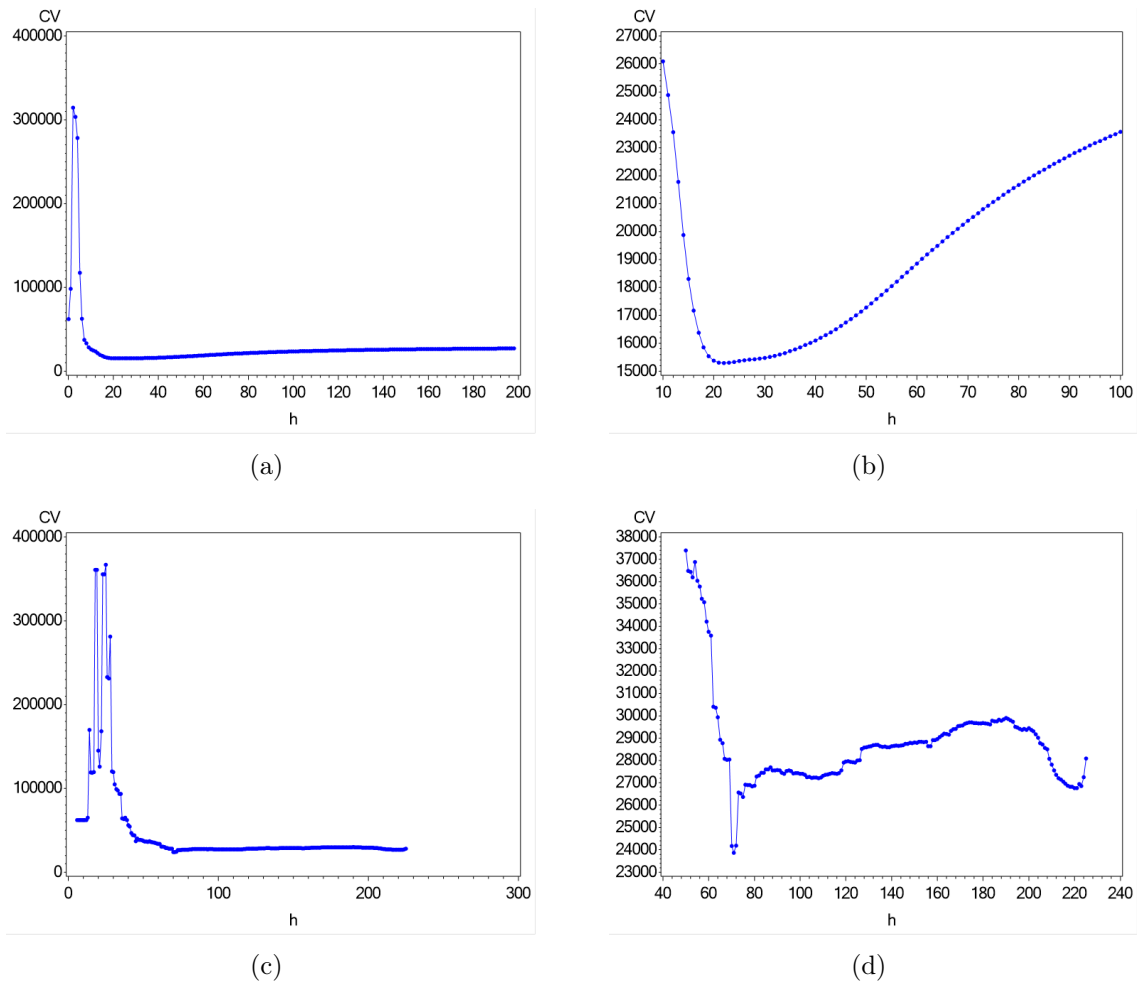


Figura 4.8: Esboço da função CV dos dados simulados aleatoriamente com $n = 225$, (a) Fixo: $0 \leq h \leq 198$, (b) Fixo: $9 \leq h \leq 100$, (c) Adaptável: $6 \leq h \leq 225$, (d) Adaptável: $50 \leq h \leq 225$.

das observações, gerando, portanto, um modelo completamente viesado.

Novamente o R^2 dos modelos RGP com parâmetro de suavização fixo mostrou-se maior que os modelos com parâmetro adaptável, como nota-se na Tabela 4.13. Enquanto que o ajuste do modelo global apresentou um resultado baixo, com $R^2 = 0,2918$, os modelos RGP's possuem ajuste bem melhor, sendo superior a 0,75. E ao focar apenas no parâmetro de suavização adaptável, o h que representa o mínimo global possui um ajuste melhor.

Tabela 4.13: Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função CV dos dados simulados aleatoriamente com $n = 225$, tempo de processamento com parâmetro de suavização fixo e adaptável, e R^2 dos modelos RGP para cada parâmetro.

Parâmetro	Algoritmo	h mínimo	$CV(h)$	Tempo(s)	R^2
Fixo	OSA comum	21,83399	15291,78	19,65	0,8272
	OSA+divisão	21,833982	15291,78	56,25	0,8272
	Relâmpago	21,83399	15291,78	272,00	0,8272
Adaptável	OSA comum	108	27206,339	18,24	0,7559
	OSA+divisões	71	23866,312	50,46	0,7840
	Relâmpago	71	23866,312	66,75	0,7840

Tabela 4.14: Frequências cruzada das estimativas significativas à 90%, 95%, 99% ou não significativas, dos modelos RGP com $n = 225$ e os parâmetros adaptáveis: $h = 71$ e $h = 108$

		Intercepto ($h = 71$)				Varx ($h = 71$)					Varx2 ($h = 71$)						
		NS	S90	S95	S99												
						NS	S90	S95	S99								
Intercepto*	NS	0	1	0	0	Varx*	NS	85	18	29	10	Varx2*	NS	36	23	88	0
	S90	0	0	0	0		S90	0	0	9	15		S90	0	0	40	6
	S95	0	0	0	4		S95	0	0	2	28		S95	0	0	4	15
	S99	0	0	0	220		S99	0	0	0	29		S99	0	0	0	13

Nota: * $h = 108$; NS - não significativo; S90, S95 e S99 - significativo com 90%, 95% e 99%.

4.4 Estudos de casos reais

4.4.1 Dados do estado de Goiás

Ao abordar o caso real o qual busca uma explicação para o rendimento médio dos domicílios pelo tamanho da população nos municípios no estado de Goiás, o primeiro fato a notar-se, é a semelhança do esboço da função CV tanto para o parâmetro fixo quanto para o adaptável, esboço o qual está apresentado na Figura 4.9. Porém, neste caso o parâmetro fixo diferenciou-se um pouco, pois a curva tem o mínimo global em 888 que é o limite de busca. Diante desse fato, intuitivamente a solução poderia ser aumentar o limite superior de busca, no entanto 888km é a maior distância entre dois municípios goianos, portanto aumentar o limite superior não teria efeito prático.

Já o parâmetro adaptável possui um mínimo global, e possui também vários mínimos locais, ou seja, muito semelhante com o que foi visto com os casos aleatórios. O mínimo encontra-se exatamente em $h = 42$ com CV igual a $2,8607 \times 10^{16}$. Novamente o algoritmo OSA não foi capaz de encontrar o mínimo, mesmo que tenha encontrado um valor muito próximo, $h = 43$ (Tabela 4.15). O algoritmo Relâmpago também convergiu para o $h = 43$, portanto, apenas o algoritmo OSA+divisão encontrou o mínimo global.

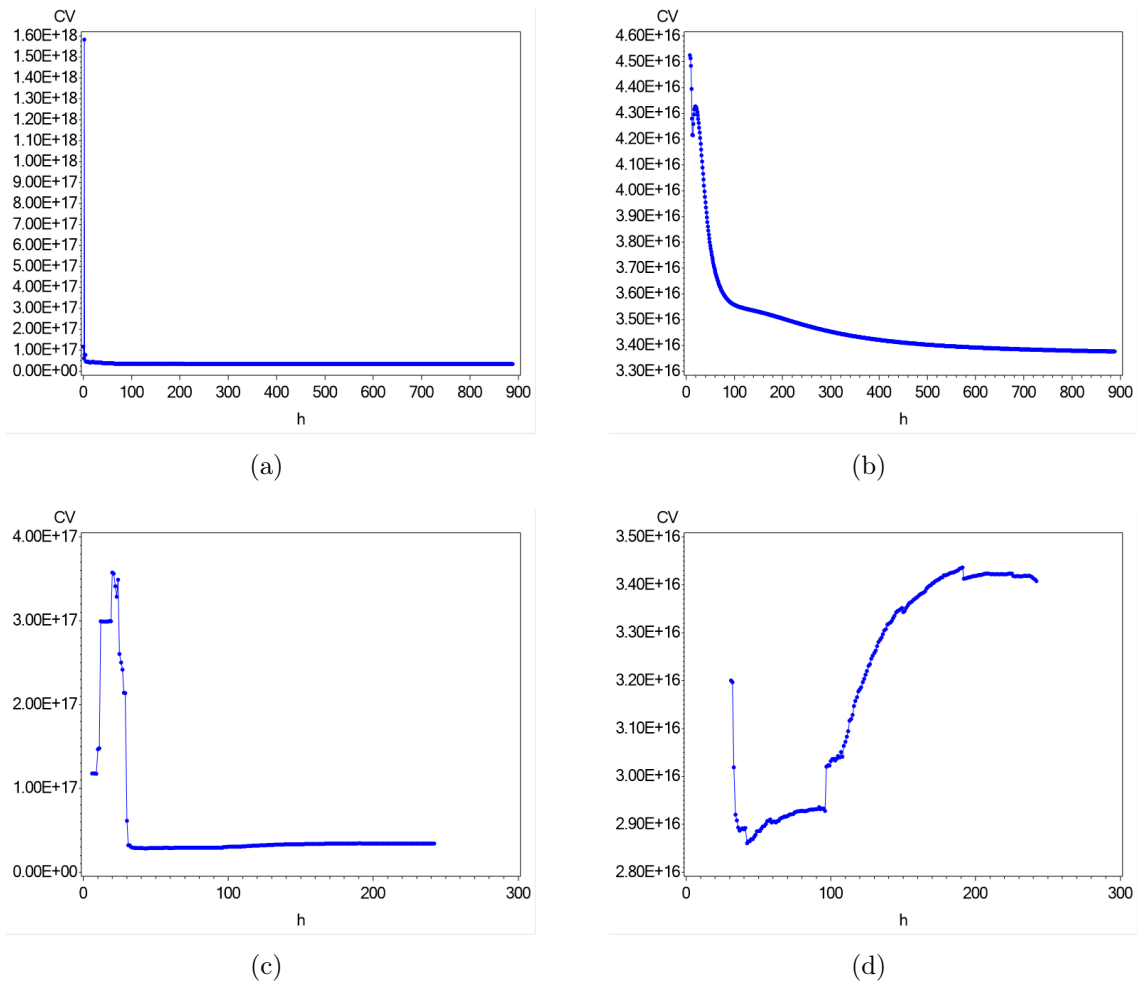


Figura 4.9: Esboço da função CV dos dados de rendimento do estado de Goiás, (a) Fixo: $0 \leq h \leq 888$, (b) Fixo: $8 \leq h \leq 888$, (c) Adaptável: $6 \leq h \leq 242$, (d) Adaptável: $31 \leq h \leq 242$.

Quanto ao ajuste do modelo, observa-se na Tabela 4.15 um bom resultado com o parâmetro adaptável. Já o modelo com parâmetro fixo possui mesmo resultado do modelo global, pois o parâmetro ótimo inclui todas as observações, ou seja, possui a mesma característica do modelo global de incluir igualmente todas as observações para a estimação. Desse modo, o R^2 tanto para o modelo global quanto para o RGP com parâmetro fixo é igual a 0,9488 que é menor que o do modelo com parâmetro adaptável.

Os resultados dos CV's, como os mostrados na Tabela 4.15, não possuem indícios de grandes diferenças para os h 's 42 e 43, o que pode levar a conclusões precipitadas. Pois, os estimadores podem sofrer mais influências do que a função CV. Na Figura 4.10 são apresentados os municípios de Goiás e a intensidade dos parâmetros estimados para a variável População. Pode-se observar na Figura 4.10 que os municípios possuem parâmetros

Tabela 4.15: Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função CV dos dados de Goiás com parâmetro de suavização adaptável e R^2 dos modelos RGP para cada parâmetro.

Algoritmo	h mínimo	$CV(h)$	R^2
OSA comum	43	2.8648×10^{16}	0,9634
OSA+divisão	42	2.8607×10^{16}	0,9637
Relâmpago	43	2.8648×10^{16}	0,9634

estimados em intervalos diferentes ou até mesmo não significativos.

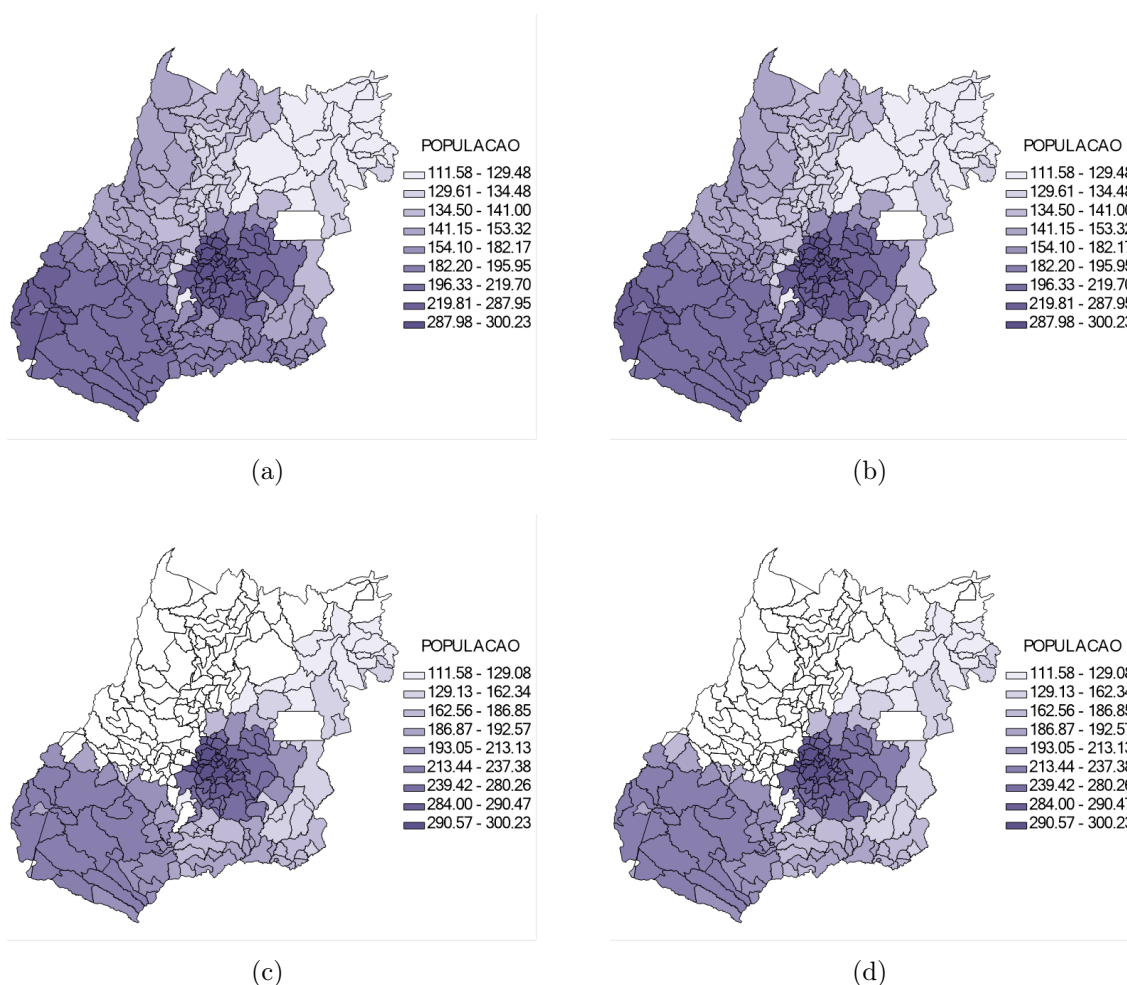


Figura 4.10: Estimadores da variável população do modelo RGP com os dados de Goiás, (a) $h = 42$ e (b) $h = 43$, e estimadores significativos com nível de 90%, (c) $h = 42$ e (d) $h = 43$.

São oito intervalos que agrupam os estimadores, e estão representados nos mapas com tons diferentes de azul, sendo azul mais claro para os estimadores com valores menores, e quanto maior o estimador mais escura torna-se a cor. Nas Figuras 4.10(a) e 4.10(b) poucas diferenças são notadas, mas 12 dos 242 municípios, ou 4,96%, mudam de intervalo. E ainda

nas Figuras 4.10(c) e 4.10(d), as quais mostram também os estimadores significativos, são três municípios que deixam de ser significativos, dois no lado oeste de Goiás e um no nordeste do estado.

As diferenças dos estimadores entre os modelos RGP com h igual a 42 e 43 são pequenas. Porém, em uma situação onde o algoritmo OSA resulta em um mínimo local com um h distante do mínimo global, como ocorreu nos dados simulados aleatoriamente, as diferenças dos estimadores podem ser maiores.

Na Figura 4.9(d) é possível notar um mínimo local em $h = 59$, então uma possível situação do modelo RGP com esse parâmetro de suavização pode-se observar a dimensão das diferenças dos estimadores. Por isso, na Figura 4.11 está representado, para comparação, os estimadores com o parâmetro de suavização ótimo e com $h = 59$.

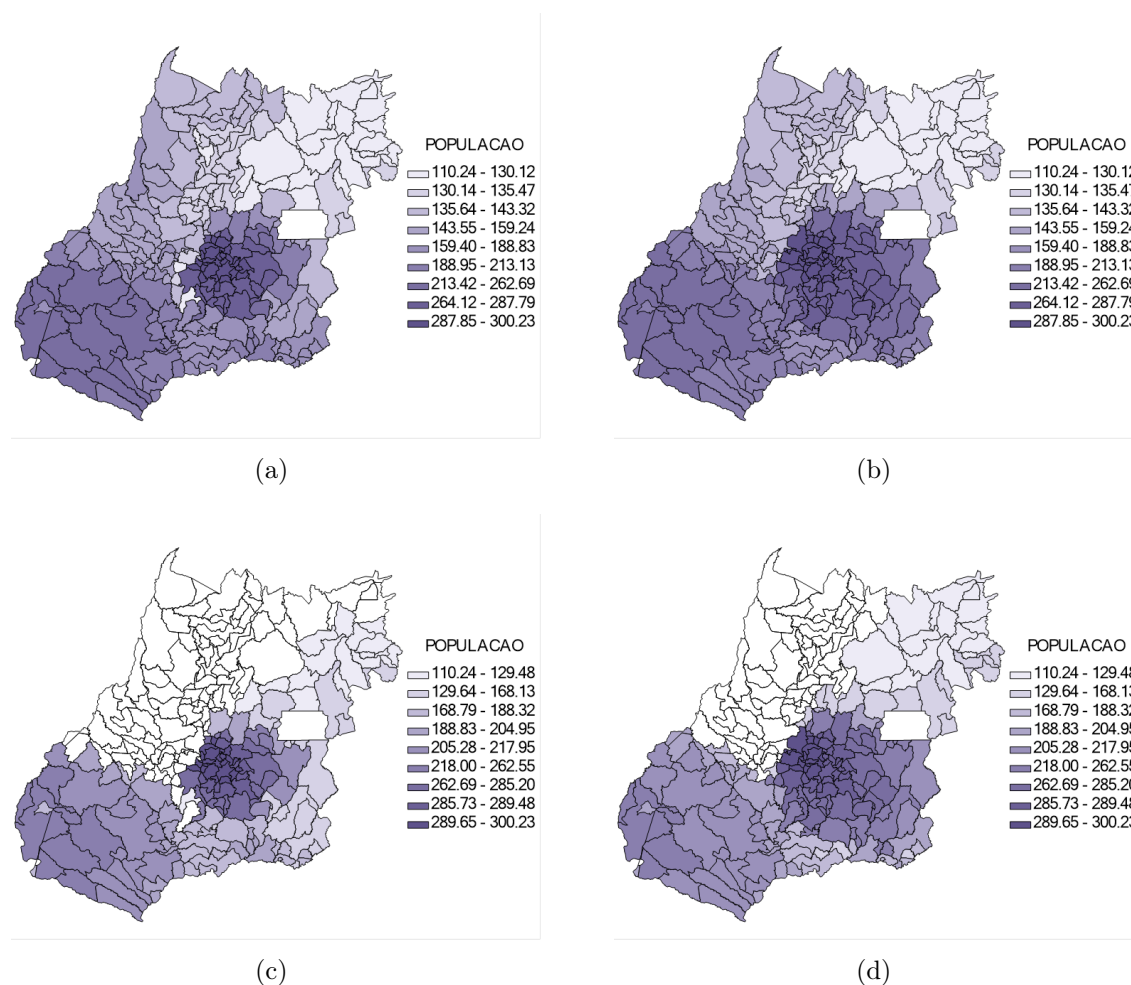


Figura 4.11: Estimadores da variável população do modelo RGP com os dados de Goiás, (a) $h = 42$ e (b) $h = 59$, e estimadores significativos com nível de 90%, (c) $h = 42$ e (d) $h = 59$.

E nota-se muitas diferenças tanto na intensidade dos parâmetros estimados dos municípios quanto dos estimadores significativos. Principalmente na região oeste e central do estado que apresentam municípios que pertencem a intervalos diferentes em cada caso. E os municípios com estimadores significativos são diferentes mais na região nordeste e central do estado. Deste modo um parâmetro de suavização definido precipitadamente, pode alterar a situação de muitos locais, levando, portanto, a conclusões imprecisas.

4.4.2 Dados de acidentes de trânsito na cidade de Fortaleza

Como os acidentes de trânsito na cidade de Fortaleza são dados de contagem, o modelo ideal para este caso deve ser feito a partir da distribuição de Poisson. Neste caso, a variável dependente é o número de acidentes, enquanto que as variáveis independentes são Área (região da cidade onde ocorreu o acidente), Arterial (se a rua é Arterial ou não), Coletora (se a rua é Coletora ou não) e População total do bairro onde aconteceu o acidente.

No modelo de Regressão Poisson Geograficamente Ponderada é mais comum o uso do critério AICc, portanto na Figura 4.12 apresenta-se o esboço utilizando este critério, e novamente observa-se comportamentos semelhantes às funções CV's apresentadas nos exemplos anteriores. Ao cobrir todo o intervalo de busca, apresentado na Figura 4.12(a), não é possível entender o comportamento da função, pois um ponto destoa muito dos outros. Recortando o intervalo entre 200 e 21000, além de retirar os pontos discrepantes, como na Figura 4.12(b), observa-se que a função possui apenas um mínimo, não existindo indícios de mínimos locais, assim como nas funções CV com o modelo com variável resposta tendo distribuição normal.

Encurtando ainda mais o intervalo, o mínimo global é encontrado no intervalo entre 900 e 1100, como mostrado na Figura 4.12(c), e nesta Figura nota-se que a função continua estritamente convexa mesmo em pequenos intervalos. Para o parâmetro de suavização adaptável o mínimo global é facilmente visto em $h = 14$, e diferente dos casos anteriores, essa função não apresentou mínimos locais, portanto o algoritmo OSA possui grande eficiência para encontrar o mínimo global neste conjunto de dados.

Mesmo que no intervalo para $h > 1000$ não possui mínimos locais, no software GWR4 o mínimo encontrado utilizando o Critério de Akaike foi para um h igual a 2240,007, que pode ser observado na Figura 4.13. No entanto, o critério padrão do software para determinar os limites de busca definiu o intervalo de 2240,007 até 10543,096, apresentado

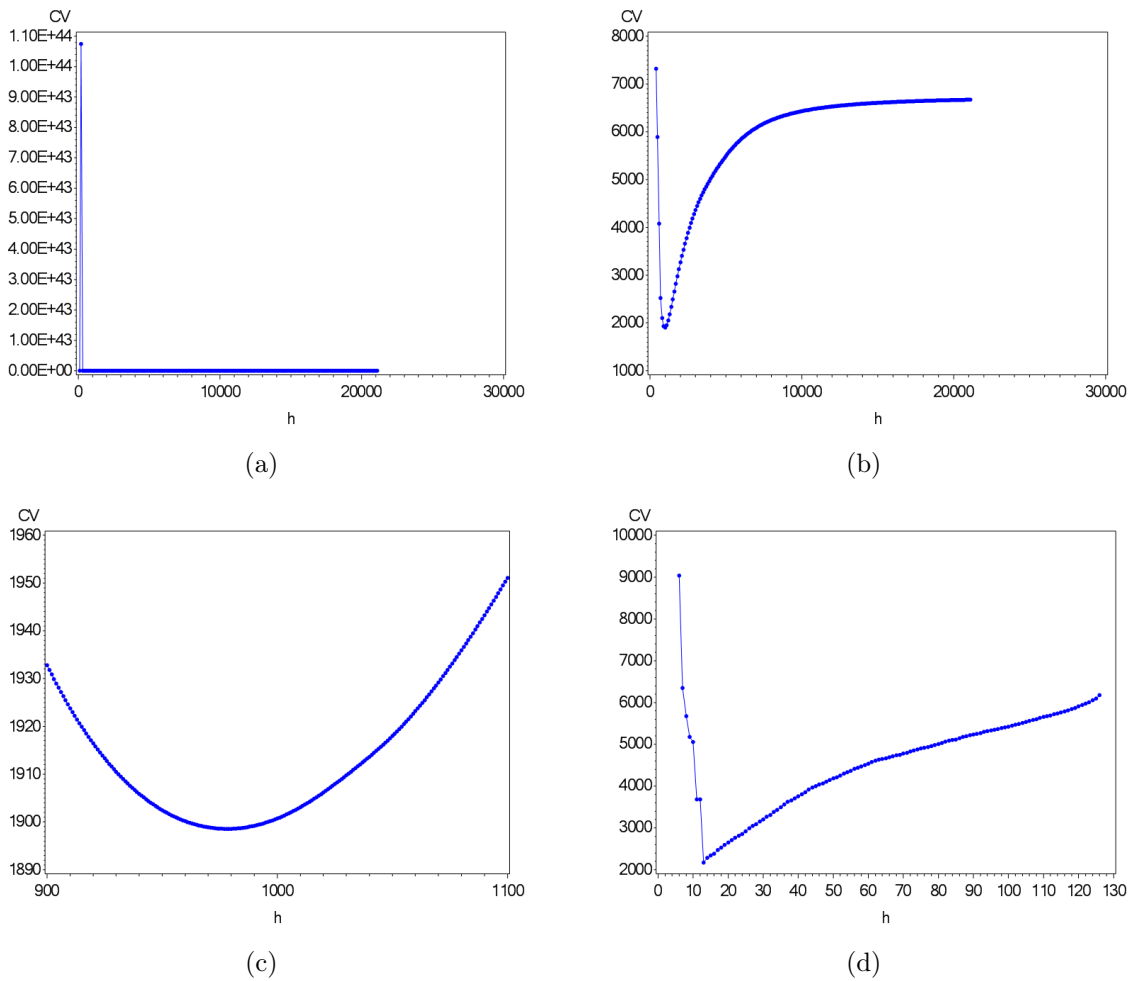
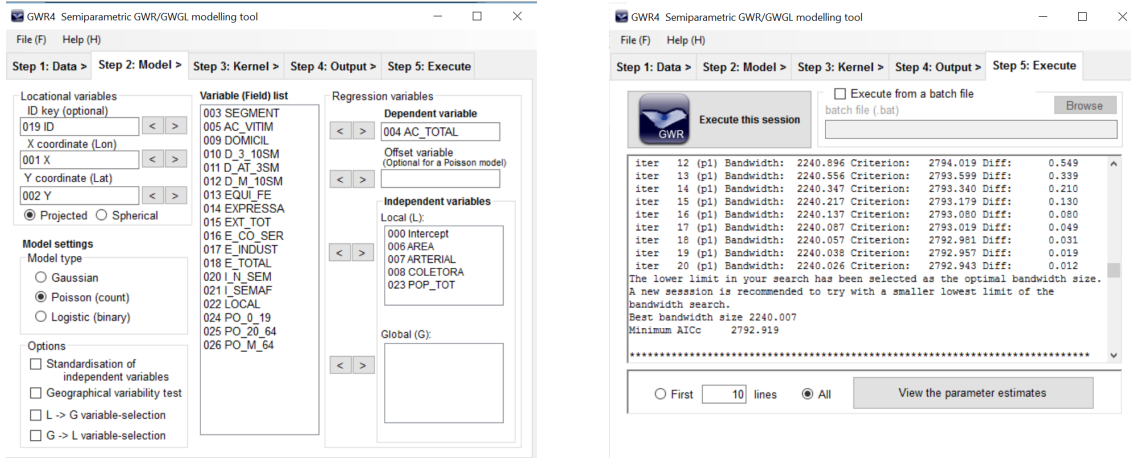


Figura 4.12: Esboço da função AICc dos dados de acidentes de trânsito da cidade de Fortaleza, (a) Fixo: $0 \leq h \leq 21000$, (b) Fixo: $200 \leq h \leq 21000$, (c) Fixo: $900 \leq h \leq 1100$, (d) Adaptável: $6 \leq h \leq 126$.

na Figura 4.13(c). Portanto, com as informações da Figura 4.12, é perfeitamente viável que o mínimo encontrado pelo GWR4 seja o limite inferior de busca, pois viu-se que para o $h > 1000$ a função é sempre crescente, com um único mínimo próximo de 980.

O algoritmo OSA, ao analisar o intervalo completo, consegue encontrar o mínimo global para o caso com o parâmetro fixo, conforme Tabela 4.16. O h encontrado foi igual a 978,12, que corresponde ao valor mínimo que pode ser notado na Figura 4.12(c). O mesmo valor também foi encontrado pelos os algoritmos OSA+divisão e o Relâmpago, diferenciando apenas no tempo de processamento de cada um. Mais uma vez o algoritmo Relâmpago apresentou tempo muito superior ao OSA, enquanto que o algoritmo OSA+divisão continua com tempo maior, mas com a diferença relativa semelhante aos outros casos.

Já o caso adaptável, os algoritmos OSA e OSA+divisão convergiram para o mínimo



(a)

(b)

```

*****
GWR (Geographically weighted regression) bandwidth selection
*****

Bandwidth search <golden section search>
Limits: 2240.00738691888, 10543.0960581092]
Golden section search begins...
Initial values

```

(c)

Figura 4.13: *Prints* dos resultados obtidos através do software GWR4, (a) Definição das variáveis, (b) Resultado da algoritmo OSA, (c) Limites de busca definidos pelo GWR4.

global, ou seja, para $h = 14$. O algoritmo Relâmpago retornou um resultado incomum, dado que o h encontrado foi 17 e esse não é um mínimo local. Pois os para o $h > 14$ a função é sempre crescente. O tempo de processamento do algoritmo Relâmpago foi bom, dado que foi próximo do OSA, mas ter encontrado $h = 17$ aumenta a insegurança sobre esse algoritmo.

Tabela 4.16: Mínimo dos algoritmos OSA, OSA+divisão e Relâmpago da função com critério AICc dos dados dos acidentes de trânsito e tempo de processamento com parâmetro de suavização fixo e adaptável

Parâmetro	Algoritmo	h mínimo	AICc(h)	Tempo(s)
Fixo	OSA comum	978,12271	1898,5481	52,24
	OSA+divisão	978,12271	1898,5481	109,44
	Relâmpago	978,12271	1898,5481	575,21
Adaptável	OSA comum	14	2171,3947	25,13
	OSA+divisões	14	2171,3947	68,81
	Relâmpago	17	2469,0074	28,79

Capítulo 5

Conclusões

O algoritmo Otimização por Seção Áurea (OSA) claramente possui limitações com funções que não sejam estritamente convexas. Em funções com essa característica o algoritmo pode encontrar o mínimo global, caso os cortes definidos pelo algoritmo direcionem convenientemente para o melhor resultado. Porém, manter o uso do OSA para buscar o mínimo com essas funções é fazer uma tentativa de sorte.

Nas duas funções determinísticas mostradas neste trabalho, que possuem dois ou três mínimos, o algoritmo OSA encontrou o mínimo local. Sendo, portanto, inadequado o uso do OSA, e necessário o uso de algoritmos mais robustos. Uma solução escolhida foi dividir o intervalo de busca e aplicar o OSA em cada divisão. Assim como os algoritmos Relâmpago e o Harmônico, o algoritmo OSA+divisão mostrou-se eficiente para encontrar os mínimos globais das duas funções determinísticas.

Um ponto importante a se destacar é o tempo de processamento muito maior do algoritmo Harmônico, o que foi um dos motivos para abandonar o seu uso nos exemplos para encontrar o parâmetro de suavização ótimo. Mesmo o algoritmo OSA+divisão que repete o mesmo processo 3 vezes, possui um tempo muito menor. A discrepância dos tempos é tão grande na função com três mínimos que tornou-se inviável a continuação com o Harmônico. Principalmente ao observar os esboços das funções CV para o parâmetro de suavização adaptável, que indicava muitos mínimos locais, o que poderia acarretar em tempos elevadíssimos.

Os dados simulados ajudaram a entender o tempo de processamento de cada algoritmo no contexto da Regressão Geograficamente Ponderada (RGP). Não somente isso, como também entender o comportamento da função CV com a variação do parâmetro de sua-

vização. Em geral, tanto para o *shape* com coordenadas regulares quanto aleatórias, os esboços das funções CV apresentam comportamento semelhantes, independente do tamanho do conjunto de dados.

O que não indica o mesmo comportamento são os esboços com parâmetro fixo comparado aos esboços com parâmetro adaptável. Pois com h fixo, observou-se uma função mais “suave”, com poucos ou nenhum mínimo local. E mesmo os mínimos locais estavam em escalas muito superiores ao do mínimo global, o que pouco influenciava no algoritmo de busca. Por isso, o algoritmo OSA convergiu sempre para o mínimo global, assim como os outros dois algoritmos.

O caso do parâmetro de suavização adaptável foi bem mais significativo o uso do algoritmo OSA. Claramente a função CV não é estritamente convexa, possuindo muitos mínimos locais. Em vista disso, basear-se apenas em busca por seção pode acarretar alguns equívocos. E dois exemplos mostrados nesta dissertação mostraram uma diferença muito grande entre o parâmetro de suavização encontrado pelo OSA e o que seria o parâmetro ótimo. Em alguns casos a diferença foi de apenas um vizinho a ser considerado, mas no caso mais grave a diferença foi de 37. Caso o pesquisador utilize o h incorreto, ele deixaria de incluir 37 vizinhos na modelagem, o que claramente perderia muita informação.

No mapa de Goiás pôde-se ver o quão diferente seria um resultado com um h muito diferente do ótimo. Neste caso, muitos estimadores dos municípios tornaram-se significativos, e mais municípios ainda mudaram muito a intensidade do seu estimador da população.

Portanto vale o esforço do pesquisador para encontrar o mínimo global da função CV, principalmente, porque a Regressão Geograficamente Ponderada foca em encontrar diferenças no espaço, sendo possível, então, notar qualquer mudança feita entre um modelo nesse contexto.

Em virtude desse problema que foram propostos outros algoritmos para esta finalidade. O algoritmo relâmpago oscilou entre resultados precisos e resultados que ficaram próximos do ótimo. Entretanto, para n grande o algoritmo Relâmpago mostra-se pouco viável para a busca do mínimo, pois o tempo de processamento dele chegou a ser 14 vezes o tempo do OSA, no conjunto de dados com $n = 1024$ e 588 vezes no conjunto com $n = 72$. E observou-se então que quanto maior o conjunto de dados e maior o intervalo de busca, maior a distância relativa entre os algoritmos OSA e Relâmpago. Assim, em um eventual conjunto de dados que considera todos os municípios brasileiros, o algoritmo Relâmpago poderia demorar dias ou semanas para encontrar o parâmetro de suavização ótimo.

O algoritmo proposto OSA+divisão foi mais constante, obtendo o parâmetro de suavização ótimo em todos os casos estudados nesse trabalho. Mesmo que ele tenha se mostrado certo, a confiança nele não deve ser 100%, pois dividir o intervalo em 3 partes não evita o erro que ocorre no algoritmo OSA comum. Dado que o comportamento da função CV não segue um padrão, o mesmo problema que acontece no intervalo inteiro pode ocorrer dentro das divisões, e dessa maneira seria necessário aumentar o número de divisões para capturar o mínimo global.

Porém, essa quantidade de divisões pode ser muito subjetiva, sendo uma quantidade que pode mudar para cada cenário. Em um caso extremo, com dezenas ou centenas de divisões, o tempo de processamento poderia se tornar ainda maior que o do algoritmo Relâmpago. Porém ele sempre encontraria o melhor h possível.

Viu-se também que o maior problema do algoritmo Relâmpago é com o tempo para encontrar o parâmetro de suavização fixo, já para o adaptável o tempo é por vezes até menor que o OSA. Seria, então, interessante optar pelo uso do algoritmo relâmpago na condição do parâmetro adaptável, e apenas neste caso. Sendo mais aconselhável, por enquanto, utilizar o algoritmo OSA com as divisões para o parâmetro fixo.

Outro ponto importante a se frisar é que o ajuste do modelo RGP, quando utilizando o parâmetro fixo, mostrou-se melhor que o modelo com parâmetro adaptável, na maioria das vezes, com valores muito superiores do R^2 . Nos casos em que isso não ocorreu a diferença do R^2 foram pequenas. Entretanto, para os dois casos, o ajuste sempre foi melhor que o modelo global.

Por fim, mostrou-se também que o software GWR4 pode resultar em modelos menos adequados, pois utiliza o algoritmo OSA e defini os limites de maneira que pode não incluir o mínimo global. Quanto aos limites, o usuário poderia alterá-los, mas em casos onde o pesquisador não se atenta a essa questão, o modelo utilizado poderia não ser o melhor possível.

5.1 Limitações do Trabalho

Algumas limitações deste trabalho são:

- Não foram comparados resultados com o software estatístico R;
- Não foram apresentados resultados para Regressão Geograficamente Ponderada para

distribuição logística ou binomial negativa;

- Foram feitas poucas comparações entre algoritmos de busca. Dado que existem inúmeros algoritmos, outros algoritmos poderiam ser incluídos para comparar com o algoritmo Harmônico e Relâmpago;
- Por optar pelo uso do SAS[®] Studio, os tempos de processamentos podem ter sido bem maiores do que seriam se utilizasse um software que fizesse o processamento na própria máquina.
- O algoritmo proposto, OSA+divisão, pode eventualmente convergir para um valor que não seja o melhor.

5.2 Sugestões para Trabalhos Futuros

Para complementar este estudo, poderia-se observa o comportamento da função CV na circunstância do parâmetro adaptável para cada observação, que seria mais extenso, pois ter-se-ia um esboço da curva para cada observação. E ainda, entender qual tipo de algoritmo seria mais preciso para essa situação. Pois, os algoritmos propostos não se adequaram perfeitamente aos dois tipos de parâmetro estudados nesta dissertação. Dessa maneira um algoritmo específico, poderia encaixar-se neste contexto para sanar essa vulnerabilidade nas estimações. E por fim comparar quais modelos apresentam melhores resultados quando o verdadeiro mínimo global é encontrado.

Referências Bibliográficas

- Anselin, L. (1988). *Spatial Econometrics*. Kluwer Academic Publishers.
- Atkinson, P. M., German, S. E., Sear, D. A., e Clark, M. J. (2003). Exploring the relations between riverbank erosion and geomorphological controls using geographically weighted logistic regression. *Geographical Analysis*, 35(1):58–82.
- Berkopec, A. (2012). Fast particles as initiators of stepped leaders in cg and ic lightnings. *Journal of Electrostatics*, 70(5):462–467.
- Brunsdon, C., Fotheringham, A. S., e Charlton, M. E. (1998). Geographically weighted regression - modelling spatial non-stationarity. *The Statistician*, 47(3):431–443.
- Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836.
- Dul'zon, A. A., Lopatin, V. V., Noskov, M. D., e Pleshkov, O. I. (1999). Modeling the development of the stepped leader of a lightning discharge. *Technical Physics*, 44(4):3.
- Fotheringham, A., Brunsdon, C., e Charlton, M. (2000). *Quantitative Geography – Perspectives on Spatial Data Analysis*. SAGE Publications.
- Fotheringham, A., Brunsdon, C., e Charlton, M. (2002). *Geographically Weighted Regression: The Analysis of Spatially Varying Relationships*. Wiley.
- Fotheringham, A. S., Charlton, M., e Brunsdon, C. (1996). The geography of parameter space: an investigation of spatial non-stationarity. *International Journal of Geographical Information Systems*, 10(5):605–627.
- Fotheringham, A. S., Crespo, R., e Yao, J. (2015). Geographical and temporal weighted regression (gtwr). *Geographical Analysis*, 47(4):431—452.
- Geem, Z. W. e Kim, J. H. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation*, 76(2):60–68.
- Glover, F. e Kochenberger, G. (2003). *Handbook of Metaheuristics*. International Series in Operations Research & Management Science. Springer US.

- Greig, D. M. (1980). *Optimisation*. Longman Publishing Group.
- Hosmer, D. W. e Lemeshow, S. (2000). *Applied Logistic Regression*, (second ed.). Wiley.
- Huang, B., Wu, B., e Barry, M. (2010). Geographically and temporally weighted regression for modeling spatio-temporal variation in house prices. *International Journal of Geographical Information Science*, 24(3):383–401.
- Islam, M., ·Shareef, H., ·Mohamed, A., e ·Wahyudie, A. (2016). A binary variant of lightning search algorithm: Blsa. *Soft Computing*, pages 1–20.
- Koenker, R. (2005). *Quantile regression*. Cambridge University Press.
- LeSage, J. P. (2004). A family of geographically weighted regression models. In: *Advances in spatial econometrics*, pages 241–264. Springer Berlin Heidelberg.
- Lima, A. (2015). Regressão beta geograficamente ponderada. Master’s thesis, Universidade de Brasília.
- Luenberger, D. G. (1984). *Linear and Nonlinear programming*. Springer.
- Luke, S. (2013). *Essentials Metaheuristics*, (second ed.). Lulu. Available for free at <http://cs.gmu.edu/~sean/book/metaheuristics/>.
- Nakaya, T., Fotheringham, A., Brunsdon, C., e e Charlton, M. (2005). Geographically weighted poisson regression for disease association mapping. *Statistics in Medicine*, 24(17).
- Neter, J., Kutner, M. H., Nachtsheim, C. J., e Wasserman, W. (1996). *Applied linear statistical models*, volume 4. Irwin Chicago.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., e Flannery, B. P. (2007). *Numerical Recipes*, (third ed.). Cambridge University Press.
- Shareef, H., Ibrahim, A. A., e Mutlag, A. H. (2015). Lightning search algorithm. *Applied Soft Computing*, 36:315–333.
- Silva, A. R. e Fotheringham, A. S. (2016). The multiple testing issue in geographically weighted regression. *Geographical Analysis. Forthcoming*, 48(3):233–247.
- Silva, A. R. e Rodrigues, T. C. V. (2014). Geographically weighted negative binomial regression - incorporating overdispersion. *Statistics and Computing*, 24(5):769–783.
- Tobler, W. (1979). Cellular geography. *Philosophy in Geography*, 20:379–386.
- Wang, G. e Guo, L. (2013). A novel hybrid bat algorithm with harmony search for global numerical optimization. *Journal of Applied Mathematics*, 2013:21.

Yang, W. (2014). *An extension of geographically weighted regression with flexible bandwidths*. PhD thesis, University of Saint Andrews.

Apêndice A

Simulações de dados para o *shape* regular

```
/****** Dados 64 *****/  
  
data simulated;  
do sim=1 to 1;  
do i=1 to 64;  
x=1000*mod(i-1,8);  
y=1000*floor((i-1)/8);  
varx=ranuni(sim);  
varx2=ranuni(sim);  
error=rannor(sim);  
b0=5;b1=0;b2=0;  
vary=b0+b1*varx+b2*varx2+error;  
output;  
end;  
end;  
run;  
data simulated;set simulated;  
if x<3000 and y<3000 then do;  
b1=1;b2=5;  
end;  
if x>4000 and y>4000 then do;  
b1=-1;b2=-5;  
end;  
vary=b0+b1*varx+b2*varx2+error;  
run;  
proc gplot data=simulated;  
plot y*x=b1;  
run;  
quit;
```

```

/***** Dados 100 *****/

data simulated;
do sim=1 to 1;
do i=1 to 100;
x=0.5*mod(i-1,10);
y=0.5*floor((i-1)/10);
varx=ranuni(sim);
varx2=ranuni(sim);
error=rannor(sim);
b0=5;b1=0;b2=0;
vary=b0+b1*varx+b2*varx2+error;
output;
end;
end;
run;

data simulated;set simulated;
if x<2 and y<2 then do;
b1=1;b2=5;
end;
if x>2.5 and y>2.5 then do;
b1=-1;b2=-5;
end;
vary=b0+b1*varx+b2*varx2+error;
run;

proc gplot data=simulated;
plot y*x=b1;
run;
quit;

```

```

/***** Dados 1024 *****/

data simulated;
do sim=1 to 1;
do i=1 to 1024;
x=0.5*mod(i-1,32);
y=0.5*floor((i-1)/32);
varx=ranuni(sim);
varx2=ranuni(sim);

```

```

error=rannor(sim);
b0=5;b1=0;b2=0;
vary=b0+b1*varx+b2*varx2+error;
output;
end;
end;
run;
data simulated;set simulated;
if x>4 and x<11 and y>4 and y<11 then do;
b1=2;b2=8;
end;
vary=b0+b1*varx+b2*varx2+error;
run;
proc gplot data=simulated;
plot y*x=b1;
run;
quit;

```

Simulações de dados para o *shape* irregular

```

/***** Dados 72 *****/

data simulated;
n=72;
sta=5;
call streaminit(38219);
do i=1 to n;
if i=49 then sta=25;
x=rand('normal',100,sta);
y=rand('normal',100,sta);
b0=0.000005*x**3+0.000005*y**3;
b1=sqrt((x*x)*(y*y));
b2=2*(-(x/5)*(x/5)-(y/5)*(y/5)+0.13);
varx1=rand('normal',b1,2);
varx2=rand('EXPONENTIAL',1/abs(b2));
vary=b0+b1*varx1+b2*varx2+rand('normal',0,5);
output;
end;
run;

```

```

proc gplot data=simulated;
plot y*x;
symbol v=dot;
run;
quit;

/***** Dados 225 *****/

data simulated;
n=225;
sta=10;
call streaminit(2831);
do i=1 to n;
if i=151 then sta=25;
x=rand('normal',100,sta);
y=rand('normal',100,sta);
b0=0.000005*x**3+0.000005*y**3;
b1=sqrt((x*x)/(y*y));
b2=2*(-(x/5)*(x/5)-(y/5)*(y/5)+0.13)/1000;
varx1=rand('normal',b1,2);
varx2=rand('EXPONENTIAL',abs(b2));
vary=b0+b1*varx1+b2*varx2+rand('normal',0,5);
output;
end;
run;
proc gplot data=simulated;
plot y*x;
symbol v=dot;
run;
quit;

```