

ANA BEATRIZ TOZZO MARTINS

**ANÁLISE GEOESTATÍSTICA DE DADOS
COMPOSICIONAIS**

CURITIBA

ABRIL 2010

ANA BEATRIZ TOZZO MARTINS

**ANÁLISE GEOESTATÍSTICA DE DADOS
COMPOSICIONAIS**

Tese apresentada ao Curso de Pós-graduação em Métodos Numéricos em Engenharia do Setor de Tecnologia do Centro de Estudos de Engenharia Civil Professor Inaldo Ayres Vieira da Universidade Federal do Paraná, como requisito parcial à obtenção do título de Doutor em Ciências.

Orientador: Prof. PhD. Paulo Justiniano
Ribeiro Jr.

**CURITIBA
ABRIL 2010**

TERMO DE APROVAÇÃO

ANA BEATRIZ TOZZO MARTINS

ANÁLISE GEOESTATÍSTICA DE DADOS COMPOSICIONAIS

Tese aprovada como requisito parcial para obtenção do grau de Doutora em Ciências, pelo Programa de Pós-Graduação em Métodos Numéricos em Engenharia do Setor de Tecnologia do Centro de Estudos de Engenharia Civil Professor Inaldo Ayres Vieira da Universidade Federal do Paraná, pela seguinte banca examinadora:

Prof. PhD. Paulo Justiniano Ribeiro Jr
Universidade Federal do Paraná

Prof. PhD. Nelson Luís Dias
Universidade Federal do Paraná

Prof. Dr. Antônio Carlos Andrade
Gonçalves
Universidade Estadual de Maringá

Profa. Dra. Edna Afonso Reis
Universidade Federal de Minas Gerais

Prof. PhD. Paulo Sérgio Lúcio
Universidade Federal do Rio Grande do
Norte

Curitiba, 9 de abril de 2010

À meus pais, Joaquim e Edila, e irmãs, Luisa e Maria do Carmo.

AGRADECIMENTOS

À Jesus que é o caminho, a verdade e a vida (João, 14-6), que me levou a caminhar este tempo todo dedicado aos estudos, descobrindo a cada dia na sua verdade o que eu deveria escrever e dando-me a graça da vida.

Ao Prof. Paulo pela orientação, dedicação na formação de seus alunos e empenho em fazer do LEG um excelente ambiente de pesquisa.

Aos meus pais, Joaquim e Edila, Luisa e Maria do Carmo (irmãs), Marcos e Júnior (cunhados), Caio, Ana Luisa, Camila e Luana (sobrinhos), pelas orações, carinho e apoio recebido durante este período.

À Terezinha Guedes pela amizade, incentivo, apoio, disponibilidade em me ajudar principalmente na elaboração do pré-projeto de Tese para admissão no curso de pós-graduação e por tudo que aprendi sobre estatística.

Aos professores Terezinha Guedes, Rosângela Getirana Santana e Marcelo Cavalcanti pelas cartas de recomendação enviadas ao PPGMNE.

Às amigas de fé, Alaíde Brita, sua mãe e Denilda, pelas orações.

À amiga Mirian Ferreira pela amizade e paciência durante meus desabafos.

À Edson Antonio Alves da Silva pela amizade, paciência nos meus primeiros passos com Linux, R e geoestatística, e leitura do trabalho com preciosas contribuições.

À Wagner Hugo Bonat, pela paciência, companheirismo e parceria em trabalhos.

À Vanderly Janeiro, Robson Rossi, Gledson Picharski, Bruno Fonseca, Melissa Souza, Prof. Raul Prado, Elias Krainski que contribuíram de alguma forma na elaboração da tese.

Às amigas de curso Bernadete Brandão, Vanessa Ales, Rosângela Villwock e Roberta Suero pelos momentos difíceis e alegres que passamos juntas.

À Maristela pela paciência e carinho com que nos trata.

Aos professores do PPGMNE.

Aos colegas do LEG e CESEC.

Aos colegas do Departamento de Estatística da UEM.

Ao Prof. Antônio Carlos por disponibilizar dados que enriqueceram a pesquisa.

À esposa do Prof. Paulo, Profa. Silvia e seus filhos Paulo Neto e Luca pelos momentos em que o privei de seu convívio familiar.

À CAPES pelo apoio financeiro. Este trabalho foi parcialmente financiado pela FINEP projeto CT-INFRA/UFPR.

Sumário

Lista de Figuras	vii
Lista de Tabelas	xi
Lista de Siglas	xiii
Resumo	xiv
Abstract	xv
1 INTRODUÇÃO	1
2 REVISÃO DA LITERATURA	5
2.1 Modelo Geostatístico Gaussiano	5
2.1.1 Definição do Modelo	5
2.1.2 Componentes do Modelo.....	7
2.1.3 Estimação dos Parâmetros do Modelo	10
2.2 Predição Linear Espacial	15
2.2.1 Conceitos de Predição	15
2.2.2 Krigagem	18
2.3 Inferência Bayesiana Para Predição Espacial	20
2.3.1 Distribuição Preditiva Bayesiana.....	20
2.3.2 Cadeias de Markov.....	22
2.3.3 Avaliação da Convergência	25
2.3.4 Intervalo de Credibilidade.....	26
2.4 Dados Composicionais	27
2.4.1 Composição Regionalizada	29
2.4.2 Representação Gráfica	30

2.4.3	Base Regionalizada	33
2.4.4	Subcomposição Regionalizada	35
2.4.5	Amalgamação e Partição Regionalizada	36
2.4.6	Transformação Logística Modificada	38
2.4.7	Perturbação e Potência	40
2.4.8	Estatísticas Descritivas e Domínio de Confiança	42
2.4.9	Estacionariedade	43
2.4.10	Estrutura de Covariância Espacial	43
2.4.11	Estrutura de Covariância Espacial Intrínscica	45
2.5	Modelo Multivariado	46
2.5.1	Definição do Modelo Multivariado	47
2.5.2	Cokrigagem	49
3	METODOLOGIA PROPOSTA	50
3.1	Modelo Geoestatístico Composicional	51
3.1.1	Estimação dos Parâmetros	52
3.1.2	Incerteza Sobre os Parâmetros	54
3.1.3	Transformação Dos Valores Preditos Para o Espaço Amostral Simplex	56
3.2	Análise de Dados Simulados	60
3.2.1	Conjunto de Dados 1	61
3.2.2	Conjunto de Dados 2	64
3.2.3	Conjunto de Dados 3	67
3.2.4	Conclusões	70
3.2.5	Intervalos de Cobertura	71
3.3	Inferência Bayesiana Para o Modelo Geoestatístico Composicional	73
4	ANÁLISE DE FRAÇÕES GRANULOMÉTRICAS DE UM SOLO ...	76
4.1	Análise Descritiva	76
4.2	Resultados da Modelagem e Predição Geoestatística	81
4.3	Resultados da Análise Bayesiana	90
4.3.1	Avaliação da Convergência	91
5	CONCLUSÕES E SUGESTÕES DE TRABALHOS FUTUROS	98

Referências Bibliográficas	101
Anexo A – Pacote <i>geoComp</i>	108
Anexo B – Help do pacote <i>geoComp</i>	130
geoComp-package	130
agl	131
as.geoComp	132
classificaSolo	133
classSolo	135
cokri.bayes	136
cokrigagem	137
infbayes	139
logarit.vero	142
mec	144
monta.V	147
pivo	149
posteriori	151
resultado.mh	153
tab.coord.pol	154
volta.cokri	155
volta.quad	157
Anexo C – Código fonte R das análises estatísticas	161
Anexo D – Código fonte R dos diagramas ternários	196

Lista de Figuras

Figura 1.1	Triângulo textural.	2
Figura 1.2	Diagrama de classificação textural do solo.	2
Figura 2.1	O ponto p , com coordenadas $(40, 40, 20)\%$ em \mathbb{S}^3 e $(-0, 115; 0, 40)$ em \mathbb{R}^2 , representado no triângulo de altura unitária.	31
Figura 2.2	O ponto p , com coordenadas $(40, 40, 20)\%$ em \mathbb{S}^3 e coordenadas $(0, 60; 0, 55)$ em \mathbb{R}^2 , representado no triângulo com lado unitário e triângulo de classificação textural.	32
Figura 2.3	Pontos pertencentes as mesmas linhas no diagrama ternário com a característica de apresentarem razões iguais dos componentes complementares ao vértice A e ao vértice F	34
Figura 2.4	Diagrama ternário para os dados do Lago Ártico, incluindo o centro da distribuição e região 2—desvios-padrão de confiança.	35
Figura 2.5	Composições de 3 partes como raios partindo da origem em \mathbb{R}_+^3	36
Figura 2.6	Interpretação geométrica da formação da subcomposição $\vec{X}_{12}(\vec{x})$ da composição $\vec{X}(\vec{x})$: (a) em \mathbb{R}_+^3 ; (b) em \mathbb{S}^3	37
Figura 3.1	Distribuição das localizações no quadrado unitário.	60
Figura 3.2	Distribuição de X_1 , X_2 e X_3 e diagrama ternário das composições para a Configuração 1.	61
Figura 3.3	Distribuição das log razões e correspondente diagrama de dispersão para a Configuração 1.	62
Figura 3.4	Mapas das percentagens de X_1 , X_2 e X_3 para dados da Configuração 1.	64
Figura 3.5	Valores observados versus preditos de X_1 , X_2 e X_3 para a Configuração 1.	64

Figura 3.6	Distribuição de X_1 , X_2 e X_3 e diagrama ternário das composições para a Configuração 2.	65
Figura 3.7	Distribuição das log razões e correspondente diagrama de dispersão para a Configuração 2.	65
Figura 3.8	Mapas das percentagens de X_1 , X_2 e X_3 para dados da Configuração 2.	66
Figura 3.9	Valores observados versus preditos de X_1 , X_2 e X_3 para a Configuração 2.	67
Figura 3.10	Distribuição de X_1 , X_2 e X_3 e diagrama ternário das composições para Configuração 3.	68
Figura 3.11	Distribuição das log razões e correspondente diagrama de dispersão para a Configuração 3.	68
Figura 3.12	Mapas das percentagens de X_1 , X_2 e X_3 para dados da Configuração 3.	69
Figura 3.13	Valores observados versus preditos de X_1 , X_2 e X_3 para a Configuração 3.	70
Figura 3.14	Distribuições das estimativas dos parâmetros para 1000 simulações e amostras de tamanho 300.	72
Figura 3.15	Boxplot das estimativas dos parâmetros para 1000 simulações e amostras de tamanho 300.	72
Figura 4.1	Foto aérea do campo experimental de irrigação da ESALQ-USP com área de estudo correspondente ao quadrante irrigado por um sistema pivô-central.	76
Figura 4.2	Percentagem de areia (à esquerda), silte (meio) e argila (à direita) segundo os 25% (círculos), 25 a 50% (triângulos), 50 a 75% (diagonais verticais), 75 a 100% (diagonais cruzadas) menores valores e a localização amostral.	77
Figura 4.3	Diagrama de classificação textural.	77
Figura 4.4	Boxplot dos percentuais de areia, silte e argila.	78
Figura 4.5	Distribuição dos percentuais de areia, silte e argila e diagrama ternário	

das composições.	79
Figura 4.6 Distribuição e diagrama de dispersão das log razões.	80
Figura 4.7 Mapas das predições das percentagens de areia, silte e argila obtidas por quadratura de Gauss-Hermite (A-C) e por simulação (D-F).	82
Figura 4.8 Comparação da frações de areia, silte e argila obtidos por quadratura de Gauss-Hermite e simulação.	82
Figura 4.9 Comparação dos dados do pivô com os valores preditos por quadratura de Gauss-Hermite e por simulação.	83
Figura 4.10 Comparação da frações de areia, silte e argila obtidos por quadratura de Gauss-Hermite e simulação.	84
Figura 4.11 Distribuição das frações máximas (1ª linha) e mínimas (2ª linha) na área, por componente.	85
Figura 4.12 Frações máximas e mínimas por componente, obtidas em 1000 simulações para cada localização.	85
Figura 4.13 Frações medianas, inferiores a 5% e 95% de probabilidade, por componente, obtidos em 1000 simulações para cada localização.	87
Figura 4.14 Probabilidades <i>a posteriori</i> de classificação nas diferentes classes de solo.	88
Figura 4.15 Log-verossimilhanças perfilhadas para ϕ e ρ	89
Figura 4.16 Deviance para ϕ , com linhas de corte de 83,5% e 56,6% de confiança (esquerda) e para ρ , com linhas de corte de 95% de confiança (direita).	89
Figura 4.17 Comparação das densidades de areia (à esquerda), silte (centro) e argila (à direita) obtidos por inferência bayesiana e não-bayesiana.	91
Figura 4.18 Valores preditos de areia (à esquerda), silte (centro) e argila (à direita) por inferência bayesiana.	91
Figura 4.19 Autocorrelações dos parâmetros <i>a posteriori</i>	92

Figura 4.20 Trajetória das <i>posteriori</i> dos parâmetros.	93
Figura 4.21 Densidades estimadas para <i>as posteriori</i> dos parâmetros com respectivos intervalos de alta densidade <i>a posteriori</i> a 95%.	94
Figura 4.22 Diagnóstico de convergência para o Teste de Geweke.	96
Figura 4.23 Diagnóstico de convergência para o Teste de Gelman e Rubin.	97

Lista de Tabelas

Tabela 2.1	Pontos nas linhas que partem do vértice A com respectivas coordenadas em \mathbb{S}^3 , \mathbb{R}^2 e razão dos componentes X_2 e X_3	33
Tabela 2.2	Pontos nas linhas que partem do vértice F com respectivas coordenadas em \mathbb{S}^3 , \mathbb{R}^2 e razão dos componentes X_1 e X_3	33
Tabela 3.1	Estimativas, erros padrão e intervalos de confiança para a Configuração 1, pelo método Delta via método de otimização “L-BFGS-B”.	63
Tabela 3.2	Estimativas, erros padrão e intervalos de confiança para a Configuração 2, pelo método Delta via método de otimização “L-BFGS-B”.	66
Tabela 3.3	Estimativas, erros padrão e intervalos de confiança para a Configuração 3, pelo método Delta via método de otimização “L-BFGS-B”.	69
Tabela 3.4	Parâmetros, média e intervalo de 95% de confiança das estimativas dos parâmetros, e cobertura (%) dos intervalos via método Delta para 1000 simulações de amostras de tamanho 300.	71
Tabela 4.1	Estatísticas descritivas dos percentuais de areia, silte e argila.	78
Tabela 4.2	Estimativas, erros-padrão e limites inferior e superior dos intervalos de confiança para os parâmetros do modelo, obtidos pelo método Delta via, método de otimização “L-BFGS-B”.	81
Tabela 4.3	Médias das frações máximas e mínimas de areia, silte e argila obtidas nas 1000 simulações de Gauss-Hermite.	86
Tabela 4.4	Média dos valores da distribuição <i>a posteriori</i> de $\vec{\theta}$ e intervalos de 95% de credibilidade.	90
Tabela 4.5	Autocorrelações das <i>posteriori</i> dos parâmetros.	92
Tabela 4.6	Correlações das <i>posteriori</i> dos parâmetros.	93

Tabela 4.7	Limites dos intervalos de 95% de credibilidade de alta densidade <i>a posteriori</i>	94
Tabela 4.8	Valores de z_G do teste de Geweke para <i>as posteriori</i> dos parâmetros. . .	95
Tabela 4.9	Fatores de redução de escala potencial do teste de Gelman e Rubin. . .	95

Lista de Siglas

- ALR - Additive logratio (correponde à transformação razão log-aditiva);
- AGL - Additive generalized logistic (corresponde à transformação logística generalizada aditiva);
- CLR - Center logratio (correponde à transformação razão log-centrada);
- EM - Erro de Medida;
- EQM - Erro (de Predição) Quadrático Médio;
- ESALQ - Escola Superior de Agricultura " Luiz de Queiroz";
- LR - Log-Razão;
- MCMC - Monte Carlo via cadeia de Markov (em inglês: Monte Carlo Markov Chain);
- SQR - Soma de Quadrados dos Resíduos;
- USP - Universidade de São Paulo;
- VME - Variação de Micro Escala (ou Pequena).

Resumo

Dados composicionais são vetores com elementos positivos e soma constante, geralmente 1 ou 100%. Essa restrição define o simplex unitário como o espaço amostral, induz correlação intrínseca entre as variáveis e impõe limitações à aplicação de técnicas estatísticas usuais para a análise e modelagem de dados. Essa estrutura de dados aparece em diversas áreas. Nas Ciências Agrárias, é usual expressar as frações granulométricas de solo como conteúdos de areia, silte e argila e usá-las para descrever e definir classes de solo. Os padrões da distribuição espacial das frações do solo são relevantes na compreensão do sistema agrícola auxiliando na determinação de práticas agrícolas, em particular, na agricultura de precisão que visa fornecer recursos para as culturas a taxas variáveis ao longo do espaço. A análise das frações, individualmente, dificulta a garantia da restrição unitária e exige ajustes. A análise de dados composicionais considera essa restrição ao modelar as frações conjuntamente. As (k) variáveis originais, proporções dos componentes, são transformadas em $(k - 1)$ variáveis para as quais supõe-se uma distribuição gaussiana multivariada. Os métodos foram desenvolvidos supondo a independência entre as observações das localizações amostrais. As extensões para o contexto espacial exige a acomodação de duas fontes de associação na matriz de covariância, a correlação intrínseca entre as frações e a correção regionalizada induzida pelos padrões espaciais nas proporções dos componentes variando suavemente na área. Sob o paradigma bayesiano, a estrutura de covariância pode ser especificada por uma distribuição *a priori* Wishart para a matriz de covariância. Outro procedimento, em geoestatística, é relacionar correlações à distância de separação entre os pontos amostrais utilizando funções de correlação. Essa estratégia foi adotada especificando-se um modelo geoestatístico bivariado paramétrico para dados composicionais transformados, supondo termos latentes comuns caracterizados por uma função de correlação induzindo a estrutura espacial, e termos correlacionados acomodando correlações composicionais intrínsecas. Métodos de inferência baseados na verossimilhança e bayesianos e predição espacial foram deduzidos e implementados. Interpolação espacial e predição do modelo foram deduzidas dos resultados da distribuição normal multivariada. Os dados foram transformados para o espaço amostral simplex considerando-se integração de Gauss-Hermite e simulação. Três conjuntos de dados simulados diferentes e um conjunto de dados de frações de solo em uma área irrigada por um sistema pivô-central ilustraram a análise. Foram utilizados a transformação log-razão aditiva para transformar as proporções originais e a função de correlação exponencial. Na inferência bayesiana foram consideradas diferentes priores para os parâmetros. Os resultados incluem mapas de predição na escala original das proporções esperadas de areia, silte e argila sobre a área, bem como outros objetivos de predição como mapas de quantis, de probabilidades de exceder um determinado patamar e a distribuição *a posteriori* das classes de solo. As rotinas computacionais foram organizadas num protótipo de pacote-R chamado *geoComp*.

Palavras-chave: Geoestatística, dados composicionais, inferência bayesiana, verossimilhança.

Abstract

Compositional data are vector with positive elements adding to a constant, typically 1 or 100%. Such restriction defines the unit simplex as the sampling space, induces intrinsic correlations among variables and places restrictions on the application of standard statistical procedures for data analysis and modelling. This data format appears in a range of different areas. In agronomical sciences, soil fractions are typically expressed as contents of sand, clay and silt and used describing properties and defining classes of texture. Patterns of the spatial distribution of the soil fractions are relevant for characterizing the agricultural system and are accounted for when defining management practices, in particular for precision agriculture which aims to provide necessary resources for crops at possibly varying rates over the space. Independent analysis of individual fractions, although often used, creates difficulties ensuring the unit restriction and requires adjustments. Compositional data analysis accounts for the unit restriction by modelling the fractions jointly. Typically, original (k) variables representing the proportions of the components are transformed in a set of $(k - 1)$ variables for which a multivariate Gaussian distribution is assumed. Methods and models were originally developed assuming independence between observations of the fractions across the sampling sites. Extensions for the spatial context requires accommodating two sources of associations in the resulting covariance matrix, the intrinsic correlation between the fractions and the correlogram induced by spatial patterns in the possibly smoothly varying proportions of the components within the area. For instance, under the Bayesian paradigm, covariance structure can be specified by a Wishart *prior* for the covariance matrix. Another possible approach usual in geostatistical modelling is to relate correlations to separation distance between sampling points by assuming correlation functions of particular forms. We follow the latter strategy assuming explicitly a parametric multivariate geostatistical model. In particular, we specify a bivariate model for the transformed compositional data assuming common latent terms characterised by a correlation function inducing the spatial structure and correlated terms accommodating the intrinsic compositional correlations. Likelihood based and Bayesian methods of inference and spatial prediction are derived and implemented. Spatial interpolation and prediction from the model are easily derived for the transformed variables for back-transforming for the original simplex space we consider Gauss-Hermite integration an alternatively by simulation. Three sets of simulated data with different patterns of spread of the compositions and a data set on soil fractions in an area irrigated by a central pivot system are used to illustrate the analysis. In our application example we use the additive log-ratio to transform the original proportions and the exponential correlation function for the latent spatial term in the model. Three different sets of priors were considered within the Bayesian inference. Results include prediction maps on the original scale of the expected proportions of sand, clay and silt over the area, as well as other prediction targets such as maps of quantiles, probabilities of exceedance and the posterior distribution of the soil classes. Routines for the analysis are organised in a prototype R-package called *geoComp*.

Key-words: Geostatistics, compositional data, bayesian inference, likelihood.

1 INTRODUÇÃO

Dados composicionais são definidos como vetores de elementos positivos e soma constante, geralmente 1 ou 100%.

Nas ciências agrárias, os diversos processos naturais relacionados ao desenvolvimento e à produção vegetal dependem, em grande extensão, de características do solo relacionadas com as capacidades de armazenar e infiltrar água, reter e disponibilizar nutrientes, com a resistência mecânica à penetração e com a capacidade de realizar trocas gasosas no seu espaço poroso devido à sua amplitude e geometria. Além dessas características fundamentais do solo, diversas outras podem exercer influência nesses processos, e todas estão intimamente relacionadas com a proporção de seus elementos constituintes. Essa composição caracteriza a textura do solo, expressa por meio da proporção relativa das frações granulométricas, tradicionalmente caracterizadas como areia, silte e argila. Por ser uma característica física de grande importância e pouco susceptível a mudanças ao longo do tempo, a análise granulométrica é uma das primeiras e principais ações a serem conduzidas na inferência a respeito das futuras estratégias de manejo. Assim, o conhecimento da proporção segundo a qual as frações granulométricas se apresentam no solo contribui, inclusive, para a sua classificação, tornando-se uma informação de relevância para o sistema agrícola.

Com o aperfeiçoamento das técnicas de produção, o avanço do conhecimento científico em torno do agrossistema e o aprimoramento das tecnologias envolvidas, notadamente em relação ao maquinário empregado, tem-se buscado, frequentemente, a compreensão de que o sistema agrícola apresenta uma distribuição espacial heterogênea das suas características que não pode ser ignorada.

Descrever a distribuição espacial dos valores das frações granulométricas do solo consiste em uma base de importância primordial para a descrição espacial das suas características, relevantes, quando se busca a otimização do processo de produção para se atingir as metas fixadas por conceitos ambientais e econômicos.

Como se trata de uma proporção entre as frações granulométricas, a análise textural do solo sempre resulta no total de 100% para a soma desses componentes. Essa soma pode ser representada por um ponto P , $P = (80, 10, 10)$ por exemplo, correspondente aos valores percentuais de areia, silte e argila, no interior de um triângulo formado pelos três elementos (Figura 1.1), usado na classificação textural do solo.

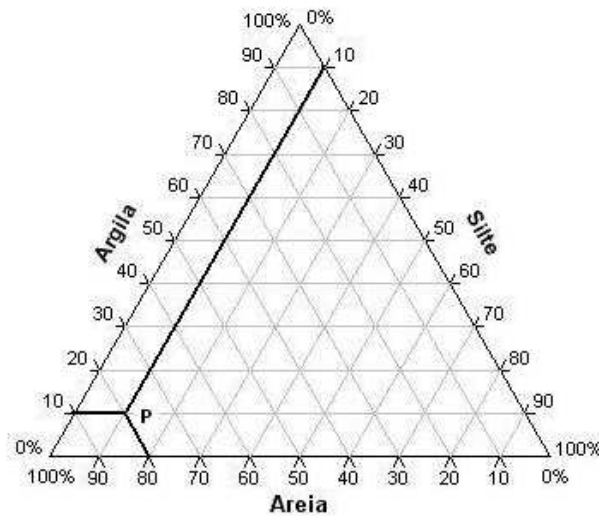


Figura 1.1: Triângulo textural.

Esse triângulo é dividido em áreas (Lemos e Santos, 1996, *apud* Reichardt e Timm, 2004) e, de acordo com a localização do ponto nessas áreas, tem-se uma classificação para o solo (Figura 1.2).

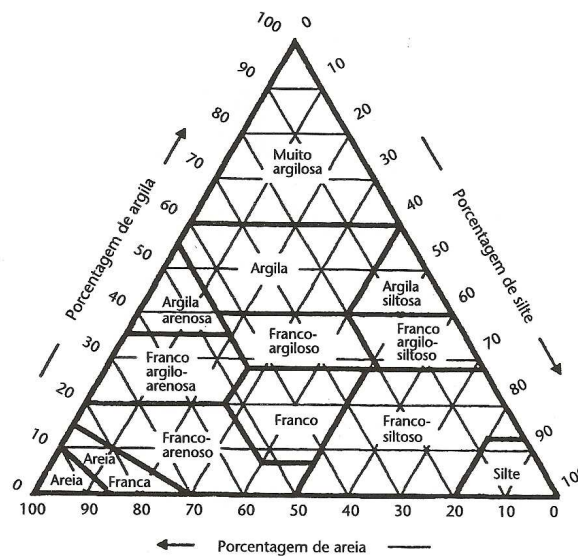


Figura 1.2: Diagrama de classificação textural do solo.

FONTE: Lemos e Santos (1996) *apud* Reichardt e Timm (2004).

A partir dessa abordagem, a descrição da distribuição espacial das três frações granulométricas pode ser feita levando-se em conta a relação entre elas, enriquecendo potencialmente a análise.

A Estatística Espacial tem se apresentado como uma área de grande importância nas mais diversas aplicações. Trabalhos de grande relevância têm sido desenvolvidos nessa área, como os de Matheron (1963), Cressie (1993), Bailey e Gatrell (1995), Banerjee, Carlin e Gelfand (2004), Schabenberger e Gotway (2005), etc. Em ciências agrárias, a Estatística Espacial é aplicada no contexto da denominada “Agricultura de Precisão”.

Citam-se, por exemplo, os trabalhos de Molin (1997a), Molin (1997b), Lake, Bock e Goode (1997), Srinivasan (2006), dentre outros. Na área de geoestatística com enfoque na agricultura de precisão, cita-se o trabalho de Silva (2000). Surgiram, ainda, trabalhos em geoestatística como os de Diggle, Tawn e Moyeed (1998), Schabenberger e Pierce (2001), Diggle e Ribeiro Jr. (2007) e Silva (2008), que diferem da linha tradicional no sentido de que a análise é baseada em modelos que induzem uma estrutura de covariância. Nesses modelos, é possível aplicar métodos clássicos de inferência baseados em verossimilhança que produzem estimativas mais eficientes dos parâmetros e permitem avaliar a incerteza das predições espaciais. Estudos como o de Gonçalves (1997), Gonçalves, Folegatti e Mata (2001), Eguchi, Silva e Oliveira (2002), por exemplo, buscam a descrição da distribuição espacial das frações de areia, silte e argila do solo, de forma independente, utilizando métodos geoestatísticos.

Trabalhos realizados por Aitchison (1982, 1986 e 1999) em análise de dados composicionais apresentam uma metodologia adequada para analisar dados caracterizados por se apresentarem em forma de proporções complementares. A partir dos anos 2000, estes tipos de dados são analisados considerando-se as localizações amostrais, mas ainda sob a abordagem geoestatística tradicional. Por exemplo, Odeh, Tood e Triantafyllis (2003) mostram que, sem considerar que a matriz de covariância do modelo composicional seja definida positiva e que os valores interpolados satisfaçam a restrição “soma um”, a interpolação espacial de dados de frações de partículas de solo produzem incerteza e valores interpolados irreais. Pawlowsky-Glahn e Olea (2004) satisfazem essas exigências, mas não adotam declaração explícita de modelo, no sentido de não adotarem um modelo paramétrico para os dados e, não consideram predição espacial bayesiana. Lark e Bishop (2007) fazem um estudo sobre cokrigagem de frações de partículas do solo concluindo que a predição pode ser feita através da cokrigagem razão log-aditiva (ALR). Segundo esses autores, a cokrigagem ALR, que considera o logaritmo das razões dos componentes, apresenta vantagens em relação à cokrigagem sem transformação, que considera apenas as razões dos componentes. Concluem ainda que existem vantagens se a transformação de volta das predições para a escala original, das composições, são calculadas por quadratura de Gauss-Hermite para aproximar a esperança condicional. Tjelmeland e Lund (2003) modelam dados composicionais espaciais com metodologia bayesiana sem adotar forma explícita para a função de covariância e sem fazer predição espacial. Obage (2007) faz inferência bayesiana para dados composicionais excluindo o contexto espacial, considerando as transformações ALR e Box-Cox e, assumindo uma distribuição gaussiana multivariada para erros correlacionados.

O objetivo geral deste trabalho foi propor e implementar um modelo geoestatístico para dados espaciais composicionais utilizando estruturas multivariadas, com componentes do modelo especificados por função de correlação espacial e “composicional”, e desenvolver métodos de inferência baseadas na verossimilhança de forma usual e sob o enfoque bayesiano.

Os objetivos específicos do trabalho foram:

- desenvolver uma forma alternativa à proposta de Pawlowsky-Glahn e Olea (2004), baseada na declaração explícita de modelos;
- construir um modelo em que a dependência espacial e entre as variáveis que formam a composição seja considerada na obtenção de uma função de covariância válida;

- deduzir métodos de inferência clássica, baseada na verossimilhança, para a estimação dos parâmetros do modelo;
- aplicar métodos de inferência bayesiana para a estimação dos parâmetros do modelo;
- desenvolver rotinas computacionais para análise de dados composicionais no espaço;
- aplicar a metodologia proposta em um conjunto de dados de solo, elaborando mapas temáticos de modelos composicionais.

O trabalho está dividido em cinco capítulos. No Capítulo 2, apresenta-se uma revisão de literatura sobre geoestatística, inferência bayesiana e teoria de dados composicionais. No Capítulo 3, propõe-se o modelo geoestatístico bivariado para dados composicionais, desenvolve-se a teoria para a estimação dos parâmetros pelo método de inferência clássico e aplica-se a metodologia proposta em três conjuntos de dados composicionais simulados; apresenta-se um estudo por simulação para a determinação dos intervalos de cobertura dos parâmetros do modelo e desenvolve-se a teoria para a estimação dos parâmetros sob o paradigma de inferência bayesiana. No Capítulo 4, apresenta-se os resultados da aplicação da metodologia proposta tanto pelo método clássico quanto por inferência bayesiana, em um conjunto de dados reais de areia, silte e argila. O Capítulo 5 finaliza o trabalho com a conclusão e sugestões de trabalhos futuros. O Anexo A apresenta as funções do pacote estatístico *geoComp* desenvolvido para este trabalho. O Anexo B contém o manual desse pacote e, no Anexo C, encontram-se os códigos fonte R das análises realizadas. No Anexo D, são apresentados os códigos fonte R da construção dos diagramas ternários.

2 REVISÃO DA LITERATURA

2.1 Modelo Geostatístico Gaussiano

2.1.1 Definição do Modelo

Hoel, Port e Stone (1972) definem um processo estocástico como qualquer coleção de variáveis aleatórias $S(t)$, $t \in T$, definida num espaço de probabilidade comum, em que t é um subconjunto de $(-\infty, +\infty)$ e geralmente considerado como parâmetro fixado. Desta forma, um processo estocástico espacial pode ser definido como qualquer coleção de variáveis aleatórias $S(\vec{x})$, $\vec{x} \in \mathbb{R}^d$, definidas num espaço de probabilidade comum, no qual d é o número de linhas do vetor \vec{x} cujos elementos são as localizações amostrais. A caracterização das propriedades de processos estocásticos requer uma forma de expressar a distribuição conjunta das variáveis aleatórias $S(\vec{x}_1), S(\vec{x}_2), \dots, S(\vec{x}_n)$ (FALEIROS; YONEYAMA, 2002).

Em particular, para $d = 2$, um *processo espacial gaussiano* $S = (S(\vec{x}) : \vec{x} \in \mathbb{R}^2)$ é um processo estocástico com a propriedade de que, para qualquer coleção de localizações $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$, com $\vec{x}_i \in \mathbb{R}^2$, $S = (S(\vec{x}_1), S(\vec{x}_2), \dots, S(\vec{x}_n))$ tem distribuição conjunta gaussiana multivariada, completamente especificada pela função média $\mu(\vec{x})$, e função de covariância $Cov(S(\vec{x}_i), S(\vec{x}_j))$, $i, j = 1, \dots, n$. (DIGGLE; RIBEIRO JR., 2007). Schabenberger e Pierce (2001) destacam que a função de distribuição acumulada é aquela de uma distribuição gaussiana n -variada

$$F(S(\vec{x})) = P(S(\vec{x}_1) < s_1, \dots, S(\vec{x}_n) < s_n).$$

Segundo Diggle, Ribeiro Jr. e Christensen (2003), a geoestatística é um ramo da estatística espacial na qual os dados consistem de mensurações y_1, y_2, \dots, y_n obtidas nas localizações $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$ amostradas em uma região $A \subset \mathbb{R}^2$ espacialmente contínua. As mensurações estão relacionadas a um fenômeno espacial que pode ser tratado como a realização de um processo estocástico $S(\vec{x}); \vec{x} \in \mathbb{R}^2$, denominado *sinál* que, em geral, não é diretamente observável.

De acordo com Matérn (1960) *apud* Diggle e Ribeiro Jr. (2007), a teoria de amostragem espacial mostra que, sob suposições de modelagem, as propriedades espaciais

são mais eficientemente estimadas por um delineamento regular, no qual as localizações formam uma grade regular sobre a região em estudo. Além disso, delineamentos em grade são geralmente eficientes para predição espacial. Se não existir relação entre a escolha da coordenada \vec{x} e o atributo y , que é o caso suposto neste trabalho, a amostragem é denominada *não preferencial*; caso contrário, é denominada *preferencial* (DIGGLE; MENEZES; SU, 2010).

A terminologia “geoestatística baseada em modelos” foi introduzida por Diggle, Tawn e Moyeed (1998). É caracterizada pela declaração explícita e completa de um modelo, o que significa que a modelagem é realizada adotando-se um modelo paramétrico para a(s) variável(veis) resposta, e a utilização de métodos de inferência estatística baseadas na verossimilhança aplicada a problemas geoestatísticos. Assim, um modelo geoestatístico é a especificação da distribuição conjunta $[Y(\vec{x}), S(\vec{x})]$, usualmente apresentada na forma fatorada

$$[S(\vec{x}), Y(\vec{x})] = [S(\vec{x})][Y(\vec{x})|S(\vec{x})].$$

Em particular, esse modelo não especifica o delineamento amostral, ou seja, a distribuição de $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n\}$ que será suposta independente de $S(\vec{x})$ e de $Y(\vec{x})$.

De acordo com Diggle, Ribeiro Jr. e Christensen (2003), no modelo geoestatístico gaussiano estacionário e *isotrópico* (cujas definições são apresentadas na Subseção 2.1.2), $S(\vec{x})$ é um processo gaussiano estacionário, com $E(S(\vec{x}_i)) = \mu$; $Var(S(\vec{x}_i)) = \sigma^2$, e função de correlação $\rho(u_{ij}) = Corr(S(\vec{x}_i), S(\vec{x}_j))$ em que $u_{ij} = \|\vec{x}_i - \vec{x}_j\|$, $i, j = 1, \dots, n$, $\rho(u) = \rho(-u)$ e $Y(\vec{x}_i)$ o valor observado na localização \vec{x}_i .

Esses autores consideram plausível um modelo geoestatístico em que a distribuição de $Y(\vec{x}_i)$, $i = 1, 2, \dots, n$, condicionada à distribuição $S(\cdot)$, é gaussiana com média $S(\vec{x}_i)$ e variância τ^2 ; e que $Y(\vec{x}_i)$ são mutuamente independentes condicionados em $S(\cdot)$. O modelo pode, então, ser escrito na forma

$$Y(\vec{x}_i) = S(\vec{x}_i) + Z(\vec{x}_i) \quad i = 1, \dots, n, \quad (2.1)$$

em que $Z(\vec{x}_i) \sim N(0; \tau^2)$ são erros aleatórios independentes de $S(\vec{x})$ e o valor observado y_i é uma realização de $Y(\vec{x}_i)$ que é uma versão ruidosa de $S(\vec{x}_i)$ no caso gaussiano. Integrando-se a distribuição conjunta de $Y(\vec{x})$ e $S(\vec{x})$ com relação a $S(\vec{x})$, obtém-se a distribuição marginal de $Y(\vec{x})$

$$\vec{Y}(\vec{x}) \sim N_n(\mu \vec{1}; \sigma^2 \mathbf{R} + \tau^2 \mathbf{I}),$$

em que $\vec{1}$ é um vetor com n elementos iguais a 1, \mathbf{R} é uma matriz de ordem $n \times n$ cujos elementos são as correlações $\rho(u_{ij})$ e \mathbf{I} é a matriz identidade de ordem $n \times n$.

Com a reparametrização $v^2 = \frac{Var(S(\vec{x}_i))}{Var(Z(\vec{x}_i))} = \frac{\tau^2}{\sigma^2}$, segue-se que:

$$Var(\vec{Y}(\vec{x})) = \sigma^2 \mathbf{R} + \tau^2 \mathbf{I} = \sigma^2 \left(\mathbf{R} + \frac{\tau^2}{\sigma^2} \mathbf{I} \right) = \sigma^2 \mathbf{V},$$

em que

$$\mathbf{V} = \mathbf{R} + v^2\mathbf{I}, \quad (2.2)$$

e a distribuição de $\vec{Y}(\vec{x})$ é escrita na forma

$$\vec{Y}(\vec{x}) \sim N_n(\mu\vec{1}; \sigma^2\mathbf{V}).$$

2.1.2 Componentes do Modelo

Segundo Schabenberger e Gotway (2005), um processo estocástico $S(\vec{x})$ é estacionário se sua distribuição espacial é invariante sob translação das coordenadas. Schabenberger e Pierce (2001) afirmam que, geometricamente, isto implica que a distribuição espacial é invariante sob rotação e estiramento do sistema de localização das amostras. Bailey e Gatrell (1995) mostram que isto acontece se as propriedades estatísticas $S(\vec{x})$ são independentes da localização absoluta na região de estudo $A \subset \mathbb{R}^2$. Isto implica que a média e a variância são constantes na região A e não dependem da localização \vec{x} . Implica também que a covariância $Cov(S(\vec{x}_i), S(\vec{x}_j))$, $i \neq j$, dependerá somente das localizações relativas destes dois pontos, da distância u que as separa e da direção entre elas, e não de sua localização absoluta na região A . Desta forma, este processo pode ser pensado como o equivalente espacial de uma amostra aleatória em estatística clássica que dá origem a variáveis aleatórias independentes com a mesma média e dispersão (SCHABENBERGER; PIERCE, 2001).

Ainda segundo estes autores, a função covariância é denominada isotrópica na ausência de dependência da direção, ou melhor, quando depende somente da distância absoluta (neste trabalho a distância euclidiana) entre os pares de pontos. O processo espacial é isotrópico se, em acréscimo à estacionariedade, a covariância depender somente da distância entre dois pontos, e não da direção nos quais estão separados (BAILEY; GATRELL, 1995).

De acordo com Diggle, Ribeiro Jr. e Christensen (2003), a especificação da função de correlação, $\rho(u)$, determina a suavidade do processo $S(\vec{x})$. A descrição matemática formal da suavidade de uma superfície espacial é dada por seu grau de diferenciabilidade. $S(\vec{x})$ é quadrado-médio contínuo se $\lim_{u \rightarrow 0} E(\{S(\vec{x}_i) - S(\vec{x}_j)\}^2) = 0$ para todo \vec{x} . Da mesma forma, é quadrado-médio diferenciável se existe um processo $S'(\vec{x})$ tal que

$$\lim_{u \rightarrow 0} E \left(\left\{ \frac{S(\vec{x}_i) - S(\vec{x}_j)}{u} - S'(\vec{x}_i) \right\}^2 \right) = 0.$$

Então, a diferenciabilidade quadrado-médio de $S(\vec{x})$ está diretamente relacionada com a diferenciabilidade de sua função de covariância através do resultado que diz que se

$S(\vec{x})$ é um processo gaussiano estacionário com função de correlação $\rho(u)$, $u \in \mathbb{R}$, então $S(\vec{x})$ é quadrado-médio contínuo se, e somente se, $\rho(u)$ é contínua em $u = 0$; e $S(\vec{x})$ é k vezes quadrado-médio diferenciável se, e somente se, $\rho(u)$ é ao menos $2k$ vezes diferenciável em $u = 0$. A demonstração deste resultado pode ser encontrada em Stein (1999).

Na estrutura do modelo proposto, a *família Matérn* de funções de correlação, apresentada em Diggle e Ribeiro Jr. (2007), é uma importante função paramétrica de correlação com expressão analítica dada por

$$\rho(u, k, \phi) = \frac{1}{2^{k-1}\Gamma(k)} \left(\frac{u}{\phi}\right)^k K_k\left(\frac{u}{\phi}\right),$$

em que $K_k(\cdot)$ denota a função de Bessel modificada de ordem k , $\phi > 0$ é um parâmetro de escala, associado à extensão de dependência espacial (alcance) e $k > 0$ é um parâmetro de forma que determina a suavidade analítica do processo $S(\vec{x})$, interpretado como uma medida da diferenciabilidade do processo. Especificamente, $S(\vec{x})$ é $[k - 1]$ vezes quadrado médio diferenciável, onde o símbolo “[$k - 1$]” é denominado *ceiling* e “[k]” significa o menor inteiro maior ou igual a k . Devido à dificuldade de identificação de todos os parâmetros do modelo, na prática, os valores de k podem ser escolhidos dentre os três valores $\{0,5; 1,5; 2,5\}$ correspondendo, respectivamente, à não diferenciabilidade e a processos estocásticos uma e duas vezes diferenciáveis na origem. Assim, para diferentes valores de k , pode-se obter diferentes valores de amplitude prática do modelo, que é a distância u_0 no qual $\rho(u_0) = \alpha$, onde α é um valor tão pequeno quanto o pesquisador determinar, usualmente 0,05. Neste caso, para o conjunto estabelecido acima, os valores de u_0 são aproximadamente $\{3\phi; 4,75\phi; 5,92\phi\}$, respectivamente.

Nessa família Matérn, fazendo $k = 0,5$, obtém-se a *função de correlação exponencial*

$$\rho(u) = \exp\left(-\frac{u}{\phi}\right), \quad u_0 = 3\phi \quad \text{e} \quad \lim_{k \rightarrow \infty} \rho(u) = \exp\left\{-\left(\frac{u}{\phi}\right)^2\right\},$$

obtendo-se a função de correlação gaussiana para a qual $u_0 \simeq \sqrt{3}\phi$.

Em textos de geoestatística tais como Isaaks e Srisvastava (1989), Kitanidis (1997), Goovaerts (1997, Seção 4.2.3), Chilès e Delfiner (1999), Schabenberger e Pierce (2001), Diggle e Ribeiro Jr. (2007) e outros, pode-se encontrar outras funções de correlação. Neste trabalho, será adotada a função de correlação exponencial.

A função de correlação, por sua vez, está associada à matriz de covariância. Isaaks e Srisvastava (1989) afirmam que a matriz de covariância deve ser definida positiva como uma garantia de que a variância de qualquer variável aleatória formada pela combinação linear ponderada de outras variáveis aleatórias será positiva.

Quando não existe estacionariedade na média, a situação mais comum é que $\mu(\vec{x})$, denominada *superfície de tendência*, seja escrita como um modelo de regressão polinomial usando potências e produtos cruzados das coordenadas cartesianas de \vec{x} como variáveis explicativas. Diggle e Ribeiro Jr. (2007) afirmam que superfícies de tendência linear e quadrática podem fornecer descritores empíricos úteis da tendência espacial não explicada, mas superfícies de ordem maior devem ser evitadas porque tendências mais complexas podem ser melhor descritas através do componente estocástico do modelo.

A especificação de $\mu(\vec{x})$ pode ser feita também em função de outras variáveis explicativas e, neste caso, a média se associa a uma “tendência externa”.

Uma superfície de tendência linear pode ser escrita como

$$\mu(\vec{x}) = \beta_0 + \sum_{j=1}^p \beta_j d_j(\vec{x}), \quad (2.3)$$

em que $d_j(\vec{x})$ são variáveis explicativas espaciais, dependentes ou não das coordenadas. Neste trabalho, será considerada uma superfície de tendência constante ($\mu(\vec{x}) = \mu$), embora uma superfície de tendência linear não acrescente complexidade à formulação do modelo.

Outro componente implícito do modelo é o “efeito pepita”, que é um termo usado para representar a variância τ^2 do erro aleatório $Z(\vec{x}_i)$ em (2.1). Essa variância pode ser dividida em dois componentes como

$$\tau^2 = \text{EM} + \text{VME} \quad (2.4)$$

em que EM é o erro de medida e VME é a variação de micro escala (pequena). VME representa uma variação não capturada pelo processo $S(\vec{x})$ e que ocorre em distâncias menores do que a menor distância observada entre duas localizações. Em muitas situações, a distinção dos dois componentes não é possível mas, conforme Equação (2.4), pode-se determinar o valor de VME se o valor de EM for conhecido. No entanto, isso só será possível se houver mais de uma observação de Y na mesma localização. Neste trabalho o efeito pepita não será considerado.

Quando a variável observada $Y(\vec{x})$ não segue uma distribuição gaussiana ou em processos de variância não estacionária, o valor do erro quadrático médio mínimo do preditor é afetado, com a consequência de se obter aproximações ruins. Em muitos casos, através de transformações, é possível que os dados passem a seguir uma distribuição gaussiana.

Box e Cox (1964) propõem a família de transformações

$$Y^*(\vec{x}) = \begin{cases} \frac{Y^\lambda(\vec{x}) - 1}{\lambda} & , \lambda \neq 0; \\ \log Y(\vec{x}) & , \lambda = 0, \end{cases}$$

em que o parâmetro λ introduz flexibilidade ao modelo. Alguns valores interpretáveis para esse parâmetro são:

$$\begin{aligned} \lambda = 1,0 & : \text{ sem transformação;} \\ \lambda = 0,5 & : \text{ transformação raiz quadrada;} \\ \lambda = -1,0 & : \text{ transformação recíproco.} \end{aligned}$$

Pode-se, ainda, modelar explicitamente os dados $Y(\vec{x})$ com outras famílias de distribuições e, neste caso, Diggle, Tawn e Moyeed (1998) propõem modelos para distribuições

não gaussianas de Y na família exponencial.

2.1.3 Estimação dos Parâmetros do Modelo

Os métodos usuais de estimação dos parâmetros de um modelo gaussiano linear são o método dos mínimos quadrados e o método da máxima verossimilhança.

De acordo com Diggle e Ribeiro Jr. (2007), na estimação dos parâmetros da superfície de tendência linear pelo método dos mínimos quadrados, a média do modelo, $E(Y(\vec{x}))$, é suposta como a Equação (2.3), em que $d_j(\vec{x})$, $j = 1, \dots, p$ são covariáveis espaciais e β_j , $j = 0, \dots, p$ são os parâmetros da regressão linear.

Usando o método dos mínimos quadrados ordinários, as estimativas $\tilde{\beta}_j$ são aquelas que minimizam a soma dos quadrados dos resíduos (SQR). Para o modelo dado pela Equação (2.1), incluída a tendência,

$$Y(\vec{x}_i) = \mu(\vec{x}_i) + S(\vec{x}_i) + Z(\vec{x}_i) \quad i = 1, \dots, n, \quad (2.5)$$

a SQR é dada por

$$\text{SQR}(\beta) = \sum_{i=1}^n Z^2(\vec{x}_i) = \sum_{i=1}^n (Y(\vec{x}_i) - \mu(\vec{x}_i) - S(\vec{x}_i))^2.$$

Para estimar corretamente os parâmetros da média, os parâmetros da correlação deveriam ser conhecidos; mas, geralmente, não o são. Por outro lado, para estimar os parâmetros da correlação, seriam necessários os parâmetros da média, que tampouco são conhecidos. O que se faz então é considerar, inicialmente, o modelo $Y(\vec{x}_i) = \mu(\vec{x}_i) + Z(\vec{x}_i)$, que não considera a correlação espacial e cuja soma de quadrados dos resíduos é dada por

$$\text{SQR}(\beta) = \sum_{i=1}^n Z^2(\vec{x}_i) = \sum_{i=1}^n (Y(\vec{x}_i) - \mu(\vec{x}_i))^2.$$

Matricialmente, a equação de regressão linear múltipla é

$$\vec{Y}(\vec{x}) = \mathbf{D}\vec{\beta},$$

sendo \mathbf{D} uma matriz $n \times p$ de covariáveis e $\vec{\beta}$ o vetor dos parâmetros da regressão. A aplicação do método dos mínimos quadrados resultará no estimador

$$\tilde{\vec{\beta}} = (\mathbf{D}'\mathbf{D})^{-1}\mathbf{D}'\vec{Y}(\vec{x}).$$

Supondo que o modelo para a média seja especificado corretamente, os esti-

madores $\tilde{\beta}_j$ são não viesados e consistentes, independentemente da estrutura de covariância. Com essas estimativas, pode-se calcular o vetor dos resíduos

$$\vec{Z}(\vec{x}) = \vec{Y}(\vec{x}) - \mathbf{D}\tilde{\beta}.$$

Os elementos $Z(\vec{x}_i)$, $i = 1, \dots, n$, substituídos na Equação (2.5), permitem a obtenção das estimativas dos parâmetros de $S(\vec{x})$ e, por sua vez, a obtenção do estimador $\hat{\mathbf{V}}$ para \mathbf{V} apresentada na Equação (2.2). Com esse estimador, é possível obter uma estimativa de $\tilde{\beta}$ mais eficiente, dada pela estimativa de mínimos quadrados generalizados, conforme a equação

$$\hat{\beta} = (\mathbf{D}'\mathbf{V}^{-1}\mathbf{D})^{-1}\mathbf{D}'\mathbf{V}^{-1}\vec{Y}(\vec{x}), \quad (2.6)$$

substituindo-se nesta, \mathbf{V} por $\hat{\mathbf{V}}$. Esse procedimento iterativo é repetido até a obtenção da convergência. O estimador $\hat{\beta}$ é não viesado e tem a menor variância entre todos os estimadores lineares não viesados.

Por outro lado, utilizando o método da máxima verossimilhança, pode-se obter estimativas consistentes dos parâmetros de média e covariância maximizando a função de log-verossimilhança. No caso de $\vec{Y}(\vec{x})$ apresentar distribuição gaussiana, $\hat{\beta}$ dado em (2.6) coincide com a estimativa de máxima verossimilhança.

Admitindo uma superfície de tendência polinomial para $\mu(\vec{x})$, tem-se

$$\vec{Y}(\vec{x}) \sim N_n(\mathbf{D}\vec{\beta}; \sigma^2\mathbf{R} + \tau^2\mathbf{I}), \quad (2.7)$$

onde \mathbf{D} é uma matriz $n \times p$ de covariáveis, $\vec{\beta}$ é o vetor de parâmetros da regressão e a matriz de correlação \mathbf{R} depende de ϕ e da distância, os parâmetros da função de correlação exponencial. A função de verossimilhança de $\vec{Y}(\vec{x})$ é dada por:

$$L(\vec{Y}(\vec{x})) = (2\pi)^{-\frac{n}{2}} |\sigma^2\mathbf{R} + \tau^2\mathbf{I}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta})' (\sigma^2\mathbf{R} + \tau^2\mathbf{I})^{-1} (\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta}) \right\},$$

e a função de log-verossimilhança é

$$\begin{aligned} l(\vec{\beta}, \tau^2, \sigma^2, \phi) &= \ln \left[(2\pi)^{-\frac{n}{2}} \right] - \ln(|\sigma^2\mathbf{R} + \tau^2\mathbf{I}|^{\frac{1}{2}}) - \frac{1}{2} (\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta})' (\sigma^2\mathbf{R} + \tau^2\mathbf{I})^{-1} (\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta}) \\ &= -\frac{1}{2} \left\{ n \ln(2\pi) + \ln(|\sigma^2\mathbf{R} + \tau^2\mathbf{I}|) + (\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta})' (\sigma^2\mathbf{R} + \tau^2\mathbf{I})^{-1} (\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta}) \right\} \end{aligned} \quad (2.8)$$

Para proceder a maximização de (2.8), considera-se $\vec{Y}(\vec{x}) \sim N_n(\mathbf{D}\vec{\beta}; \sigma^2\mathbf{V})$ e os

seguintes resultados da álgebra matricial:

$$\frac{\partial}{\partial \vec{X}}(\mathbf{A}\vec{X}) = \mathbf{A}' \quad (2.9)$$

$$\frac{\partial}{\partial \vec{X}}(\vec{X}'\mathbf{A}\vec{X}) = 2\mathbf{A}\vec{X} \quad (2.10)$$

em que \mathbf{A} é uma matriz quadrada de ordem $n \times n$ e \vec{X} um vetor de ordem $n \times 1$.

Dessa forma, a Equação (2.8) pode ser reescrita como

$$l(\vec{\beta}, \sigma^2) = -\frac{1}{2} \left\{ n \ln(2\pi) + \ln(|\sigma^2 \mathbf{V}|) + (\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta})'(\sigma^2 \mathbf{V})^{-1}(\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta}) \right\}. \quad (2.11)$$

Nessa equação,

$$\begin{aligned} (\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta})'(\sigma^2 \mathbf{V})^{-1}(\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta}) &= \vec{Y}'(\vec{x})(\sigma^2 \mathbf{V})^{-1}\vec{Y}(\vec{x}) - \vec{Y}'(\vec{x})(\sigma^2 \mathbf{V})^{-1}\mathbf{D}\vec{\beta} - \\ &\quad \vec{\beta}'\mathbf{D}'(\sigma^2 \mathbf{V})^{-1}\vec{Y}(\vec{x}) + \vec{\beta}'\mathbf{D}'(\sigma^2 \mathbf{V})^{-1}\mathbf{D}\vec{\beta}, \end{aligned}$$

de modo que é permitido expressar (2.11) como

$$\begin{aligned} l(\vec{\beta}, \sigma^2) &= -\frac{1}{2} \left\{ n \ln(2\pi) + \ln(|\sigma^2 \mathbf{V}|) + \frac{1}{\sigma^2} \left[\vec{Y}(\vec{x})'\mathbf{V}^{-1}\vec{Y}(\vec{x}) - \right. \right. \\ &\quad \left. \left. 2\vec{Y}'(\vec{x})\mathbf{V}^{-1}\mathbf{D}\vec{\beta} + \vec{\beta}'(\mathbf{D}'\mathbf{V}^{-1}\mathbf{D})\vec{\beta} \right] \right\}. \quad (2.12) \end{aligned}$$

Aplicando os resultados (2.9) e (2.10) em (2.12), tem-se:

$$\frac{\partial}{\partial \vec{\beta}} l(\vec{\beta}, \sigma^2) = -\frac{1}{\sigma^2} \left[-(\vec{Y}'(\vec{x})\mathbf{V}^{-1}\mathbf{D})' + (\mathbf{D}'\mathbf{V}^{-1}\mathbf{D})\vec{\beta} \right].$$

Fazendo

$$\frac{1}{\sigma^2} \left(\mathbf{D}'\mathbf{V}^{-1}\vec{Y}(\vec{x}) - \mathbf{D}'\mathbf{V}^{-1}\mathbf{D}\hat{\vec{\beta}} \right) = 0,$$

vem

$$\hat{\vec{\beta}} = (\mathbf{D}'\mathbf{V}^{-1}\mathbf{D})^{-1}\mathbf{D}'\mathbf{V}^{-1}\vec{Y}(\vec{x}). \quad (2.13)$$

Nessa equação, $\hat{\vec{\beta}}$ coincide com o estimador de mínimos quadrados generalizados e depende somente de ϕ e \mathbf{v}^2 . Reescrevendo (2.11) e usando o fato de que $|\text{Var}(\vec{Y}(\vec{x}))| = |\sigma^2 \mathbf{V}| = (\sigma^2)^n |\mathbf{V}|$, tem-se

$$l(\vec{\beta}, \sigma^2) = -\frac{1}{2} \left\{ n \ln(2\pi) + n \ln(\sigma^2) + \ln|\mathbf{V}| + \frac{1}{\sigma^2} (\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta})'\mathbf{V}^{-1}(\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta}) \right\}. \quad (2.14)$$

Derivando (2.14) em relação a σ^2 , vem

$$\frac{\partial}{\partial \sigma^2} l(\vec{\beta}, \sigma^2) = -\frac{1}{2} \left\{ \frac{n}{\sigma^2} + (\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta})' \mathbf{V}^{-1} (\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta}) \left(-\frac{1}{(\sigma^2)^2} \right) \right\},$$

e, igualando esse termo a zero, obtém-se a expressão do estimador

$$\hat{\sigma}^2 = n^{-1} \left[(\vec{Y}(\vec{x}) - \mathbf{D}\hat{\vec{\beta}})' \mathbf{V}^{-1} (\vec{Y}(\vec{x}) - \mathbf{D}\hat{\vec{\beta}}) \right]. \quad (2.15)$$

No ponto de máximo, $\vec{\beta} = \hat{\vec{\beta}}$ e $\sigma^2 = \hat{\sigma}^2$ de forma que substituindo as Equações (2.13) e (2.15) na Equação (2.11) obtém-se a log-verossimilhança concentrada nos parâmetros ϕ e v^2

$$l_0(\phi, v^2) = -\frac{1}{2} \{ n \ln(2\pi) + n \ln(\hat{\sigma}^2) + \ln|\mathbf{V}| + n \}, \quad (2.16)$$

que recebe as constantes $\vec{Y}(\vec{x})$, a matriz \mathbf{D} , cujos elementos são os valores das covariáveis, e a matriz \mathbf{V} . Para se obter a estimativa dos parâmetros, a Equação (2.16) deve ser otimizada numericamente com relação a ϕ e v . Com o valor obtido para $\hat{\phi}$ e \hat{v}^2 , obtém-se a matriz $\hat{\mathbf{V}}$ que, substituída na Equação (2.13), permite a obtenção de $\hat{\vec{\beta}}$ e, conseqüentemente, $\hat{\sigma}^2$ de (2.15). Desta forma, substituindo-se \hat{v}^2 e $\hat{\sigma}^2$ em $v^2 = \tau^2/\sigma^2$ obtém-se $\hat{\tau}^2$.

Quando os dados $\vec{Y}(\vec{x})$ sofrem uma transformação $\vec{Y}^*(\vec{x})$, como sugerido na Subseção 2.1.2, a aplicação do método Jacobiano leva à obtenção da log-verossimilhança que, é uma função dos parâmetros e de $\vec{Y}(\vec{x})$:

$$l(\vec{\beta}, \sigma^2, \phi, v^2, \lambda) = (\lambda - 1) \sum_{i=1}^n \ln(y_i(\vec{x})) - 0,5 \left\{ n \ln(2\pi) + \ln|\sigma^2 \mathbf{V}_{(\phi, v^2)}| + (\vec{Y}^*(\vec{x}) - \mathbf{D}\vec{\beta})' (\sigma^2 \mathbf{V}_{(\phi, v^2)})^{-1} (\vec{Y}^*(\vec{x}) - \mathbf{D}\vec{\beta}) \right\}.$$

Essa função é otimizável, no sentido de que podem ser obtidas estimativas de máxima verossimilhança dos parâmetros do modelo. A transformação exige um método para transformar os valores $\vec{Y}^*(\vec{x})$ de volta para a escala original, e isto pode ser feito tanto por formas analíticas como também por simulação.

Para a avaliação da incerteza sobre os parâmetros do modelo, pode-se aplicar o método Delta, útil para aproximar a distribuição de alguma função das estatísticas (DEGROOT; SCHERVISH, 2002). O método pode ser visto, inclusive, como um procedimento de aproximação quadrática da função de verossimilhança, e é dado pelo resultado (PAWITAN, 2001), a seguir.

Teorema 2.1. : *Seja $\hat{\theta}$ um estimador de θ , baseado numa amostra de tamanho n , tal*

que

$$(\hat{\theta} - \theta) \rightarrow N\left(0; \frac{\sigma^2}{n}\right).$$

Então, para qualquer função $h(\cdot)$ diferenciável ao redor de θ e que $h'(\hat{\theta}) \neq 0$, tem-se

$$(h(\hat{\theta}) - h(\theta)) \rightarrow N\left(0; \frac{\sigma^2 |h'(\theta)|^2}{n}\right).$$

Pode-se dizer, então, que $h(\hat{\theta})$ é aproximadamente gaussiana com média $h(\theta)$ e variância $|h'(\theta)|^2 \text{Var}(\hat{\theta})$.

A obtenção dos intervalos de confiança pode ser baseada na função *deviance*, definida por

$$D(\theta) = 2 \ln \left(\frac{L(\hat{\theta})}{L(\theta)} \right) = 2[l(\hat{\theta}) - l(\theta)].$$

Se θ é o valor verdadeiro do parâmetro, assintoticamente,

$$D(\theta) \sim \chi_{1, (1-\alpha)}^2. \quad (2.17)$$

Assim, o intervalo para θ baseado na *deviance* (IC_D) será a região

$$\text{IC}_D = \{\theta; D(\theta) - D(\hat{\theta})\} < \chi_{1, (1-\alpha)}^2. \quad (2.18)$$

Lindsey (2001) e Pawitan (2001), por exemplo, apresentam outra forma de se obter a incerteza sobre os parâmetros, através da verossimilhança *perfilhada*. Pawitan (2001) afirma que existem casos onde o interesse pode estar em um subconjunto de parâmetros. Por exemplo, em um modelo gaussiano, o interesse pode estar em μ , enquanto σ é um parâmetro *nuisance* que torna o modelo capaz de acomodar a variabilidade dos dados. Torna-se necessário, então, “concentrar” a verossimilhança em um único parâmetro, eliminando-se o parâmetro *nuisance*, mas considerando-se a incerteza devido à esse parâmetro.

O procedimento para eliminar o parâmetro *nuisance* é substituí-lo por sua estimativa de máxima verossimilhança em cada valor fixado do parâmetro de interesse, resultando na função de verossimilhança perfilhada. Dessa verossimilhança constróem-se os intervalos de confiança.

No caso de um vetor de parâmetros $\theta = (\theta_1, \theta_2)$, dada a função de verossimilhança conjunta $L(\theta_1, \theta_2)$ a verossimilhança perfilhada de θ_1 é definida como

$$L(\theta_1) = \max_{\theta_2} L(\theta_1, \theta_2)$$

onde a maximização é realizada em um valor de θ_1 fixado. Pawitan (2001) observa que, no valor fixado de θ_1 , a estimativa de máxima verossimilhança de θ_2 é geralmente uma

função de θ_1 ,

$$L(\theta_1) = L(\theta_1, \hat{\theta}_{2\theta_1}). \quad (2.19)$$

Alternativamente à verossimilhança perfilhada, existem outras formas de eliminar parâmetros *nuisance* em inferência estatística, que são os métodos da verossimilhança condicional, marginal e verossimilhança perfilhada modificada, descritos, por exemplo, em Lindsey (2001) e Pawitan (2001).

Finalmente, a quantificação da incerteza sobre os parâmetros também pode ser feita mediante a aplicação de métodos bayesianos, conforme será apresentado adiante.

2.2 Predição Linear Espacial

2.2.1 Conceitos de Predição

Considere o vetor $\mathbf{S}(\vec{x}) = (S(\vec{x}_1), S(\vec{x}_2), \dots, S(\vec{x}_n))'$ com distribuição multivariada

$$\mathbf{S}(\vec{x}) \sim N_n(\mu \vec{1}; \sigma^2 \mathbf{R}),$$

em que \mathbf{R} é uma matriz de ordem $n \times n$, com elementos r_{ij} definidos pela função de correlação $\rho(\|\vec{x}_i - \vec{x}_j\|)$, e

$$\vec{Y}(\vec{x}) \sim N_n(\mu \vec{1}; \sigma^2 \mathbf{V}).$$

O vetor $\vec{Y}(\vec{x})$ tem como elementos variáveis aleatórias cujos valores são observados e T é uma variável aleatória cujo valor será predito a partir do vetor observado $\vec{Y}(\vec{x})$.

Um preditor pontual de T é uma função qualquer de $\vec{Y}(\vec{x})$, representada por

$$\hat{T} = t(\vec{Y}(\vec{x})),$$

que tem erro quadrático médio (EQM) de predição definido como

$$\text{EQM}(\hat{T}) = E((T - \hat{T})^2).$$

O $\text{EQM}(\hat{T})$ assume o valor mínimo quando $\hat{T} = E(T|\vec{Y}(\vec{x}))$. A demonstração é

apresentada a seguir, observando inicialmente que,

$$\begin{aligned} E((T - \hat{T})^2) &= E_{\vec{Y}(\vec{x})}(E_T((T - \hat{T})^2|\vec{Y}(\vec{x}))) \\ &= E_{\vec{Y}(\vec{x})}(Var_T((T - \hat{T})|\vec{Y}(\vec{x})) + \{E_T((T - \hat{T})|\vec{Y}(\vec{x}))\}^2), \end{aligned} \quad (2.20)$$

em que os subscritos nos dois operadores da esperança indicam que as esperanças são calculadas com relação a $\vec{Y}(\vec{x})$ e T , respectivamente. Tem-se, também,

$$Var_T((T - \hat{T}|\vec{Y}(\vec{x}))) = Var_T(T|\vec{Y}(\vec{x})) + Var_T(\hat{T}|\vec{Y}(\vec{x})) - 2Cov_T(T|\vec{Y}(\vec{x}), \hat{T}|\vec{Y}(\vec{x})),$$

e, como $Var_T(\hat{T}|\vec{Y}(\vec{x}))$ e $Cov_T(T|\vec{Y}(\vec{x}), \hat{T}|\vec{Y}(\vec{x}))$ são zero, já que condicionado em $\vec{Y}(\vec{x})$, \hat{T} (que é função de $\vec{Y}(\vec{x})$) é constante, pode-se escrever

$$Var_T((T - \hat{T}|\vec{Y}(\vec{x}))) = Var_T(T|\vec{Y}(\vec{x})).$$

Por outro lado,

$$\begin{aligned} E_T((T - \hat{T})|\vec{Y}(\vec{x})) &= E(T|\vec{Y}(\vec{x})) - E(\hat{T}|\vec{Y}(\vec{x})) \\ &= E(T|\vec{Y}(\vec{x})) - \hat{T}, \end{aligned}$$

que, substituídas em (2.20), fornecem

$$E((T - \hat{T})^2) = E_{\vec{Y}(\vec{x})}(Var_T(T|\vec{Y}(\vec{x})) + \{E(T|\vec{Y}(\vec{x})) - \hat{T}\}^2). \quad (2.21)$$

Da Equação (2.21), obtém-se o erro quadrático médio de \hat{T} , dado por

$$E((T - \hat{T})^2) = E_{\vec{Y}(\vec{x})}(Var_T(T|\vec{Y}(\vec{x}))), \quad (2.22)$$

quando $\hat{T} = E(T|\vec{Y}(\vec{x}))$. Além disso,

$$\begin{aligned} E((T - \hat{T})^2) &= Var(T - \hat{T}) + \{E(T - \hat{T})\}^2 \\ &= Var(T) + Var(\hat{T}) - 2Cov(T, \hat{T}) + \{E(T) - \hat{T}\}^2 \\ &= Var(T) - 2Cov(T, \hat{T}). \end{aligned}$$

Então,

$$Var(T) = E((T - \hat{T})^2) + 2Cov(T, \hat{T})$$

e, conseqüentemente,

$$E((T - \hat{T})^2) \leq Var(T)$$

se T e $\vec{Y}(\vec{x})$ são independentes, o que completa a demonstração.

Se $S(\vec{x})$ for um processo gaussiano estacionário, se os dados $\vec{Y}(\vec{x})$ são gerados por um modelo gaussiano estacionário e se T é igual a $S(\vec{x})$, o vetor $(T, \vec{Y}(\vec{x})) = (S(\vec{x}), \vec{Y}(\vec{x}))$ tem

distribuição gaussiana multivariada e a distribuição condicional de T dado $\vec{Y}(\vec{x})$ (MOOD; GRAYBILL; BOES, 1974) é também gaussiana, com média

$$\mu_{T|\vec{Y}(\vec{x})} = \mu_T + \Sigma_{T\vec{Y}(\vec{x})} \Sigma_{\vec{Y}(\vec{x})\vec{Y}(\vec{x})}^{-1} (\vec{Y}(\vec{x}) - \mu_{\vec{Y}(\vec{x})}),$$

e variância

$$\Sigma_{T|\vec{Y}(\vec{x})} = \Sigma_{TT} - \Sigma_{T\vec{Y}(\vec{x})} \Sigma_{\vec{Y}(\vec{x})\vec{Y}(\vec{x})}^{-1} \Sigma_{\vec{Y}(\vec{x})T}.$$

Logo, $[T, \vec{Y}(\vec{x})]$ é gaussiana multivariada com média $\mu \vec{1}$ e matriz de covariância

$$\begin{bmatrix} \sigma^2 & \sigma^2 \vec{r}' \\ \sigma^2 \vec{r} & \sigma^2 \mathbf{V} \end{bmatrix},$$

em que \vec{r} é um vetor com elementos $r_i = \rho(\|\vec{x} - \vec{x}_i\|)$.

Dessa forma, com $\Sigma_{T\vec{Y}(\vec{x})} = \sigma^2 \vec{r}'$ e $\Sigma_{\vec{Y}(\vec{x})\vec{Y}(\vec{x})} = \sigma^2 \mathbf{V}$, o preditor do erro quadrado médio mínimo para $S(\vec{x})$ é:

$$\hat{S}(\vec{x}) = \mu + \vec{r}' \mathbf{V}^{-1} (\vec{Y}(\vec{x}) - \mu \vec{1}),$$

com variância de predição

$$\text{Var}(S(\vec{x})|\vec{Y}(\vec{x})) = \sigma^2 (1 - \vec{r}' \mathbf{V}^{-1} \vec{r}).$$

Como a variância de predição não depende de $\vec{Y}(\vec{x})$, da Equação (2.22) tem-se $E((S(\vec{x}) - \hat{S}(\vec{x}))^2) = \text{Var}(S(\vec{x})|\vec{Y}(\vec{x}))$.

Ao escrever o preditor de S em termos de $\hat{S}(\vec{x}_0)$ onde \vec{x}_0 é a localização de predição, pode-se observar que $\vec{r}' \mathbf{V}^{-1}$ nada mais é do que uma combinação linear da média μ e de $Y(\vec{x}_i)$, de modo que

$$\begin{aligned} \hat{S}(\vec{x}_0) &= \mu + \sum_{i=1}^n a_i(\vec{x}_0) (Y(\vec{x}_i) - \mu) \\ &= \mu + \sum_{i=1}^n a_i(\vec{x}_0) Y(\vec{x}_i) - \sum_{i=1}^n a_i(\vec{x}_0) \mu \\ &= \left\{ 1 - \sum_{i=1}^n a_i(\vec{x}_0) \right\} \mu + \sum_{i=1}^n a_i(\vec{x}_0) Y(\vec{x}_i), \end{aligned}$$

em que $a_1(\vec{x}_0), a_2(\vec{x}_0), \dots, a_n(\vec{x}_0)$, são denominados pesos de predição.

Segundo Diggle, Ribeiro Jr. e Christensen (2003), em muitas aplicações, o foco inferencial pode não estar em $S(\vec{x}_0)$, mas em alguma outra propriedade do processo, tal como a média ou valor máximo sobre a área de estudo ou subárea desta, por exemplo.

Primeiramente, os autores consideram T qualquer funcional linear de $S(\vec{x})$, ou seja,

$$T = \int_A a(\vec{x})S(\vec{x}) d\vec{x}$$

para alguma função peso $a(\vec{x})$. Como já visto, sob o modelo gaussiano, o vetor $[T, \vec{Y}(\vec{x})]$ é gaussiano multivariado e, assim, $[T|\vec{Y}(\vec{x}) = \vec{y}(\vec{x})]$ é gaussiano univariado se T é um escalar, com média dada por

$$E(T|\vec{Y}(\vec{x})) = \int_A a(\vec{x})E(S(\vec{x})|\vec{Y}(\vec{x})) d\vec{x},$$

o que resulta em

$$\hat{T} = \int_A a(\vec{x})\hat{S}(\vec{x}) d\vec{x}.$$

A variância de $T|\vec{Y}(\vec{x})$ é

$$Var(T|\vec{Y}(\vec{x})) = \int_A \int_A a(\vec{x})a(\vec{x}')Cov(S(\vec{x}), S(\vec{x}')) d\vec{x}d\vec{x}'.$$

Em outras palavras, os autores afirmam que, dada a superfície predita $\hat{S}(\vec{x})$, é razoável calcular qualquer propriedade linear dessa superfície e usar o resultado como o preditor para a propriedade linear correspondente da superfície verdadeira $S(\vec{x})$. Isso não será válido para propriedades não lineares.

2.2.2 Krigagem

Banerjee, Carlin e Gelfand (2004) colocam que *krigagem* é um problema de predição espacial ótima: dadas as observações de um processo estocástico $\vec{Y}(\vec{x}) = (Y(\vec{x}_1), Y(\vec{x}_2), \dots, Y(\vec{x}_n))'$ deseja-se prever a variável $Y(\vec{x})$ em uma localização não observada. Em outras palavras, deseja-se encontrar o melhor preditor do valor de $Y(\vec{x}_0)$ baseado nas observações $y(\vec{x})$ de $Y(\vec{x})$.

Como visto na Subseção 2.2.1, no modelo gaussiano o melhor preditor é o que apresenta o menor erro quadrático médio para $T = S(\vec{x}_0)$ e é dado por

$$\hat{T} = \mu + \vec{r}'\mathbf{V}^{-1}(\vec{Y}(\vec{x}) - \mu\vec{1}), \quad (2.23)$$

com variância de predição

$$Var(T|\vec{Y}(\vec{x})) = \sigma^2(1 - \vec{r}'\mathbf{V}^{-1}\vec{r}). \quad (2.24)$$

Observa-se que, devido ao fato dos parâmetros do modelo serem quantidades desconhecidas, a estimativa $\hat{\sigma}^2$ obtida a partir de (2.13) e (2.15), e as estimativas dos

parâmetros em r e V que são ϕ e v , obtidas através do processo de maximização da função de log-verossimilhança do modelo são substituídas em (2.23) e (2.24). O preditor \hat{T} é linear nos dados e esse método é conhecido como krigagem simples quando um valor arbitrário ou conhecido de μ é especificado.

No método conhecido como krigagem ordinária, o parâmetro média é tratado como desconhecido e os parâmetros da covariância são supostos conhecidos. Assim, o preditor é escrito como a combinação linear

$$\hat{T} = \hat{S}(\vec{x}) = \sum_{i=1}^n a_i(\vec{x})Y(\vec{x}_i),$$

em que $a_i(\vec{x})$, são os pesos de krigagem e satisfazem $\sum_{i=1}^n a_i(\vec{x}) = 1$ para qualquer localização de predição. De forma equivalente, substitui-se em (2.23) a média μ pelo estimador de mínimos quadrados generalizados

$$\hat{\mu} = (\vec{1}'\mathbf{V}^{-1}\vec{1})^{-1}\vec{1}'\mathbf{V}^{-1}\vec{Y}(\vec{x}),$$

de onde segue que

$$\hat{T} = (\vec{1}'\mathbf{V}^{-1}\vec{1})^{-1}\vec{1}'\mathbf{V}^{-1}\vec{Y}(\vec{x}) + \vec{r}'\mathbf{V}^{-1} \left[\vec{Y}(\vec{x}) - (\vec{1}'\mathbf{V}^{-1}\vec{1})^{-1}\vec{1}'\mathbf{V}^{-1}\vec{Y}(\vec{x}) \right].$$

De acordo com Diggle, Ribeiro Jr. e Christensen (2003) e Diggle e Ribeiro Jr. (2007), quando uma transformação nos dados originais é feita com o objetivo de que esses passem a seguir uma distribuição gaussiana, em geral os dados transformados são escritos como $\vec{Y}^*(\vec{x}) = h_\lambda(\vec{Y}(\vec{x}))$, onde $h(\cdot)$ é uma função de transformação. Para diferentes especificações de valores de λ , tem-se modelos e resultados diferentes, e as predições são consideradas como as médias das predições obtidas através dos modelos correspondentes aos diferentes valores de λ . Para transformar de volta os dados $\vec{Y}^*(\vec{x})$ para a escala original, seria necessário calcular a função inversa $h_\lambda^{-1}(\cdot)$ para cada valor de λ e calcular as médias obtendo-se valores aproximados. A maneira mais simples de fazer isso é supor que o objetivo de predição seja

$$T(\vec{x}) = \exp\{\mu + S(\vec{x})\}. \quad (2.25)$$

A avaliação do preditor do erro quadrático médio para a Equação (2.25) não é direta exceto para $h(\cdot) = \ln(\cdot)$, que implica $h^{-1}(\cdot) = \exp(\cdot)$, ou seja,

$$T(\vec{x}) = \exp\{\mu\}\exp\{S(\vec{x})\} = \exp\{\mu\}T_0(\vec{x}).$$

A distribuição de $S(\vec{x})$ dado $\vec{Y}^*(\vec{x})$ é gaussiana univariada com média $\hat{S}(\vec{x})$ e variância $v(\vec{x})$ dadas por (2.23) e (2.24), respectivamente, substituindo-se $\vec{Y}(\vec{x})$ por $\vec{Y}^*(\vec{x})$. A distribuição de $T_0(\vec{x})$ condicionada a $\vec{Y}^*(\vec{x})$ é log-normal e desta forma a média de $T_0(\vec{x})$ é dada por

$$E(T_0(\vec{x})) = E(\exp\{S(\vec{x})\}) = \exp\{\hat{S}(\vec{x}) + 0.5v(\vec{x})\}.$$

Então, a função geratriz de momentos de $S(\vec{x})$ é

$$\Psi_{S(\vec{x})}(a) = E\left(e^{aS(\vec{x})}\right) = e^{a\hat{S}(\vec{x}) + \frac{1}{2}a^2v(\vec{x})} \quad a \in \mathbb{R},$$

a qual para $a = 1$, resulta em

$$\hat{T}_0(\vec{x}) = E(T_0(\vec{x})) = e^{\hat{S}(\vec{x}) + \frac{v(\vec{x})}{2}},$$

e para $a = 2$ em

$$E((T_0(\vec{x}))^2) = e^{2\hat{S}(\vec{x}) + 2v(\vec{x})},$$

de onde se obtém a variância de predição:

$$\text{Var}(T_0(\vec{x})|\vec{Y}^*) = e^{2\hat{S}(\vec{x}) + v(\vec{x})} [e^{v(\vec{x})} - 1].$$

Como já mencionado anteriormente, a transformação de volta dos valores $\vec{Y}^*(\vec{x})$ para a escala original pode, ainda, ser feita por simulação como será descrito adiante.

2.3 Inferência Bayesiana Para Predição Espacial

2.3.1 Distribuição Preditiva Bayesiana

Na inferência bayesiana, as distribuições de probabilidade são associadas aos parâmetros do modelo para refletir a incerteza sobre esses. Essas incertezas sobre os parâmetros podem ser incorporadas formalmente nas predições. Para isto, considera-se um vetor de parâmetros $\vec{\theta}$ e um vetor aleatório $\vec{Y}(\vec{x})$ com distribuição de probabilidade determinada pela função $P(\vec{Y}(\vec{x})|\vec{\theta})$. Ao desprezar o termo constante da função $P(\vec{Y}(\vec{x})|\vec{\theta})$, a função de verossimilhança $L(\vec{\theta}; \vec{Y}(\vec{x}))$ pode ser escrita como

$$L(\vec{\theta}; \vec{Y}(\vec{x})) \propto |\sigma^2\mathbf{R} + \tau^2\mathbf{I}|^{-\frac{1}{2}} \cdot \exp\left\{-\frac{1}{2}(\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta})'(\sigma^2\mathbf{R} + \tau^2\mathbf{I})^{-1}(\vec{Y}(\vec{x}) - \mathbf{D}\vec{\beta})\right\}. \quad (2.26)$$

De acordo com Gilks, Richardson e Spiegelhalter (1996) e Gamerman e Lopes (2006), para se fazer inferência é necessário construir a distribuição conjunta dos vetores aleatórios $\vec{Y}(\vec{x})$ e $\vec{\theta}$ em duas partes: uma distribuição *a priori* $P(\vec{\theta})$ e uma função de verossimilhança $P(\vec{Y}(\vec{x})|\vec{\theta})$, ou seja:

$$P(\vec{Y}(\vec{x}), \vec{\theta}) = P(\vec{Y}(\vec{x})|\vec{\theta})P(\vec{\theta}).$$

Gill (2002) acrescenta que a expressão $P(\vec{\theta})$ é uma afirmação formalizada na forma de uma distribuição de probabilidade do conhecimento prévio sobre $\vec{\theta}$ antes de se observar os dados. A idéia básica é especificar uma distribuição *a priori* para $\vec{\theta}$ que descreva em

termos probabilísticos a informação *a priori* e a incerteza sobre θ .

O Teorema de Bayes é aplicado para determinar a distribuição de $\vec{\theta}$ condicionada aos dados $\vec{Y}(\vec{x})$:

$$P(\vec{\theta}|\vec{Y}(\vec{x})) = \frac{P(\vec{\theta}, \vec{Y}(\vec{x}))}{P(\vec{Y}(\vec{x}))} = \frac{P(\vec{Y}(\vec{x})|\vec{\theta})P(\vec{\theta})}{P(\vec{Y}(\vec{x}))} = \frac{P(\vec{Y}(\vec{x})|\vec{\theta})P(\vec{\theta})}{\int P(\vec{\theta}, \vec{Y}(\vec{x}))d\vec{\theta}}, \quad (2.27)$$

e é chamada de distribuição *a posteriori* de $\vec{\theta}$. Como o denominador da expressão não depende de $\vec{\theta}$, pode ser considerado uma constante normalizadora e, ao desprezá-la, a Equação (2.27) pode ser escrita como

$$P(\vec{\theta}|\vec{Y}(\vec{x})) \propto P(\vec{Y}(\vec{x})|\vec{\theta})P(\vec{\theta}). \quad (2.28)$$

Para Gill (2002) o lado direito da Equação (2.28) implica que a inferência para $\vec{\theta}$ é um compromisso entre informação *a priori* e a informação fornecida pelos dados, enquanto o lado esquerdo fornece a distribuição atualizada para $\vec{\theta}$ após condicionar aos dados.

Em alguns casos é possível deduzir analiticamente uma distribuição *a posteriori* de modo que se apresente como uma distribuição conhecida, caso contrário, utilizam-se métodos numéricos computacionalmente intensivos. Mas, como afirmam Gamerman e Lopes (2006), a obtenção da distribuição *a posteriori* é um passo importante, mas não o final. É preciso ser capaz de se extrair informações significativas desta distribuição e isto está relacionado com a avaliação de resumos estatísticos tais como média, mediana ou moda, ou intervalos de probabilidade. Estas quantidades podem ser expressas em termos de médias *a posteriori* de funções de $\vec{\theta}$ (GILKS; RICHARDSON; SPIEGELHALTER, 1996). A média *a posteriori* de uma função $g(\vec{\theta})$ é

$$E(g(\vec{\theta})|\vec{Y}(\vec{x})) = \frac{\int g(\vec{\theta})P(\vec{\theta})P(\vec{Y}(\vec{x})|\vec{\theta})d\vec{\theta}}{\int P(\vec{\theta})P(\vec{Y}(\vec{x})|\vec{\theta})d\vec{\theta}}, \quad (2.29)$$

ou

$$E(g(\vec{\theta})|\vec{Y}(\vec{x})) \propto \int f(\vec{\theta})P(\vec{\theta})P(\vec{Y}(\vec{x})|\vec{\theta})d\vec{\theta}.$$

definidas no domínio de variação das distribuições *a priori*.

A resolução numérica destas integrais inclui aproximações analíticas tais como aproximação de Laplace (Nott, Fielding e Leonte (2009), Gelman et al. (2003), Gilks, Richardson e Spiegelhalter (1996)), aproximação numérica como por exemplo quadratura de Gauss-Hermite (Paulino, Turkman e Murteira (2003), Gamerman e Lopes (2006)) e integração Monte Carlo, incluindo cadeias de Markov de Monte Carlo - MCMC (Gamerman e Lopes (2006), Rue e Held (2005), Lee (2004), Gelman et al. (2003), Gill (2002), Gilks, Richardson e Spiegelhalter (1996)). Neste trabalho foram adotados os dois últimos procedimentos.

Quanto *as priori*, Ribeiro Jr. e Diggle (1999) mencionam que a escolha é uma questão delicada em inferência bayesiana. Distribuições *a priori* que levam a uma distribuição *a posteriori* da mesma família de distribuições são chamadas *priori conjugadas*.

Essas distribuições *a priori* podem ser computacionalmente convenientes mas não deveriam ser escolhidas somente por isso. Dois casos extremos para a escolha da *priori* são: quando os parâmetros são perfeitamente conhecidos *as priori* podem ser vistas como distribuições degeneradas nos valores dos parâmetros; quando o conhecimento da *priori* sobre os parâmetros é vaga podem ser adotadas *priori* não informativas, vagas ou impróprias.

Por outro lado, de acordo com Ribeiro Jr. e Diggle (1999), a base da predição bayesiana é a distribuição preditiva $P(\mathbf{S}(\vec{x})|\vec{Y}(\vec{x}))$. Esta distribuição leva em consideração a incerteza sobre os parâmetros calculando, por exemplo, a média da distribuição condicional $P(\mathbf{S}(\vec{x})|\vec{Y}(\vec{x}), \vec{\theta})$, sobre o espaço dos parâmetros, com pesos dados pela distribuição *a posteriori* dos parâmetros do modelo $P(\vec{\theta}|\vec{Y}(\vec{x}))$:

$$\begin{aligned} P(\mathbf{S}(\vec{x})|\vec{Y}(\vec{x})) &= \int P(\mathbf{S}(\vec{x}), \vec{\theta}|\vec{Y}(\vec{x})) d\vec{\theta} \\ &= \int P(\mathbf{S}(\vec{x})|\vec{Y}(\vec{x}), \vec{\theta})P(\vec{\theta}|\vec{Y}(\vec{x})) d\vec{\theta}. \end{aligned}$$

Cabe ressaltar que podem ser calculadas outras estatísticas de interesse ou mais apropriadas, como a mediana ou moda, a partir da distribuição preditiva.

2.3.2 Cadeias de Markov

Gelman et al. (2003) definem uma *cadeia de Markov* como uma sequência de variáveis aleatórias $\vec{\theta}_1, \vec{\theta}_2, \dots$ para os quais, para qualquer t , a distribuição de $\vec{\theta}_t$ depende somente do anterior, $\vec{\theta}_{t-1}$.

A simulação de uma *cadeia de Markov* (*Monte Carlo via cadeia de Markov* ou MCMC, na sigla em inglês) é definida como um método baseado na amostragem de valores de $\vec{\theta}$ de distribuições aproximadas, corrigindo-os para uma melhor aproximação da distribuição *a posteriori*, $P(\vec{\theta}|\vec{Y}(\vec{x}))$. Os valores de $\vec{\theta}$ são amostrados sequencialmente, com a distribuição dependente do último valor amostrado. Estes valores amostrados formam então uma *cadeia de Markov*.

O MCMC é usado quando não é possível, ou não é computacionalmente eficiente, amostrar $\vec{\theta}_t$ diretamente de $P(\vec{\theta}|\vec{Y}(\vec{x}))$. Amostra-se iterativamente de forma tal que a cada passo do processo espera-se que os valores sejam amostrados de uma distribuição que seja a mais próxima possível de $P(\vec{\theta}|\vec{Y}(\vec{x}))$.

O princípio básico do MCMC (GILL, 2002) é que, se uma cadeia iterativa de valores consecutivos, gerados computacionalmente, pode ser construído com cuidado suficiente, estimativas empíricas podem ser obtidas dos últimos valores da cadeia. Em geral, inicia-se a cadeia após um período descarte de iterações, denominado período de aquecimento (*burn-in*) da cadeia deixando-a “correr” até que se aproxime da distribuição estacionária.

As técnicas MCMC utilizadas neste trabalho foram o algoritmo Metropolis-Hastings e amostrador de Gibbs. Para o algoritmo Metropolis-Hastings (GILKS;

RICHARDSON; SPIEGELHALTER, 1996), em cada estado $\vec{\theta}^{(t)}$ da cadeia, o próximo estado $\vec{\theta}^{(t+1)}$ é escolhido amostrando-se um valor $\vec{\theta}'$ de uma distribuição proposta $q(\cdot|\vec{\theta}^{(t)})$ que pode depender do valor atual $\vec{\theta}^{(t)}$. O valor $\vec{\theta}'$ será aceito com probabilidade $\alpha(\vec{\theta}^{(t)}, \vec{\theta}')$, em que

$$\alpha(\vec{\theta}, \vec{\theta}') = \min \left(1, \frac{P(\vec{Y}(\vec{x})|\vec{\theta})P(\vec{\theta}')q(\vec{\theta}|\vec{\theta}')}{P(\vec{Y}(\vec{x})|\vec{\theta})P(\vec{\theta})q(\vec{\theta}'|\vec{\theta})} \right).$$

Se o novo valor for aceito, o próximo estado torna-se $\vec{\theta}^{(t+1)} = \vec{\theta}'$. Caso contrário, a cadeia “não se move”, ou seja, $\vec{\theta}^{(t+1)} = \vec{\theta}^{(t)}$.

Ehlers (2006), Gilks, Richardson e Spiegelhalter (1996), Gamerman e Lopes (2006) desenvolveram os seguintes passos de um algoritmo com este propósito de aceitar/rejeitar valores:

1. escolher o número n de etapas (número de simulações ou número de passos da cadeia);
2. iniciar o contador de iterações t , fazendo $t = 1$;
3. especificar um valor inicial para $\vec{\theta}^{(t)}$;
4. gerar um valor $\vec{\theta}'$ de uma distribuição proposta $q(\vec{\theta}'|\vec{\theta})$. Esta distribuição deve ser escolhida com cautela para garantir a eficiência do algoritmo;
5. gerar $u \sim U(0;1)$ (*priori*);
6. determinar a probabilidade de aceitação $\alpha(\vec{\theta}; \vec{\theta}')$ dada por:

$$\alpha(\vec{\theta}; \vec{\theta}') = \min \left\{ 1; \frac{P(\vec{Y}(\vec{x})|\vec{\theta}')P(\vec{\theta}')q(\vec{\theta}|\vec{\theta}')}{P(\vec{Y}(\vec{x})|\vec{\theta})P(\vec{\theta})q(\vec{\theta}'|\vec{\theta})} \right\}$$

em que $P(\vec{\theta}|\vec{Y}(\vec{x}))$, a distribuição *a posteriori*, é a distribuição de interesse;

7. se $u < \alpha(\vec{\theta}; \vec{\theta}')$, o valor de $\vec{\theta}'$ é aceito e $\vec{\theta}^{(t)} = \vec{\theta}'$. Caso contrário, $\vec{\theta}'$ é rejeitado e $\vec{\theta}^{(t)} = \vec{\theta}^{(t-1)}$;
8. se $t = n$ (é o último passo) encerrar o processo, senão:
9. incrementar o contador t fazendo $t = t + 1$.
10. voltar para o passo 4;
11. FIM.

Segundo Ehlers (2006) no amostrador de Gibbs, as transições de estado são feitas de acordo com as distribuições condicionais completas $P(\theta_i|\vec{\theta}_{-i})$, obtidas a partir da distribuição conjunta como:

$$P(\theta_i|\vec{\theta}_{-i}) = \frac{P(\vec{\theta})}{\int P(\vec{\theta}) d\theta_i} \quad (2.30)$$

em que $\vec{\theta}_{-i} = \mathbf{1}_{-i}\vec{\theta}$, com $\mathbf{1}_{-i}$ uma matriz identidade excluída a i -ésima linha e $\vec{\theta} = (\theta_1, \theta_2, \dots, \theta_p)'$, $i = 1, 2, \dots, p$, sendo p neste caso o número de parâmetros a serem estimados, ou seja:

$$\vec{\theta}_{-i} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{i-1} \\ \theta_{i+1} \\ \vdots \\ \theta_p \end{bmatrix}_{(p-1) \times 1} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 \end{bmatrix}_{(p-1) \times p} \times \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \vdots \\ \theta_p \end{bmatrix}_{p \times 1}.$$

Em outras palavras, pode-se dizer que a distribuição condicional completa é a distribuição da i -ésima componente de $\vec{\theta}$ condicionada em todas as outras componentes e a Equação (2.30) pode ser escrita como $P(\vec{\theta}) = P(\theta_i | \vec{\theta}_{-i}) P_{\vec{\theta}_{-i}}(\vec{\theta}_{-i})$.

Desta forma, deseja-se gerar uma amostra de $P(\vec{\theta})$ onde a transição ocorre da seguinte forma:

- i. especificar valores para um vetor inicial $\vec{\theta}^{(0)}$;
- ii. calcular a probabilidade de θ_1 condicionada aos θ^s restantes com os valores especificados inicialmente;
- iii. nas transições de θ_{r-1} para θ_r , $r = 2, 3, \dots, p$, calcular, para θ_r , a probabilidade $P(\theta_r | \vec{\theta}_{-r})$ substituindo em $\vec{\theta}_{-r}$ o valor de θ_{r-1} calculado no passo anterior.

Se as distribuições condicionais completas forem conhecidas, então o algoritmo será desenvolvido seguindo os passos:

1. iniciar o contador de iterações em $t = 0$ e definir o número n de iterações;
2. especificar valores iniciais para $\vec{\theta}^{(0)}$;
3. avançar t fazendo $t = t + 1$ e obter $\vec{\theta}^{(t)}$ a partir de $\vec{\theta}^{(t-1)}$ por geração sucessiva de valores como:

$$\begin{aligned} \theta_1^{(t)} &\sim P(\theta_1 | \theta_2^{(t-1)}, \theta_3^{(t-1)}, \theta_4^{(t-1)}, \dots, \theta_p^{(t-1)}) \\ \theta_2^{(t)} &\sim P(\theta_2 | \theta_1^{(t)}, \theta_3^{(t-1)}, \theta_4^{(t-1)}, \dots, \theta_p^{(t-1)}) \\ \theta_3^{(t)} &\sim P(\theta_3 | \theta_1^{(t)}, \theta_2^{(t)}, \theta_4^{(t-1)}, \dots, \theta_p^{(t-1)}) \\ &\vdots \\ \theta_p^{(t)} &\sim P(\theta_p | \theta_1^{(t)}, \theta_2^{(t)}, \theta_3^{(t)}, \dots, \theta_{p-1}^{(t)}) \end{aligned}$$

4. Se $t = n$, encerrar;
5. Voltar à etapa 3.

2.3.3 Avaliação da Convergência

Uma vez que métodos de simulação de cadeias de Markov são utilizados, faz-se necessário um estudo sobre a convergência das cadeias. Diagnósticos de convergência são discutidos em Gilks, Richardson e Spiegelhalter (1996), Gamerman e Lopes (2006), Gill (2002), Carlin e Louis (2009), entre outros.

Para Gilks, Richardson e Spiegelhalter (1996), a tarefa de monitorar a convergência é estimar o quanto a inferência baseada em simulações de cadeia de Markov diferem da distribuição *a posteriori* e o método adotado, inspirado na análise de variância, é formar uma super e uma subestimativa da variância da distribuição *a posteriori*, com a propriedade que as estimativas serão grosseiramente iguais em convergência, mas não antes.

Análise de convergência pode ser feita, por exemplo, calculando-se as autocorrelações dos parâmetros *a posteriori* (GILL, 2002). Decréscimos nas autocorrelações à medida que as defasagens aumentam são indicativos de convergência da cadeia.

Gráficos da trajetória da cadeia também são utilizados na avaliação. Espera-se que a série ocorra de forma estável em torno da linha média. Observa-se a não convergência quando a trajetória se afasta seguindo uma tendência que não seja a linha média.

Os gráficos das densidades estimadas dos parâmetros *a posteriori* auxiliam no sentido de verificação de multimodalidade, indicativo de não convergência da cadeia.

O teste de Geweke apresentado em Gill (2002) e Gamerman e Lopes (2006), por exemplo, baseia-se na comparação de alguma proporção do início da cadeia após o período de aquecimento com alguma proporção do fim da cadeia. É um teste de diferença de médias usando uma aproximação assintótica para o erro padrão da diferença.

Considere uma cadeia com $m + n$ iterações, $\theta_1, \dots, \theta_{m+n}$, uma função de interesse $f()$ e as médias

$$\bar{f}_b = \frac{1}{n_b} \sum_{i=m+1}^{m+n_b} f(\theta_i) \quad \text{e} \quad \bar{f}_a = \frac{1}{n_a} \sum_{i=m+n-n_a+1}^{m+n} f(\theta_i)$$

em que $n_b + n_a < n$. Se m é o comprimento do período de aquecimento, então \bar{f}_a e \bar{f}_b são as médias ergódicas (não sensíveis às condições iniciais) no fim e começo do período de convergência e deveriam se comportar de forma similar. À medida que n aumenta e as razões n_a e n_b permanecem fixas então

$$z_G = \frac{\bar{f}_a - \bar{f}_b}{\sqrt{\hat{V}ar(f_a) + \hat{V}ar(f_b)}} \rightarrow N(0; 1)$$

É sugerido o uso de $n_b = 0,1n$ e $n_a = 0,5n$ e estimadores da densidade espectral são usados para as variâncias. Por fim, valores grandes para z_G indicam falta de convergência mas valores pequenos não implicam convergência. Uma preocupação deve ocorrer para valores maiores que 2 em termos absolutos.

Outra forma de avaliação da convergência apresentada pelos mesmos autores é o

diagnóstico de uma sequência de cadeias, denominado Diagnóstico de Sequência Múltipla de Gelman e Rubin. Este diagnóstico é baseado na comparação de um conjunto de cadeias com diferentes pontos que são super dispersos com relação à distribuição *a posteriori*. Baseia-se também na teoria da aproximação gaussiana para *as posteriori* marginais usando testes como ANOVA e diagnóstico com distribuição *t-Student*.

Para a realização deste teste considera-se m cadeias paralelas de comprimento n :

$$\begin{array}{cccc} \theta_{(1)}^{[1]} & , & \theta_{(1)}^{[2]} & , \dots , \theta_{(1)}^{[n]} \\ \theta_{(2)}^{[1]} & , & \theta_{(2)}^{[2]} & , \dots , \theta_{(2)}^{[n]} \\ \vdots & & & \\ \theta_{(m)}^{[1]} & , & \theta_{(m)}^{[2]} & , \dots , \theta_{(m)}^{[n]} \end{array}$$

As variâncias entre cadeias, V_E , e dentro das cadeias, V_D , são dadas por

$$V_E = \frac{n}{m-1} \sum_{j=1}^m (\bar{\theta}_{(j)} - \bar{\bar{\theta}})^2 \text{ e } V_D = \frac{1}{m(n-1)} \sum_{j=1}^m \sum_{i=1}^n (\bar{\theta}_{(j)}^{[i]} - \bar{\theta}_{(j)})^2$$

em que $\bar{\bar{\theta}}$ é a média das médias das cadeias já que todas tem o mesmo comprimento e $\bar{\theta}_{(j)}$ é a média dos n valores da j -ésima cadeia. Sob convergência, todos os mn valores são selecionados da distribuição *a posteriori* e a variância $Var(\theta)$ pode ser estimada por

$$\hat{V}ar(\theta) = \left(1 + \frac{1}{n}\right) V_D + \left(\frac{1}{n}\right) V_E.$$

Se as cadeias ainda não convergiram, os valores iniciais estarão influenciando as trajetórias. Devido às suas superdispersão, os valores forçarão $\hat{V}ar(\theta)$ a superestimar $Var(\theta)$. Por outro lado, após a convergência, V_D tenderá a subestimar $Var(\theta)$ porque cada cadeia ainda não terá atravessado todo o espaço. Desta forma, um indicador de convergência é o fator *redução de escala estimado*:

$$\sqrt{\hat{F}_{RE}} = \sqrt{\frac{\hat{V}ar(\theta)}{V_D}}.$$

De acordo com a literatura, valores de $\sqrt{\hat{F}_{RE}}$ próximos a 1, mais precisamente inferiores a 1,2 são aceitáveis para convergência.

2.3.4 Intervalo de Credibilidade

As distribuições de probabilidade *a posteriori* dos parâmetros no vetor $\vec{\theta}$ contêm toda a informação sobre estes parâmetros. Resumos estatísticos destas distribuições

podem ser calculados como média, mediana e moda representando medidas de locação e a variação pode ser resumida calculando-se o desvio padrão e quantis. Segundo Gelman et al. (2003) as médias, por exemplo, são as esperanças *a posteriori* dos parâmetros, as modas podem ser interpretadas como os únicos valores “mais prováveis”, uma vez conhecidos os dados e o modelo. Mas além disso, é importante descrever a incerteza *a posteriori*. Para isto, é usual apresentar “intervalos de credibilidade” para os parâmetros. Estes intervalos podem ser obtidos de diferentes formas sendo as mais usuais intervalos baseados em quantis das distribuições *a posteriori* ou um intervalo central da probabilidade *a posteriori*, que corresponde no caso de um intervalo de $100(1 - \alpha)\%$, ao intervalo dos valores acima e abaixo do qual encontra-se exatamente $100(\alpha/2)\%$ da probabilidade *a posteriori*. Estes intervalos, denominados intervalo central *a posteriori* ou intervalo de credibilidade são o análogo a intervalo de confiança na teoria clássica de inferência. Carlin e Louis (2009) definem um intervalo de credibilidade de $100(1 - \alpha)\%$ para $\vec{\theta}$ como um subconjunto C de Θ tal que

$$P(C|\vec{Y}) = \int_C P(\vec{\theta}|\vec{Y})d\vec{\theta} \geq 1 - \alpha.$$

em que o sinal “ \geq ” é usado para considerar conjuntos discretos nos quais pode não ser possível a obtenção de um intervalo de cobertura de exatamente $1 - \alpha$ de probabilidade.

Gelman et al. (2003) apresentam uma outra maneira de resumir a incerteza *a posteriori* que é calcular uma região de alta densidade *a posteriori*. Esta região corresponde a região de valores que contém $100(1 - \alpha)\%$ da probabilidade *a posteriori*, tem a característica que a densidade dentro da região nunca é menor que a densidade fora da região e será idêntica ao intervalo *a posteriori* se a distribuição *a posteriori* é unimodal e simétrica. Carlin e Louis (2009) definem um intervalo de credibilidade de *mais alta densidade a posteriori* como

$$C = \vec{\theta} \in \Theta : P(\vec{\theta}|\vec{Y}) \geq k(\alpha)$$

em que $k(\alpha)$ é a maior constante satisfazendo

$$P(C|\vec{Y}) \geq 1 - \alpha.$$

2.4 Dados Composicionais

A análise de dados composicionais foi sistematizada por Aitchison (1982). Tais dados consistem de vetores \vec{X} de proporções de algum “todo” e apresentam variabilidade de vetor para vetor. Cada vetor é denominado uma *composição* e os componentes de

qualquer composição de B partes (X_1, X_2, \dots, X_B) devem satisfazer as exigências de que cada componente é não negativo

$$X_1 \geq 0, \dots, X_B \geq 0,$$

e que a soma de todos os componentes é 1

$$X_1 + X_2 + \dots + X_B = 1. \quad (2.31)$$

Aitchison (1986) apresenta duas maneiras de se determinar uma composição. Na primeira, a composição fica completamente especificada pelos componentes de um subvetor de $(B-1)$ -partes onde $(X_1, X_2, \dots, X_{B-1})$,

$$X_B = 1 - X_1 - X_2 - \dots - X_{B-1},$$

o que significa que uma composição de B -partes é um vetor $(B-1)$ -dimensional, podendo ser representado em algum conjunto $(B-1)$ -dimensional. Na segunda maneira, especifica-se $B-1$ razões r_i por

$$r_i = X_i/X_B, \quad i = 1, \dots, B-1,$$

e a composição é determinada por

$$\begin{aligned} X_i &= r_i/(r_1 + \dots + r_{B-1} + 1), \quad i = 1, \dots, B-1; \\ X_B &= 1/(r_1 + \dots + r_{B-1} + 1). \end{aligned}$$

Aitchison e Greenacre (2002) apresentam três maneiras equivalentes de se considerar razões dentro de uma composição:

- a) as $\frac{1}{2}B(B-1)$ razões X_i/X_j entre os componentes i e j , $i < j$;
- b) as $B-1$ razões X_i/X_B , $i = 1, 2, \dots, B-1$ componentes;
- c) as B razões $X_i/g(\vec{X})$ entre os componentes e sua média geométrica $g(\vec{X}) = \sqrt[B]{X_1 X_2 \dots X_B}$.

Razões como as do item (b) foram adotadas neste trabalho.

As aplicações com dados composicionais são frequentemente encontrados em áreas como Geologia, Agricultura, Biologia, Literatura, Meio Ambiente, Estatística Médica, Economia, dentre outras (LABUS, 2005). As características principais de um conjunto de dados composicionais, segundo Reyment e Savazzi (1999) *apud* Labus (2005), são:

- podem ser representados na forma de uma matriz, na qual cada coluna da matriz representa uma única parte (variável) e as linhas da matriz somam 1, no caso de proporções; 100% no caso de percentagens; ou alguma outra constante, de acordo com a forma particular de representação dos dados adotada pelo pesquisador como, por exemplo, unidades ppm (partes por milhão, equivalente a 1 miligrama por litro), ppb (partes por bilhão, equivalente a 1 micrograma por litro) ou outra unidade de concentração.

- os coeficientes de correlação amostrais mudam se uma das variáveis (partes) é excluída da matriz de dados, e as linhas somam 1 ou 100 novamente. O mesmo ocorre se um novo componente é adicionado.

Essa última propriedade significa que alterar uma ou mais variáveis do conjunto de dados pode ter um efeito numericamente significativo nas correlações entre as variáveis restantes.

Pawlowsky-Glahn e Olea (2004) e Tolosana-Delgado, Otero e Pawlowsky-Glahn (2005) apontam que os dados composicionais apresentam um efeito de correlação espúria, o que significa que a aplicação dos métodos estatísticos padrão pode levar a resultados inconsistentes. Isso significa que as covariâncias estão sujeitas a controles não estocásticos, isto é, sofrem distorções devido à restrição da soma totalizar 1, levando à interpretação errônea da estrutura de covariância espacial. Um outro problema, levantado por Labus (2005), é o fato de que os componentes não são independentes, o que é uma consequência da Equação (2.31). Esse fato, segundo Pawlowsky-Glahn e Olea (2004), implica em singularidade da matriz de covariância de uma composição, excluindo, por exemplo, o uso de técnicas de estimação como cokrigagem de todos os componentes.

Pawlowsky-Glahn e Olea (2004) relatam que os problemas com correlação espacial espúria e singularidade da matriz de covariância estão relacionados à suposição básica de que o espaço amostral é irrestrito, implícita na análise estatística de corregeionalizações e, à suposição de que a distribuição do erro de estimação em cada ponto da região amostral é gaussiana.

Dessa forma, surge a necessidade de expandir adequadamente os métodos de análise de dados composicionais para acomodar estruturas espaciais ou vice-versa: expandir os métodos de análise espacial para acomodar dados composicionais.

2.4.1 Composição Regionalizada

Uma composição regionalizada é um vetor função aleatória $\vec{X}(\vec{x})$, com $\vec{x} \in A \subset \mathbb{R}^n$, que satisfaz as seguintes condições em cada localização \vec{x} (PAWLOWSKY-GLAHN; OLEA, 2004):

- i) $X_1(\vec{x}) > 0, X_2(\vec{x}) > 0, \dots, X_B(\vec{x}) > 0$;
- ii) $X_1(\vec{x}) + X_2(\vec{x}) + \dots + X_B(\vec{x}) = 1$.

O caso $X_i(\vec{x}) = 0, i = 1, \dots, B$, foi omitido por simplificação da apresentação. Os procedimentos para tratar essa situação podem ser encontrados em Pawlowsky-Glahn e Olea (2004, Seção 6.1).

O espaço amostral natural para uma composição é o simplex. McBratney, De Gruijter e Brus (1992) *apud* Odeh, Tood e Triantafilis (2003) definem um simplex como uma representação geométrica do espaço de atributos, onde uma composição de B partes é representada por um número mínimo de vértices em um espaço de um dado número

de dimensões. Em outras palavras, o espaço amostral para $\vec{X}(\vec{x})$ é o simplex- B unitário embutido no espaço real B -dimensional \mathbb{R}^B , dado por:

$$\mathbb{S}^B = \{\vec{X}(\vec{x}) \in \mathbb{R}^B; X_i(\vec{x}) > 0, i = 1, \dots, B; \vec{1}'\vec{X}(\vec{x}) = 1\},$$

sendo $\vec{1}$ ou seu transposto, um vetor com B elementos iguais a 1.

2.4.2 Representação Gráfica

O *diagrama ternário*, triângulo cujos vértices representam os três componentes da composição (BUTLER; GLASBEY, 2008), é uma técnica gráfica adotada neste trabalho para a compreensão de dados composicionais. A descrição de outras técnicas gráficas, como os diagramas de variação, diagramas ternários, diagramas de dispersão de razão e log razão, gráfico bivariado composicional e diagrama de dispersão das coordenadas podem ser encontradas, por exemplo, em Aitchison e Egozcue (2005).

De acordo com Pawlowsky-Glahn, Egozcue e Tolosana-Delgado (2007), para composições de três componentes, o diagrama ternário é uma forma de visualização do simplex que é um triângulo equilátero, ou seja, é uma representação equivalente. Com o objetivo de simplificar a exposição, a notação espacial é omitida nesta Seção, mas os conceitos apresentados a seguir são igualmente válidos para o caso espacial.

Para construir um diagrama ternário, considere o triângulo de altura $h = 1$, lados de comprimento L e vértices A , F e M , como mostrado na Figura 2.1, que foi feita no ambiente estatístico R (R development Core Team, 2008), a partir de uma modificação da função *ternary* (VENABLES; RIPLEY, 2002) do pacote *MASS*. De acordo com esses autores, para uma composição (X_1, X_2, X_3) no simplex, as expressões das correspondentes coordenadas $(u; v)$ no sistema de coordenadas cartesianas bidimensional, \mathbb{R}^2 , são dadas por:

$$(u; v) = ((X_3 - X_2) \frac{1}{\sqrt{3}}; X_1), \quad (2.32)$$

e adotadas neste trabalho. Assim, os pontos com coordenadas $A = (100, 0, 0)$, $F = (0, 100, 0)$ e $M = (0, 0, 100)$ no simplex correspondem aos pontos $A = (0; 1)$, $F = (-\frac{1}{\sqrt{3}}; 0)$ e $M = (\frac{1}{\sqrt{3}}; 0)$ em \mathbb{R}^2 .

Para qualquer ponto $p = (X_1, X_2, X_3)$ no triângulo da Figura 2.1, os comprimentos das perpendiculares do ponto aos eixos satisfazem $X_1 + X_2 + X_3 = 1$, ou seja, p é marcado a uma distância X_1 do lado oposto ao vértice A , a uma distância X_2 do lado oposto ao vértice F e a uma distância X_3 do lado oposto ao vértice M .

De fato, como os ângulos internos de um triângulo equilátero são iguais a 60° , $\widehat{AMH} = 60^\circ$, de modo que $\text{sen}(60^\circ) = h/L$, a altura é $h = L(\sqrt{3}/2)$ e, se o triângulo AHM tem altura unitária, $L = 2/\sqrt{3}$. Considerando a composição $(40, 40, 20)\%$, de acordo com o triângulo de classificação textural, o valor do componente F é 40%, e esse valor é representado pelo ponto I . Então, a relação de correspondência entre o triângulo de

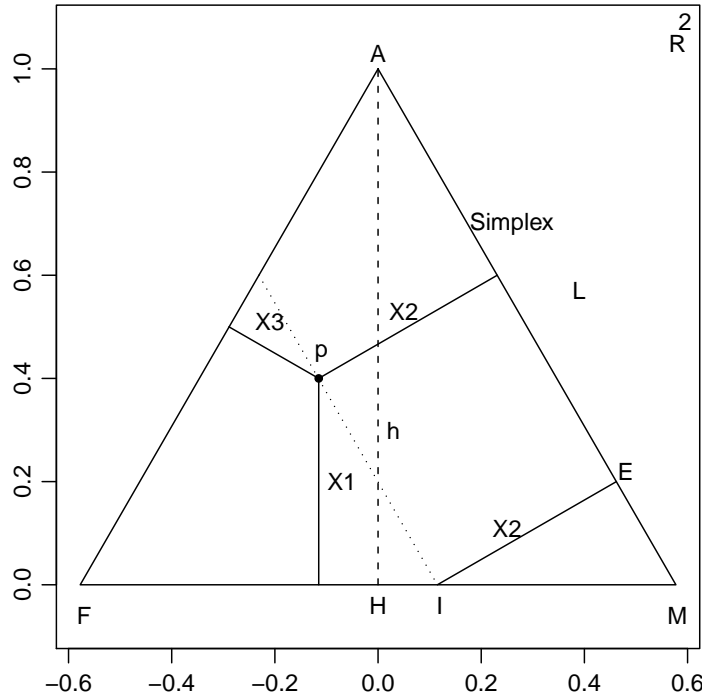


Figura 2.1: O ponto p , com coordenadas $(40,40,20)\%$ em \mathbb{S}^3 e $(-0,115;0,40)$ em \mathbb{R}^2 , representado no triângulo de altura unitária.

classificação textural e o diagrama ternário é que a distância do ponto I ao ponto M (\overline{IM}) corresponde a 40% do lado L, ou seja, $\overline{IM} = 0,40(2/\sqrt{3})$. Logo, como o ângulo \widehat{EMI} é igual a 60° , tem-se

$$\text{sen}(60^\circ) = \frac{X_2}{\overline{IM}} \Rightarrow X_2 = 0,40.$$

Portanto, os comprimentos das perpendiculares correspondem aos valores dos componentes e satisfazem $X_1 + X_2 + X_3 = 1$. Por consequência, pode-se observar, através da Figura 2.1, que todas as paralelas ao segmento de reta de comprimento X_2 , no interior do triângulo, têm o mesmo comprimento 40%.

Uma maneira alternativa de determinar as coordenadas $(\mathbf{u};\mathbf{v})$ é considerar o triângulo com lado unitário (Figura 2.2) e $(\mathbf{u}_0;\mathbf{v}_0)$, as coordenadas do vértice F com escala escolhida de forma a centrar o triângulo no sistema de coordenadas cartesianas, como em Pawlowsky-Glahn, Egozcue e Tolosana-Delgado (2007). Dessa forma, as expressões das coordenadas do vértice M em \mathbb{R}^2 são $(\mathbf{u}_0 + 1;\mathbf{v}_0)$. Pelo teorema de Pitágoras, $h = \sqrt{3}/2$ e, conseqüentemente, $A = (\mathbf{u}_0 + 0,5;\mathbf{v}_0 + \sqrt{3}/2)$. As coordenadas $(\mathbf{u};\mathbf{v})$ são obtidas como uma combinação linear convexa das coordenadas dos vértices:

$$(\mathbf{u},\mathbf{v}) = \frac{1}{100}(X_1A + X_2F + X_3M).$$

Escolhendo $(\mathbf{u}_0;\mathbf{v}_0) = (0,2;0,2)$ e considerando a composição $(40,40,20)\%$, as correspondentes coordenadas em \mathbb{R}^2 são $(0,60;0,55)$.

No diagrama ternário, pode-se observar, ainda, que todos os pontos em uma linha reta partindo de um dos vértices têm razões dos outros dois componentes iguais. Na

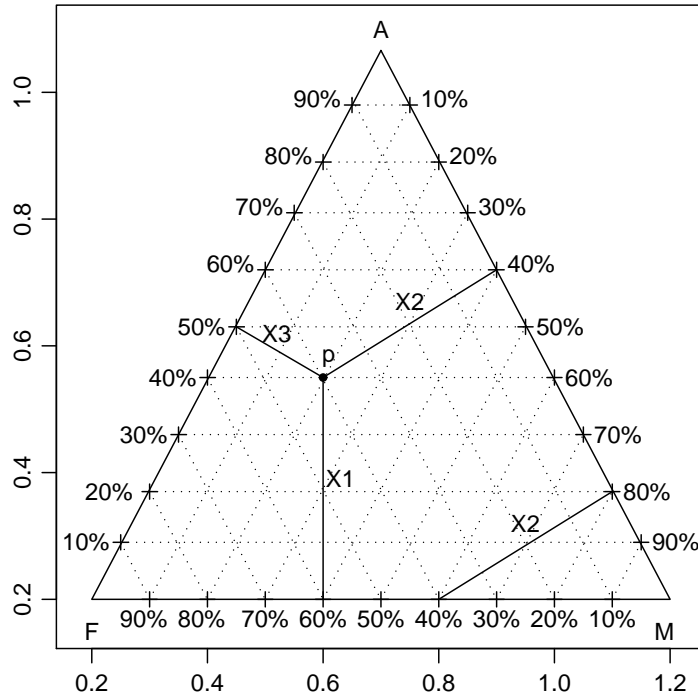


Figura 2.2: O ponto p , com coordenadas $(40, 40, 20)\%$ em \mathbb{S}^3 e coordenadas $(0, 60; 0, 55)$ em \mathbb{R}^2 , representado no triângulo com lado unitário e triângulo de classificação textural.

Figura 2.3, por exemplo, considera-se a linha partindo do vértice A passando pelos pontos a , b , c e d . Para todos os pontos desta linha, fixando um valor para o componente X_1 e escolhendo um valor para a razão, r_A , pode-se determinar X_2 e X_3 , pois as composições têm a forma $(X_1, r_A X_3, X_3)$, satisfazendo

$$X_1 + r_A X_3 + X_3 = 1. \quad (2.33)$$

Supondo $r_A = 1$ e $X_1 = 0,80$, por exemplo, da Equação (2.33) obtém-se $X_3 = 0,10$. Daí, $X_2 = 0,10$ e a composição é $(80, 10, 10)\%$. Aplicando a Equação (2.32), as correspondentes coordenadas dessa composição em \mathbb{R}^2 são $(0; 0,80)$. Alterando o valor da razão para 1,5, depois para 4 e utilizando o mesmo raciocínio, obtém-se pontos em outras duas linhas (Figura 2.3), cujas coordenadas são apresentadas na Tabela 2.1. Da mesma maneira, considerando linhas partindo do vértice F , as composições são da forma $(X_1, X_2, r_F X_1)$, para uma razão r_F dos componentes X_1 e X_3 , e satisfazem

$$X_1 + X_2 + r_F X_1 = 1. \quad (2.34)$$

Para $r_F = 1$ e $X_2 = 0,80$, da Equação (2.34) obtém-se $X_3 = 0,15$ e, em seguida, $X_1 = 0,05$, e a composição é $(5, 80, 15)\%$. Da Equação (2.32), as correspondentes coordenadas dessa composição em \mathbb{R}^2 são $(-0,375; 0,05)$. O mesmo raciocínio é válido para os outros pontos nessa linha, na linha correspondente à razão 3. Os resultados encontram-se na Tabela 2.2.

A interpretação de um diagrama ternário é feita de modo que pontos localizados próximo a um vértice indicam altas proporções do componente associado àquele vértice,

Tabela 2.1: Pontos nas linhas que partem do vértice A com respectivas coordenadas em \mathbb{S}^3 , \mathbb{R}^2 e razão dos componentes X_2 e X_3 .

Ponto	Composição	Coord. \mathbb{R}^2	Razão	Ponto	Composição	Coord. \mathbb{R}^2	Razão
a	(80, 10, 10)	(0;0, 80)	1	e	(50, 30, 20)	(-0,06;0,50)	1,5
b	(60, 20, 20)	(0;0, 60)	1	f	(20, 48, 32)	(-0,09;0,20)	1,5
c	(40, 30, 30)	(0;0, 40)	1	g	(0, 60, 40)	(-0,115;0)	1,5
d	(20, 40, 40)	(0;0, 20)	1				
Ponto	Composição	Coord. \mathbb{R}^2	Razão				
i	(50, 40, 10)	(-0,17;0,50)	4				
j	(0, 80, 20)	(-0,346;0)	4				

Tabela 2.2: Pontos nas linhas que partem do vértice F com respectivas coordenadas em \mathbb{S}^3 , \mathbb{R}^2 e razão dos componentes X_1 e X_3 .

Ponto	Composição	Coord. \mathbb{R}^2	Razão	Ponto	Composição	Coord. \mathbb{R}^2	Razão
m	(20, 60, 20)	(-0,23;0,20)	1	q	(5, 15, 80)	(-0,375;0,05)	3
n	(30, 40, 30)	(-0,06;0,30)	1	s	(17, 5; 30, 52, 5)	(0, 129;0, 175)	3
o	(40, 20, 40)	(0, 115;0,40)	1				

enquanto pontos localizados no centro do triângulo tem proporções iguais para todos os três componentes. Para exemplificar, a Figura 2.4 mostra um diagrama ternário apresentado em Aitchison (1986), baseado em dados de areia, silte e argila. Nessa amostra, observa-se que os menores valores ocorreram para o componentes argila. O ponto vermelho representa o centro da distribuição, conforme Subseção 2.4.8.

De acordo com Aitchison (1986), para avaliar a extensão da variabilidade da razão entre dois componentes, deve-se traçar linhas retas partindo do vértice complementar a esses dois componentes, passando por todos os pontos. A amplitude da intersecção dessas linhas com o eixo oposto ao vértice complementar resulta na extensão da variabilidade. Por exemplo, na Figura 2.4, verifica-se que a variabilidade da razão Areia/Silte indicada pelo segmento a/b é maior que a da razão Argila/Silte indicada pelo segmento c/d. Interpretações semelhantes são feitas para os outros pares de razões de componentes.

2.4.3 Base Regionalizada

Uma base regionalizada $\vec{W}(\vec{x})$, $\vec{x} \in A \subset \mathbb{R}^n$, é um vetor função aleatória cujos componentes são todos positivos e medidos na mesma escala (PAWLOWSKY-GLAHN; OLEA, 2004). Isso implica que o espaço amostral de uma base de B -partes é dado por:

$$\mathbb{R}_+^B = \{\vec{W}(\vec{x}) \in \mathbb{R}^B; W_i(\vec{x}) > 0, i = 1, \dots, B\}.$$

A transformação de uma base regionalizada em uma composição regionalizada é

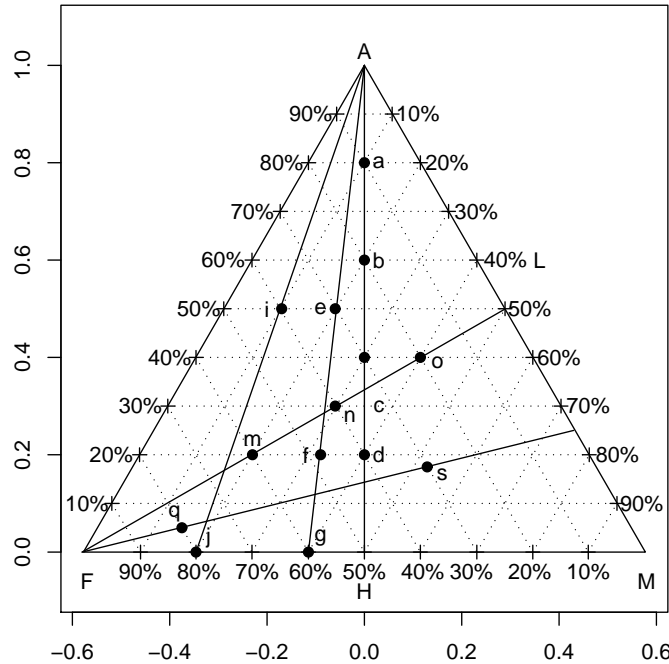


Figura 2.3: Pontos pertencentes às mesmas linhas no diagrama ternário com a característica de apresentarem razões iguais dos componentes complementares ao vértice A e ao vértice F.

feita através do operador fechamento, \mathcal{C} , definido por:

$$\mathcal{C} : \mathbb{R}_+^B \longrightarrow \mathbb{S}^B$$

$$\vec{W}(\vec{x}) \longrightarrow \mathcal{C}(\vec{W}(\vec{x})) = \frac{\vec{W}(\vec{x})}{\mathbf{1}'\vec{W}(\vec{x})},$$

garantindo soma unitária dos componentes. Ou seja, fazendo

$$X_i(\vec{x}) = \frac{W_i(\vec{x})}{W_1(\vec{x}) + W_2(\vec{x}) + \dots + W_B(\vec{x})}, \quad i = 1, \dots, B,$$

obtém-se a composição $\vec{X}(\vec{x}) = (X_1(\vec{x}), X_2(\vec{x}), \dots, X_B(\vec{x}))'$ e, segundo Pawlowsky-Glahn e Olea (2004), a relação $\frac{W_i(\vec{x})}{W_j(\vec{x})} = \frac{X_i(\vec{x})}{X_j(\vec{x})}$ é válida para quaisquer índices $i, j = 1, 2, \dots, B$, para $W_j(\vec{x})$ e $X_j(\vec{x}) \neq 0$, enfatizando que o dado composicional contém informação somente sobre magnitudes relativas e não absolutas.

Barceló-Vidal, Martín-Fernández e Pawlowsky-Glahn (2001) apresentam uma interpretação geométrica para o operador fechamento \mathcal{C} . Considerando como exemplo uma composição com três componentes, pode-se entender \mathcal{C} como a intersecção do raio partindo da origem através de $\vec{W}(\vec{x})$ e o hiperplano de \mathbb{R}^3 definido pela equação $W_1(\vec{x}) + W_2(\vec{x}) + W_3(\vec{x}) = 1$, como na Figura 2.5. O conjunto de todos estes pontos é o

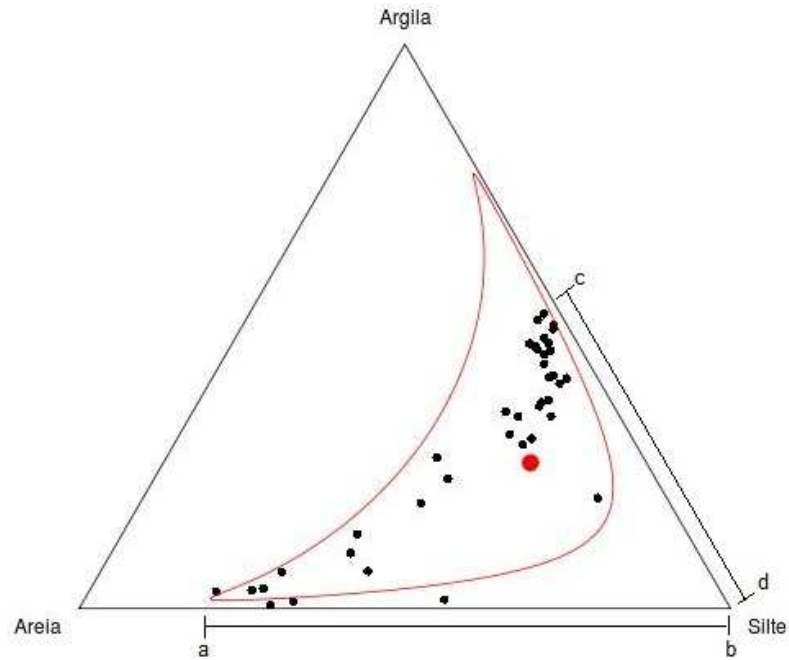


Figura 2.4: Diagrama ternário para os dados do Lago Ártico, incluindo o centro da distribuição e região 2-desvios-padrão de confiança.

simplex regular dado por:

$$\mathbb{S}^3 = \{(W_1(\vec{x}), W_2(\vec{x}), W_3(\vec{x}))'; W_1(\vec{x}) > 0, W_2(\vec{x}) > 0, W_3(\vec{x}) > 0; W_1(\vec{x}) + W_2(\vec{x}) + W_3(\vec{x}) = 1\}.$$

2.4.4 Subcomposição Regionalizada

Uma subcomposição regionalizada é um subconjunto de uma composição regionalizada. Assim, considerando-se $\vec{X}(\vec{x})$ uma composição regionalizada de B partes e $s \subset \{1, 2, \dots, B\}$, de forma a tornar $\vec{X}_s(\vec{x})$ um subvetor cujos elementos são os componentes de $\vec{X}(\vec{x})$ correspondentes às partes em s , uma subcomposição regionalizada será:

$$\mathcal{C}(\vec{X}_s(\vec{x})) = \frac{\vec{X}_s(\vec{x})}{\mathbf{1}'\vec{X}_s(\vec{x})}.$$

Segundo Aitchison (1986), Aitchison e Greenacre (2002) e Aitchison e Egozcue (2005), o conceito de subcomposição é importante no que se refere a coerência subcomposicional. Isso significa que, se um pesquisador, tendo acesso à uma composição, também fizer inferência a partir de uma subcomposição desta e, se um outro, com acesso somente a uma subcomposição, mas com partes comuns à do primeiro pesquisador, fizer inferência a partir dessa, os dois resultados deverão coincidir.

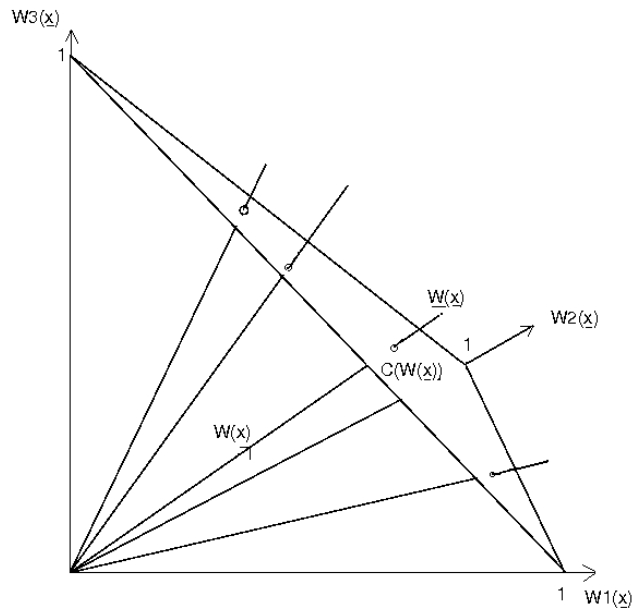


Figura 2.5: Composições de 3 partes como raios partindo da origem em \mathbb{R}_+^3 .
 FONTE: Adaptado de Barceló-Vidal, Martín-Fernández e Pawlowsky-Glahn (2001).
 NOTA: Em particular, nessa figura, $\vec{W}(\vec{x}) = \{kW(\vec{x}) : k \in \mathbb{R}^+\}$.

As correlações momento-produto e a análise de componentes principais baseadas em covariâncias calculadas no dado composicional em linha não tem coerência subcomposicional. A característica importante de uma subcomposição é a preservação das relações de razões. Daí, se $\vec{s}(\vec{x}) = \mathcal{C}(\vec{X}_s(\vec{x}))$, então $\frac{s_i(\vec{x})}{s_j(\vec{x})} = \frac{X_i(\vec{x})}{X_j(\vec{x})}$, $i, j \in s$. Observa-se que as razões são formadas com os componentes do vetor de dado composicional (linha) através das colunas da matriz de dados.

Da mesma forma que na Subseção 2.4.3, Barceló-Vidal, Martín-Fernández e Pawlowsky-Glahn (2001) apresentam uma interpretação geométrica para uma subcomposição. Novamente, considerando como exemplo uma composição de 3-partes, a formação de uma subcomposição corresponde à projeção ortogonal do raio associado a $\vec{X}(\vec{x})$ em \mathbb{R}^+ em um subespaço gerado pelos eixos coordenados das partes selecionadas para formar a subcomposição (Figura 2.6).

2.4.5 Amalgamação e Partição Regionalizada

Seja $\vec{X}(\vec{x}) = (X_1(\vec{x}), X_2(\vec{x}), \dots, X_B(\vec{x}))'$ uma composição regionalizada de B partes, dividida em C ($C \leq B$) subconjuntos mutuamente exclusivos. Seja $\vec{X}_i(\vec{x})$, $i = 1, \dots, C$, o vetor cujos componentes são os elementos do i -ésimo subconjunto. Pawlowsky-Glahn e Olea (2004) definem *amalgamação regionalizada* como a composição regionalizada com C

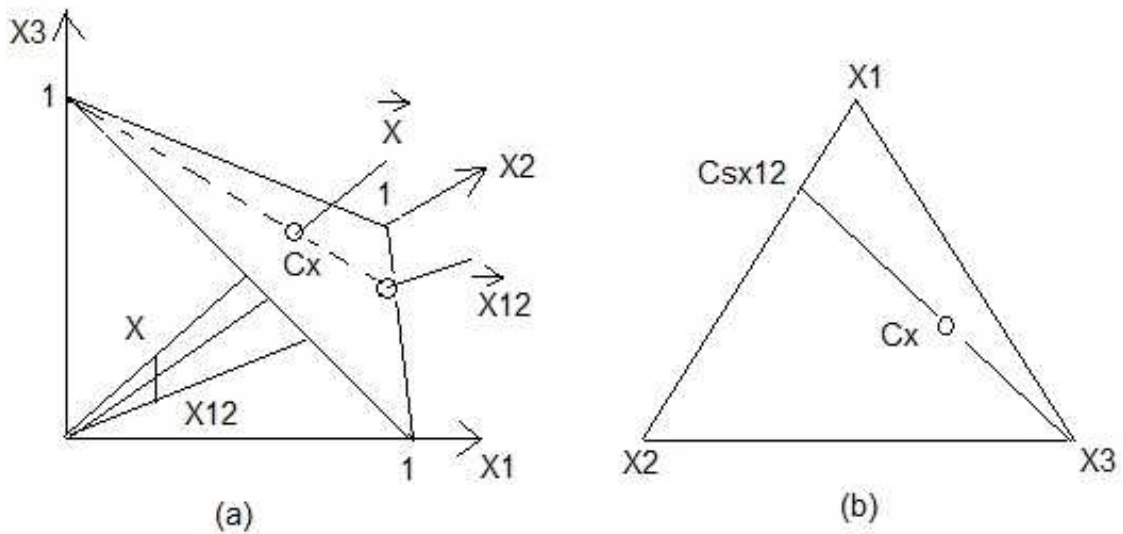


Figura 2.6: Interpretação geométrica da formação da subcomposição $\vec{X}_{12}(\vec{x})$ da composição $\vec{X}(\vec{x})$: (a) em \mathbb{R}_+^3 ; (b) em \mathbb{S}^3 .

FONTE: Adaptado de Barceló-Vidal, Martín-Fernández e Pawlowsky-Glahn (2001).

NOTA: Em particular, nessa figura, $\vec{X}(\vec{x}) = \{kX(\vec{x}) : k \in \mathbb{R}^+\}$.

componentes, na qual os componentes de cada um dos C subconjuntos são somados:

$$\vec{A}(\vec{x}) = (A_1(\vec{x}), A_2(\vec{x}), \dots, A_C(\vec{x}))',$$

com $A_i(\vec{x}) = \vec{1}'\vec{X}_i(\vec{x})$.

Ao considerar a amalgamação regionalizada $\vec{A}(\vec{x})$ juntamente com as subcomposições regionalizadas, $\vec{s}_i(\vec{x}) = \mathcal{C}(\vec{X}_i(\vec{x}))$, $i = 1, \dots, C$:

$$P_C(\vec{X}(\vec{x})) = (\vec{s}_1(\vec{x}), \vec{s}_2(\vec{x}), \dots, \vec{s}_C(\vec{x}); \vec{A}(\vec{x})),$$

tem-se definida a partição regionalizada de ordem C de $\vec{X}(\vec{x})$. Assim, as informações contidas nos subvetores são preservadas em uma partição.

Segundo Pawlowsky-Glahn e Olea (2004) uma amalgamação regionalizada é entendida como uma mistura e combinação de vários componentes. Aitchison (1986) exemplifica a idéia no contexto de uma enquete sobre gastos de uma dona de casa, onde a composição $\vec{X}(\vec{x})$ de proporções das despesas é formada pelos nove componentes

$X_1(\vec{x})$: Gêneros alimentícios,	$X_6(\vec{x})$: Bens duráveis,
$X_2(\vec{x})$: Habitação,	$X_7(\vec{x})$: Bens diversos,
$X_3(\vec{x})$: Combustível e luz,	$X_8(\vec{x})$: Transporte e veículos,
$X_4(\vec{x})$: Cigarro e álcool,	$X_9(\vec{x})$: Serviços.
$X_5(\vec{x})$: Vestuário e calçado,	

O interesse pode estar em grupos de mercadorias, como na amalgamação $\vec{A}(\vec{x}) = (A_1(\vec{x}), A_2(\vec{x}), A_3(\vec{x}), A_4(\vec{x}))$, por exemplo, onde

- $A_1(\vec{x}) = (X_2(\vec{x}) + X_3(\vec{x})),$
- $A_2(\vec{x}) = (X_1(\vec{x}) + X_4(\vec{x})),$
- $A_3(\vec{x}) = (X_5(\vec{x}) + X_6(\vec{x}) + X_7(\vec{x})),$
- $A_4(\vec{x}) = (X_8(\vec{x}) + X_9(\vec{x})).$

2.4.6 Transformação Logística Modificada

Graf (2006) expõe de forma clara que a restrição de que a soma dos valores dos componentes é a unidade implica que existe uma correlação negativa entre os componentes. Isto mostra que as correlações entre os pares de componentes não são diretamente interpretáveis. Para relaxar ou evitar essa restrição, Aitchison (1986) propôs transformações que generalizam a transformação logística para um vetor composicional de duas partes fazendo com que o vetor resultante não seja mais restrito. Consequentemente, as correlações passam a ser interpretáveis. O autor afirma que a transformação log-razão é viável para investigar a importância ou irrelevância de componentes individuais, por fornecer uma estrutura de dependência correta e interpretável para descrever os padrões reais de variabilidade composicional e subcomposicional. A natureza essencial de uma composição é que as magnitudes relativas dos componentes são as unidades relevantes sob estudo. Essas magnitudes relativas (ou razões) garantem tratabilidade algébrica e interpretação estatística das quantidades (AITCHISON, 1999). Algumas transformações logísticas, como a aditiva, a aditiva modificada, a multiplicativa e a híbrida, podem ser encontradas em Aitchison (1982), Aitchison et al. (2000) e Odeh, Tood e Triantafilis (2003), por exemplo.

No contexto de composições regionalizadas, Pawlowsky-Glahn, Olea e Davis (1995), Pawlowsky-Glahn e Olea (2004) definem a transformação *log razão aditiva* (ALR) como a função

$$\begin{aligned} \text{alr} : \mathbb{S}^B &\longrightarrow \mathbb{R}^{B-1} \\ \vec{X}(\vec{x}) &\longrightarrow \text{alr}(\vec{X}(\vec{x})) = \left(\ln \left(\frac{X_1(\vec{x})}{X_B(\vec{x})} \right), \dots, \ln \left(\frac{X_{B-1}(\vec{x})}{X_B(\vec{x})} \right) \right)'. \end{aligned}$$

Pode-se observar que, para $B = 2$, a transformação ALR torna-se a transformação logística convencional. Nota-se também que, se a composição possuir três componentes, após a transformação a composição passará a ter dois componentes, ou seja, será um vetor bivariado. Então, um vetor auxiliar $\vec{X}^*(\vec{x}) = (X_1(\vec{x}) \ X_2(\vec{x}) \ \dots \ X_{B-1}(\vec{x}) \ , \ 0)$ é introduzido para definir a transformação inversa, denominada transformação *logística generalizada aditiva* (AGL), que traz os dados de volta ao simplex:

$$\begin{aligned} \text{agl} : \mathbb{R}^{B-1} &\longrightarrow \mathbb{S}^B \\ \vec{Y}(\vec{x}) &\longrightarrow \vec{X}(\vec{x}) = \mathcal{C} \left(\left(\exp \left\{ \ln \left(\frac{X_1(\vec{x})}{X_B(\vec{x})} \right) \right\}, \dots, \exp \{0\} \right) \right)'. \end{aligned}$$

Se a restrição de soma constante é $c \neq 0$, os dados no espaço simplex são obtidos fazendo-se

$$\vec{X}(\vec{x}) = c \cdot \text{agl}(\vec{Y}(\vec{x})).$$

Outra transformação viável é a transformação *log razão centrada* (CLR), definida como

$$\begin{aligned} \text{clr} : \mathbb{R}_+^B &\longrightarrow \mathbb{R}^B \\ \vec{W}(\vec{x}) &\longrightarrow \text{clr}(\vec{W}(\vec{x})) = \ln \left(\frac{\vec{W}(\vec{x})}{g(\vec{W}(\vec{x}))} \right), \end{aligned}$$

ou

$$\begin{aligned} \text{clr} : \mathbb{S}^B &\longrightarrow \mathbb{R}^B \\ \vec{X}(\vec{x}) &\longrightarrow \text{clr}(\vec{X}(\vec{x})) = \ln \left(\frac{\vec{X}(\vec{x})}{g(\vec{X}(\vec{x}))} \right), \end{aligned}$$

onde $g(\vec{W}(\vec{x})) = \sqrt[B]{\prod_{i=1}^B W_i(\vec{x})}$ é a média geométrica dos componentes da base regionalizada $\vec{W}(\vec{x})$ e $g(\vec{X}(\vec{x})) = \sqrt[B]{\prod_{i=1}^B X_i(\vec{x})}$ é a média geométrica dos componentes da composição regionalizada $\vec{X}(\vec{x})$.

Aitchison (1986) apresenta um teste para avaliar a suposição de normalidade logística das composições log-razões, mas seu uso limita-se a observações independentes. O teste baseia-se em Andrews, Gnadesikan e Warner (1973) e nas recomendações de Stephens (1974) para o uso de testes da função de distribuição empírica, nas formas de Anderson-Darling, Cramer-von Mises e Watson. Considera-se todas as $B - 1$ distribuições marginais univariadas das composições log-razão.

Para a realização do teste de Aitchison, considera-se que, para a j -ésima distribuição marginal da composição log razão aditiva, as observações são

$$y_{ij} = \ln \frac{x_{ij}}{x_{iB}}, \quad i = 1, \dots, n; \quad j = 1, B - 1.$$

Usando $\hat{\mu}_j$ e $\hat{\sigma}_j^2$, as estimativas da média e variância marginais, calculam-se os valores $z_r = (y_{ij} - \hat{\mu}_j) / \sqrt{\hat{\sigma}_j^2}$, $r = 1, \dots, N$, em que N é o número de observações. Em seguida, calcula-se $\Phi(z_r)$, em que $\Phi(\cdot)$ é a função de distribuição de uma variável aleatória normal padrão. Os valores de z_r são arranjados em ordem crescente de magnitude. Calculam-se, então, as seguintes estatísticas teste para a função de distribuição empírica:

$$\text{Anderson-Darling: } Q_A = -\frac{1}{N} \sum_{r=1}^N (2r-1) \{ \ln(z_r) + \ln(1+z_{N+1-r}) \} - N,$$

$$\text{Cramer-von Mises: } Q_C = \sum_{r=1}^N \left[\frac{z_r - (2r-1)}{2N} \right]^2 + \frac{1}{12N},$$

$$\text{Watson: } Q_W = Q_C - N \left(\bar{z} - \frac{1}{2} \right)^2 \quad \text{onde } \bar{z} = \frac{1}{N} \sum_{r=1}^N z_r.$$

A idéia do teste é que, se as observações são normalmente distribuídas, então z_r deve ser

aproximadamente a estatística de ordem r de uma distribuição uniforme sobre o intervalo $(0, 1)$. Para avaliar desvios da normalidade logística aditiva, comparam-se as seguintes estatísticas da função de distribuição empírica modificada

$$\begin{aligned} \text{Anderson-Darling:} & \quad Q_A \left[1 + \frac{4}{N} + \frac{25}{N^2} \right], \\ \text{Cramer-von Mises:} & \quad Q_C \left[1 + \frac{1}{2N} \right], \\ \text{Watson:} & \quad Q_W \left[1 + \frac{1}{2N} \right], \end{aligned}$$

com valores críticos para um valor escolhido do nível de significância do teste (STEPHENS, 1974). Para o nível de significância de 5%, por exemplo, os valores críticos são 0,787 (Anderson-Darling), 0,126 (Cramer-von Mises) e 0,116 (Watson). A regra de decisão é a de que valores da estatística teste inferiores ao valor crítico evidenciam desvios não significativos da normalidade logística aditiva.

Neste trabalho, esses testes não são aplicados porque as observações são consideradas dependentes devido a estrutura espacial.

2.4.7 Perturbação e Potência

Os conceitos de *perturbação* e *potência* são válidos nos contextos espacial e não espacial e são definidos omitindo-se o índice espacial “ (\vec{x}) ”. Perturbação no simplex, para Eynatten, Barceló-Vidal e Pawlowsky-Glahn (2003), é uma operação que pode ser usada para descrever numericamente mudanças em uma composição. A combinação de perturbação e transformação potência fornecem um método para a análise de processos lineares composicionais no simplex. Para Aitchison (1986), Tolosana-Delgado, Otero e Pawlowsky-Glahn (2005) e Aitchison e Egozcue (2005), por exemplo, essas operações definem uma estrutura de espaço vetorial de dimensão $B - 1$ no simplex, com a perturbação como uma operação comutativa e a potência como um produto externo.

A perturbação, representada pelo símbolo \oplus , corresponde a multiplicar composições componente a componente e dividir cada um pela soma de todos, para se obter soma igual a 1. Por exemplo, a perturbação da composição \vec{X}_1 pela composição \vec{X}_2 é dada por

$$\vec{X}_1 \oplus \vec{X}_2 = (X_{11}, X_{12}, \dots, X_{1B}) \oplus (X_{21}, X_{22}, \dots, X_{2B}) = \mathcal{C}(X_{11}X_{21}, X_{12}X_{22}, \dots, X_{1B}X_{2B}).$$

É a operação análoga à translação no espaço real.

A potência, representada por \odot , é a análoga à multiplicação por um escalar no espaço real. Por exemplo, a transformação potência α da composição \vec{X}_1 é dada por

$$\alpha \odot (X_{11}, X_{12}, \dots, X_{1B}) = \mathcal{C}(X_{11}^\alpha, X_{12}^\alpha, \dots, X_{1B}^\alpha).$$

A partir destas definições, o vetor de diferenças composicionais é obtido fazendo-se

$$\vec{X}_1 \ominus \vec{X}_2 = \vec{X}_1 \oplus (-1 \odot \vec{X}_2).$$

Acrescentando às operações perturbação e potência um produto interno,

$$\langle \vec{X}_1, \vec{X}_2 \rangle = \sum_{i=1}^B \ln \left(\frac{X_{1i}}{g(\vec{X}_1)} \right) \ln \left(\frac{X_{2i}}{g(\vec{X}_2)} \right)$$

tem-se uma estrutura de espaço euclidiano real para o simplex, em que $g(\vec{X}_1) = \sqrt[B]{\prod_{j=1}^B X_{1j}}$ é a média geométrica dos componentes de \vec{X}_1 . Esse produto interno induz uma distância (uma medida que pode ser entendida, por exemplo, como grau de alteração) no simplex, denominada *distância de Aitchison*, dada por

$$d(\vec{X}_1, \vec{X}_2) = \sqrt{\sum_{i=1}^B \left(\ln \left(\frac{X_{1i}}{g(\vec{X}_1)} \right) - \ln \left(\frac{X_{2i}}{g(\vec{X}_2)} \right) \right)^2}.$$

Essa medida é usada para calcular a distância ou diferença entre duas composições e útil para entender a variabilidade dentro de um conjunto de dados.

Com isso, tem-se a geometria de Aitchison do simplex.

Para exemplificar, Tolosana-Delgado, Otero e Pawlowsky-Glahn (2005) consideram a composição de óxido de massa de uma amostra de sedimento $(\text{Al}_2\text{O}_3, \text{CaO} + \text{Na}_2\text{O}, \text{K}_2\text{O}) = (63, 8; 22, 7; 13, 5)\%$. Essa amostra é exposta à ação climatológica que provoca mudanças na sua estrutura e após um período de tempo, ficam restando apenas metade do conteúdo de Al_2O_3 , uma proporção de 0,17 de $\text{CaO} + \text{Na}_2\text{O}$ e 0,33 de K_2O . A composição da amostra passa a ser

$$(\text{Al}_2\text{O}_3, \text{CaO} + \text{Na}_2\text{O}, \text{K}_2\text{O}) = \mathcal{C}(63, 8 \times 0, 5; 22, 7 \times 0, 17; 13, 5 \times 0, 33) = (79, 3; 9, 6; 11, 1)\%,$$

e essa operação que modela a mudança na composição é a operação perturbação

$$(\text{Al}_2\text{O}_3, \text{CaO} + \text{Na}_2\text{O}, \text{K}_2\text{O}) = (63, 8; 22, 7; 13, 5) \oplus (0, 5; 0, 17; 0, 33).$$

Essa operação, aplicada várias vezes, resulta na transformação potência

$$(0, 5; 0, 17; 0, 33) \oplus (0, 5; 0, 17; 0, 33) \oplus (0, 5; 0, 17; 0, 33) = 3 \odot (0, 5; 0, 17; 0, 33),$$

que leva a

$$(\text{Al}_2\text{O}_3, \text{CaO} + \text{Na}_2\text{O}, \text{K}_2\text{O}) = (63, 8; 22, 7; 13, 5)(3 \odot (0, 5; 0, 17; 0, 33)) = (93, 04; 1, 30; 5, 66).$$

2.4.8 Estatísticas Descritivas e Domínio de Confiança

A média aritmética não é uma medida representativa para dados composicionais. Pawlowsky-Glahn e Olea (2004) sugerem como medida de tendência central ou centro da distribuição o fechamento da média geométrica $\text{cen}(\vec{X})$ que, para uma amostra de n composições com B componentes, é dada por:

$$\text{cen}(\vec{X}) = \frac{1}{g_s} [g(X_1) \quad g(X_2) \quad \dots \quad g(X_B)]', \quad (2.35)$$

em que $g(X_i)$ é a média geométrica do i -ésimo componente, como definido na Subseção 2.4.6, e

$$g_s = g(X_1) + g(X_2) + \dots + g(X_B).$$

Pelo mesmo motivo da Subseção 2.4.7, o índice espacial é omitido.

Para Pawlowsky-Glahn e Olea (2004) nem o desvio padrão nem a variância são medidas de dispersão adequadas quando o problema é considerado do ponto de vista multivariado. Mas consideram que uma medida de dispersão é importante para comparações entre conjuntos de dados e que uma medida da variabilidade total como medida de dispersão ao redor de uma medida do centro da distribuição faz sentido. Essa medida considera as $\frac{1}{2}(B-1)B$ variâncias das log-razões e é definida por:

$$\text{Totvar}(\vec{X}) = \frac{1}{B} \sum_{i < j} \text{Var} \left(\ln \left(\frac{X_i}{X_j} \right) \right). \quad (2.36)$$

Considerando que $\ln(X_i/X_i) = 0$, $i = 1, \dots, B$, e que $\text{Var}(\ln(X_i/X_j)) = \text{Var}(\ln(X_j/X_i))$, $i = 1, \dots, B-1$; $j = i+1, \dots, B$, pode-se dispor as variâncias das log-razões na forma de uma matriz de ordem $B \times B$, denominada *matriz variação*, permitindo uma forma simples de visualizar a variabilidade composicional.

Sob a suposição de que $\left(\ln \left(\frac{X_1}{X_B} \right), \ln \left(\frac{X_2}{X_B} \right), \dots, \ln \left(\frac{X_{B-1}}{X_B} \right) \right)$ tem distribuição assintoticamente gaussiana (GRAF, 2006; AITCHISON, 1986), o *domínio de confiança* para $\vec{X} = (X_1, X_2, \dots, X_B)$ é um subconjunto do simplex \mathbb{S}^B dado por:

$$B_{1-\alpha}(\vec{X}) = \left\{ \vec{X} \in \mathbb{S}^B \mid \left(\ln \left(\frac{\vec{X}_{-B}}{\vec{X}_B} \right) - \vec{\mu} \right)' \Sigma^{-1} \left(\ln \left(\frac{\vec{X}_{-B}}{\vec{X}_B} \right) - \vec{\mu} \right) \leq \chi_{B-1; 1-\alpha}^2 \right\},$$

em que $\chi_{B-1; 1-\alpha}^2$ é o quantil $(1-\alpha)$ da distribuição qui-quadrado com $(B-1)$ graus de liberdade, $\vec{\mu}$ é o vetor $(B-1) \times 1$ de médias aritméticas das log-razões e Σ é a matriz de covariâncias $(B-1) \times (B-1)$ das log-razões.

Como exemplo, o ponto vermelho na Figura 2.4 representa o centro da distribuição e a isolinha no interior do diagrama representa a região 2-desvios-padrão de confiança para os dados do lago Ártico (AITCHISON, 1986).

2.4.9 Estacionariedade

Seja $\vec{X}(\vec{x})$ um vetor função aleatória, $\vec{x} \in A \subset \mathbb{R}^n$, e f uma função tal que $f(\vec{X}(\vec{x}))$ também seja um vetor função aleatória. Pawlowsky-Glahn e Olea (2004) definem que $\vec{X}(\vec{x})$ é uma função *estacionária* de 2^a ordem se $f(\vec{X}(\vec{x}))$ é estacionária de 2^a ordem, isto é, se $f(\vec{X}(\vec{x}))$ satisfaz as condições:

- a) o vetor de valores esperados $E\left(f(\vec{X}(\vec{x}))\right) = \mu$ existe e não depende de \vec{x} ;
- b) a matriz função de covariância

$$\text{Cov}\left(f(\vec{X}(\vec{x}_1)), f(\vec{X}(\vec{x}_2))\right) = \Sigma(\vec{x}_2 - \vec{x}_1)$$

existe e não depende de (\vec{x}_1, \vec{x}_2) , mas somente da diferença $\vec{h} = \vec{x}_2 - \vec{x}_1$.

Além disso, $\vec{X}(\vec{x})$ é uma função *intrínseca* se $f(\vec{X}(\vec{x}))$ satisfaz:

- a) o vetor de valores esperados $E\left(f(\vec{X}(\vec{x}))\right) = \mu$ existe e não depende de \vec{x} ;
- b) a matriz função de covariância

$$\text{Cov}\left(f(\vec{X}(\vec{x}_2)) - f(\vec{X}(\vec{x}_1))\right) = \Gamma(\vec{x}_2 - \vec{x}_1)$$

existe e não depende de (\vec{x}_1, \vec{x}_2) , mas somente da diferença $\vec{h} = \vec{x}_2 - \vec{x}_1$.

Nota-se que $\Sigma(\vec{x}_2 - \vec{x}_1)$ é a matriz função de covariância cruzada e $\Gamma(\vec{x}_2 - \vec{x}_1)$ é a matriz de variogramas e variogramas cruzados.

Acrescenta-se, ainda, que $\vec{X}(\vec{x})$ é *log razão estacionária* de 2^a ordem (LR estacionária) se o conjunto das log razões entre todos os pares é estacionário de 2^a ordem; e é *log razão intrínseca* (LR intrínseca) se o conjunto das log razões entre todos os pares é intrínseco.

2.4.10 Estrutura de Covariância Espacial

O interesse principal deste trabalho está na metodologia de geoestatística aplicada a dados composicionais. Em geoestatística, considera-se a existência de dependência espacial entre os locais de observação. Por outro lado, a restrição da soma constante faz com que a correlação entre os componentes de uma composição sempre exista. Desse modo, a estrutura de covariância é essencial na modelagem de dados composicionais.

Pawlowsky-Glahn e Olea (2004) definem a estrutura de covariância espacial de uma composição regionalizada $\vec{X}(\vec{x})$ como o conjunto de funções

$$\sigma_{ij\cdot kl}(\vec{h}) = Cov \left(\ln \left(\frac{X_i(\vec{x})}{X_k(\vec{x})} \right), \ln \left(\frac{X_j(\vec{x} + \vec{h})}{X_l(\vec{x} + \vec{h})} \right) \right), \quad i, j, k, l \in \{1, 2, \dots, B\},$$

para $\vec{x}, \vec{x} + \vec{h} \in A$. Nesse caso, as seguintes propriedades são válidas

- a) em geral, $\sigma_{ij\cdot kl}(\vec{h}) \neq \sigma_{ji\cdot lk}(\vec{h})$;
- b) em geral, $\sigma_{ij\cdot kl}(\vec{h}) \neq \sigma_{ij\cdot kl}(-\vec{h})$;
- c) $\sigma_{ij\cdot kl}(\vec{h}) = \sigma_{ji\cdot lk}(-\vec{h})$;
- d) $\sigma_{ij\cdot kl}(\vec{h}) = -\sigma_{kj\cdot il}(\vec{h}) = \sigma_{kl\cdot ij}(\vec{h}) = -\sigma_{il\cdot kj}(\vec{h})$;
- e) $\sigma_{ij\cdot il}(\vec{h}) = \sigma_{ij\cdot kj}(\vec{h}) = \sigma_{ij\cdot ij}(\vec{h}) = \sigma_{ii\cdot ii}(\vec{h}) = 0$;
- f) $\sigma_{ij\cdot kl}(\vec{h}) = \sigma_{ij\cdot mn}(\vec{h}) + \sigma_{in\cdot ml}(\vec{h}) + \sigma_{mj\cdot kn}(\vec{h}) + \sigma_{mn\cdot kl}(\vec{h})$,

para $i, j, k, l, m, n \in \{1, 2, \dots, B\}$.

Do exposto, segue que, para quaisquer dois componentes $X_i(\vec{x})$ e $X_j(\vec{x})$, $i, j \in \{1, 2, \dots, B\}$ de uma composição $\vec{X}(\vec{x})$:

- a) a autocovariância das log razões, denominada *LR autocovariância*, é dada por

$$\tau_{i\cdot j}(\vec{h}) = \sigma_{ii\cdot jj}(\vec{h}),$$

que são os elementos da matriz de LR autocovariâncias (matriz variação), $\mathbf{T}(\vec{h})$ de dimensão $B \times B$.

- b) a *ALR covariância cruzada* é a função

$$\sigma_{ij}(\vec{h}) = \sigma_{ij\cdot BB}(\vec{h}),$$

que são os elementos da matriz de ALR covariâncias cruzadas $\Sigma(\vec{h})$, de dimensão $(B-1) \times (B-1)$.

- c) a *CLR covariância cruzada* é a função

$$\xi_{ij}(\vec{h}) = Cov \left(\ln \left(\frac{X_i(\vec{x})}{g(\vec{X}(\vec{x}))} \right), \ln \left(\frac{X_j(\vec{x} + \vec{h})}{g(\vec{X}(\vec{x} + \vec{h}))} \right) \right),$$

que são os elementos da matriz de CLR covariâncias cruzadas $\Xi(\vec{h})$, de dimensão $B \times B$.

2.4.11 Estrutura de Covariância Espacial Intrínica

A estrutura de covariância espacial intrínica de uma composição regionalizada $\vec{X}(\vec{x})$ é definida como o conjunto de funções

$$V_{ij \cdot kl}(\vec{h}) = \frac{1}{2} \text{Cov} \left(\ln \left(\frac{X_i(\vec{x})}{X_k(\vec{x})} \right) - \ln \left(\frac{X_i(\vec{x} + \vec{h})}{X_k(\vec{x} + \vec{h})} \right), \ln \left(\frac{X_j(\vec{x})}{X_l(\vec{x})} \right) - \ln \left(\frac{X_j(\vec{x} + \vec{h})}{X_l(\vec{x} + \vec{h})} \right) \right),$$

para $i, j, k, l \in \{1, 2, \dots, B\}$. Apresenta como propriedades

- a) $V_{ij \cdot kl}(0) = 0$;
- b) $V_{ij \cdot kl}(\vec{h}) = V_{ji \cdot lk}(\vec{h}) = V_{ij \cdot kl}(-\vec{h})$;
- c) $V_{ij \cdot il}(\vec{h}) = V_{ij \cdot kj}(\vec{h}) = V_{ij \cdot ij}(\vec{h}) = V_{ii \cdot ii}(\vec{h}) = 0$;
- d) $V_{ij \cdot kl}(\vec{h}) = -V_{kj \cdot il}(\vec{h}) = V_{kl \cdot ij}(\vec{h}) = -V_{il \cdot kj}(\vec{h})$;
- e) $|V_{ij \cdot kl}(\vec{h})| \leq \sqrt{V_{ii \cdot kk}(\vec{h})} \sqrt{V_{jj \cdot ll}(\vec{h})}$ (Desigualdade de Cauchy-Schwarz).

Em relação ao variograma, considerando quaisquer dois componentes $X_i(\vec{x})$, $X_j(\vec{x})$, $i, j \in \{1, 2, \dots, B\}$, de uma composição $\vec{X}(\vec{x})$, tem-se:

- a) o variograma da log razão, *LR variograma* dado por

$$V_{i \cdot j}(\vec{h}) = V_{ii \cdot jj}(\vec{h}),$$

que são os elementos da matriz de LR variogramas (matriz variação intrínica) $\Gamma(\vec{h})$, de dimensão $B \times B$.

- b) o *ALR variograma cruzado* definido como a função

$$\psi_{ij}(\vec{h}) = V_{ij \cdot BB}(\vec{h}),$$

que são os elementos da matriz de ALR covariâncias cruzadas intrínicas $\Psi(\vec{h})$, de dimensão $(B-1) \times (B-1)$.

- c) a *CLR covariância cruzada* definida como a função

$$\delta_{ij}(\vec{h}) = \text{Cov} \left(\ln \left(\frac{X_i(\vec{x})}{g(\vec{X}(\vec{x}))} \right) - \ln \left(\frac{X_i(\vec{x} + \vec{h})}{g(\vec{X}(\vec{x} + \vec{h}))} \right), \ln \left(\frac{X_j(\vec{x})}{g(\vec{X}(\vec{x}))} \right) - \ln \left(\frac{X_j(\vec{x} + \vec{h})}{g(\vec{X}(\vec{x} + \vec{h}))} \right) \right),$$

que são os elementos da matriz de CLR covariâncias cruzadas $\Delta(\vec{h})$, de dimensão $B \times B$.

2.5 Modelo Multivariado

Ao se realizar uma pesquisa, os dados coletados por amostragem em localizações espaciais frequentemente são multivariados, ou seja, mais de uma variável é mensurada em cada localização. As técnicas multivariadas encontradas em Johnson e Wichern (1998) e Reis (1997), por exemplo, são usadas segundo Bailey e Gatrell (1995) para fins de redução dos dados e exploração do espaço do atributo multidimensional, com o objetivo de identificar um número pequeno de sub-dimensões de interesse dado por combinações dos atributos, que podem então ser examinados sob uma perspectiva espacial, explorando padrões espaciais e relacionamentos, ou para uso em classificação e discriminação espacial. Wackernagel (1998) introduz geoestatística de uma perspectiva multivariada, Schabenberger e Pierce (2001) apresentam teoria de modelos estatísticos para dados espaciais uni e multivariados com enfoque para plantas e ciências do solo e Silva (2000), Silva et al. (2003) apresentam estudos sobre a variabilidade espacial de atributos do solo que são importantes para a agricultura de precisão.

Da perspectiva de Diggle e Ribeiro Jr. (2007), procura-se descrever a distribuição espacial conjunta das variáveis; a distribuição condicional de uma variável resposta de interesse dado uma ou mais covariáveis referenciadas espacialmente, ou ainda, como em Bognola et al. (2008), modelos multivariados são utilizados quando a variável de interesse é de difícil obtenção ou de custo elevado mas existe uma variável secundária correlacionada com esta, de fácil obtenção de modo a combinar poucas mensurações da variável de interesse com um número maior de mensurações da outra.

Em se tratando de mais de uma única variável resposta, considera-se o processo estocástico $\vec{Y}(\vec{x}) = (Y_1(\vec{x}), \dots, Y_p(\vec{x}))'$ de dimensão p , $\vec{x} \in \mathbb{R}^2$, como sendo o valor observado da variável resposta associada a localização \vec{x}_i da forma $\vec{y}(\vec{x}_i) = (y_1(\vec{x}_i), \dots, y_p(\vec{x}_i))'$. Desta maneira, de acordo com Diggle e Ribeiro Jr. (2007), para duas localizações \vec{x} e \vec{x}' , a função de covariância de $\vec{Y}(\vec{x})$ será uma função matriz $\Gamma(\vec{x}, \vec{x}')$, simétrica, dada por

$$\Gamma(\vec{x}, \vec{x}') = \begin{bmatrix} \text{Cov}(Y_1(\vec{x}), Y_1(\vec{x}')) & \text{Cov}(Y_1(\vec{x}), Y_2(\vec{x}')) & \cdots & \text{Cov}(Y_1(\vec{x}), Y_p(\vec{x}')) \\ \cdot & \text{Cov}(Y_2(\vec{x}), Y_2(\vec{x}')) & \cdots & \text{Cov}(Y_2(\vec{x}), Y_p(\vec{x}')) \\ \vdots & \vdots & \ddots & \vdots \\ \cdot & \cdot & \cdots & \text{Cov}(Y_p(\vec{x}), Y_p(\vec{x}')) \end{bmatrix},$$

denominada matriz de covariância cruzada.

Agora, se $\vec{Y}(\vec{x})$ for um processo estacionário, $\gamma_{jj}(\vec{x}, \vec{x}) = \text{Var}(Y_j(\vec{x})) = \sigma_j^2$ e para $j \neq k$, $\gamma_{kj}(\vec{x}, \vec{x}')$ depende somente de $u = \|\vec{x} - \vec{x}'\|$, de modo que:

$$\Gamma(\vec{x}, \vec{x}') = \begin{bmatrix} \sigma_1^2 & \text{Cov}(Y_1(\vec{x}), Y_2(\vec{x}')) & \cdots & \text{Cov}(Y_1(\vec{x}), Y_p(\vec{x}')) \\ \cdot & \sigma_2^2 & \cdots & \text{Cov}(Y_2(\vec{x}), Y_p(\vec{x}')) \\ \vdots & \vdots & \ddots & \vdots \\ \cdot & \cdot & \cdots & \sigma_p^2 \end{bmatrix}.$$

A função de correlação de $\vec{Y}(\vec{x})$ é definida então como a função $\mathbf{R}(u)$ uma matriz dada

por:

$$\mathbf{R}(u) = \begin{bmatrix} \rho_1(u) & \frac{\gamma_{12}(u)}{\sigma_1\sigma_2} & \dots & \frac{\gamma_{1p}(u)}{\sigma_1\sigma_p} \\ \cdot & \rho_2(u) & \dots & \frac{\gamma_{2p}(u)}{\sigma_2\sigma_p} \\ \vdots & \vdots & \ddots & \vdots \\ \cdot & \cdot & \dots & \rho_p(u) \end{bmatrix},$$

em que para $j = k$, as funções $\rho_{jj}(u)$ são funções de correlação do processo univariado de $Y_j(\vec{x})$ e satisfazem $\rho_{jk}(u) = \rho_{kj}(-u)$. Para $j \neq k$ as funções $\rho_{jk}(u)$, funções de correlação cruzadas de $\vec{Y}(\vec{x})$, não são necessariamente simétricas mas satisfazem $\rho_{jk}(u) = \rho_{kj}(-u)$.

O grande problema encontrado na modelagem de dados multivariados é garantir que a matriz de covariância $\Gamma(\vec{x}, \vec{x}')$ seja definida positiva, $\forall \vec{x}, \vec{x}' \in \mathbb{R}^2$. Isto pode ser garantido mediante a construção de combinações lineares de componentes independentes cuja descrição podem ser encontradas em Diggle e Ribeiro Jr. (2007), Schmidt e Gelfand (2003), Banerjee, Carlin e Gelfand (2004), Schmidt e Sansó (2006). O modelo de correionalização linear é apresentado em Goovaerts (1997, Seção 4.2.3), Wackernagel (1998), Schmidt e Sansó (2006). Procedimentos baseados em separabilidade, médias móveis, convolução e correionalização, são descritos em Banerjee, Carlin e Gelfand (2004).

2.5.1 Definição do Modelo Multivariado

Considere que as variáveis Y_1, Y_2, \dots, Y_p , sigam alguma distribuição, $\vec{S}(\vec{x}) = (S_1(\vec{x}), S_2(\vec{x}), \dots, S_p(\vec{x}))$ seja um processo gaussiano estacionário não observado (latente) e que $\vec{Y}(\vec{x})$ seja condicionalmente independente dado $\vec{S}(\vec{x})$. Muitas vezes a distribuição conjunta de probabilidade de $\vec{Y}(\vec{x}) = (Y_1(\vec{x}), Y_2(\vec{x}), \dots, Y_p(\vec{x}))'$ não é conhecida, mas ao escrever

$$[\vec{Y}(\vec{x}), \vec{S}(\vec{x})] = [\vec{S}(\vec{x})][\vec{Y}(\vec{x})|\vec{S}(\vec{x})] \quad (2.37)$$

em que $[\cdot]$ representa a distribuição de probabilidade e, considerando como exemplo dados bivariados, $\vec{Y}(\vec{x}) = (Y_1(\vec{x}), Y_2(\vec{x}))'$, observa-se que para um conjunto finito de pontos, $\vec{S}(\vec{x})$ é gaussiana multivariada e $\vec{Y}(\vec{x})|\vec{S}(\vec{x})$ é um produto de densidades univariadas, já que $Y_1(\vec{x})$ e $Y_2(\vec{x})$ que marginalmente são dependentes, dado o conhecimento de $S_1(\vec{x})$ e $S_2(\vec{x})$, tornam-se independentes. Logo, com um processo gaussiano multivariado latente $\vec{S}(\vec{x})$ e uma independência condicional ($\vec{Y}(\vec{x})|\vec{S}(\vec{x})$) é possível construir um modelo multivariado para $\vec{Y}(\vec{x})$ sem a necessidade de que este tenha distribuição gaussiana, bastando para isso integrar (2.37) com relação a $\vec{S}(\vec{x})$ obtendo-se a distribuição marginal de $\vec{Y}(\vec{x})$

$$[\vec{Y}(\vec{x})] = \int [\vec{Y}(\vec{x}), \vec{S}(\vec{x})] d\vec{S}(\vec{x}) = \int [\vec{S}(\vec{x})][\vec{Y}(\vec{x})|\vec{S}(\vec{x})] d\vec{S}(\vec{x}).$$

Portanto este produto, integrado em relação a $\vec{S}(\vec{x})$, resulta numa distribuição de probabilidade que é o objetivo da modelagem. No caso em que $[\vec{Y}(\vec{x})|\vec{S}(\vec{x})]$ é suposta gaussiana,

$[\vec{Y}(\vec{x})]$ será gaussiana multivariada.

Seguindo a proposta de Diggle e Ribeiro Jr. (2007), Bognola et al. (2008), no caso bivariado o modelo é dado por:

$$\begin{cases} Y_1(\vec{x}_i) = \mu_1(\vec{x}_i) + S_1(\vec{x}_i) + Z_1(\vec{x}_i) \\ Y_2(\vec{x}_{i'}) = \mu_2(\vec{x}_{i'}) + S_2(\vec{x}_{i'}) + Z_2(\vec{x}_{i'}) \end{cases} \quad (2.38)$$

em que as respostas

$$\vec{Y}(\vec{x}) = (Y_1(\vec{x}), Y_2(\vec{x}))' = (Y_1(\vec{x}_1), Y_1(\vec{x}_2), \dots, Y_1(\vec{x}_{n_1}), Y_2(\vec{x}_1), Y_2(\vec{x}_2), \dots, Y_2(\vec{x}_{n_2}))'$$

são medidas nas localizações $\vec{x}_i, \vec{x}_{i'}$, $i, i' = 1, \dots, n_j, j = 1, 2$; $\vec{S}(\vec{x}) = (S_1(\vec{x}), S_2(\vec{x}))$ é considerado um processo gaussiano estacionário com média $\vec{\mu} = (0, 0)'$, variâncias $\sigma_j^2 = \text{Var}(S_j(\vec{x}))$ e estrutura de correlação dada por

$$\begin{bmatrix} \text{Corr}(S_1(\vec{x}), S_1(\vec{x} - u)) & \text{Corr}(S_1(\vec{x}), S_2(\vec{x} - u)) \\ \text{Corr}(S_1(\vec{x}), S_2(\vec{x} - u)) & \text{Corr}(S_2(\vec{x}), S_2(\vec{x} - u)) \end{bmatrix};$$

e $Z_j \sim N(0; \tau^2)$.

É suposto para $Y_1(\vec{x}_i)$ que $\mu_1(\vec{x}_i)$ é o efeito fixo e o efeito aleatório $S_1(\vec{x}_i)$ é decomposto em dois outros efeitos aleatórios: $S_{01}^*(\vec{x}_i)$ comum a $Y_1(\vec{x}_i)$ e $Y_2(\vec{x}_{i'})$, e $S_1^*(\vec{x}_i)$ específico a $Y_1(\vec{x}_i)$. O mesmo ocorre para $Y_2(\vec{x}_{i'})$. Mas, como em geral $Y_j(\vec{x}_i)$ são medidas em escalas diferentes, é necessário fazer uma padronização que é dada por $\sigma_{0j}U_0(\vec{x}_i)$ e $\sigma_j U_j(\vec{x}_i)$ sendo $U_0(\vec{x}_i)$ e $U_j(\vec{x}_i)$ efeitos aleatórios padronizados que têm correlação espacial, logo adimensionais, com as unidades preservadas nas constantes padronizadoras σ_{0j} e σ_j . Estes efeitos apresentam distribuição gaussiana com vetor de médias iguais a zero e matriz de covariância com variâncias unitárias na diagonal principal e covariâncias cruzadas dadas pela função de correlação adotada. Dessa forma, o modelo bivariado pode ser reescrito como

$$\begin{cases} Y_1(\vec{x}_i) = \mu_1(\vec{x}_i) + \sigma_{01}U_0(\vec{x}_i) + \sigma_1 U_1(\vec{x}_i) + Z_1(\vec{x}_i) \\ Y_2(\vec{x}_{i'}) = \mu_2(\vec{x}_{i'}) + \sigma_{02}U_0(\vec{x}_{i'}) + \sigma_2 U_2(\vec{x}_{i'}) + Z_2(\vec{x}_{i'}) \end{cases} \quad (2.39)$$

É possível calcular a correlação entre Y_1 em uma localização \vec{x}_i e Y_1 em uma outra localização $\vec{x}_{i'}$, $i, i' : 1, \dots, n_1$ através do modelo espacial dado pela Equação (2.39), primeira linha, em que $U_0(\vec{x}_i)$ e $U_1(\vec{x}_i)$ contribuem para este cálculo. Da mesma forma é possível calcular a correlação considerando-se Y_2 , segunda linha do modelo (2.39). Também, a correlação de Y_1 em uma localização \vec{x}_i e Y_2 em uma localização $\vec{x}_{i'}$ decorre do modelo (2.39) e, neste caso, apenas $U_0(\vec{x}_i)$ e $U_0(\vec{x}_{i'})$ contribuem para o cálculo já que os outros termos são independentes.

Considerando-se apenas duas localizações (\vec{x}_1 e \vec{x}_2) a matriz Σ será dada por:

$$\Sigma = \begin{bmatrix} \sigma_{01}^2 \rho_0(\vec{x}_1, \vec{x}_1) + \sigma_1^2 \rho_1(\vec{x}_1, \vec{x}_1) & \sigma_{01}^2 \rho_0(\vec{x}_1, \vec{x}_2) + \sigma_1^2 \rho_1(\vec{x}_1, \vec{x}_2) & \sigma_{01} \sigma_{02} \rho_0(\vec{x}_1, \vec{x}_1) & \sigma_{01} \sigma_{02} \rho_0(\vec{x}_1, \vec{x}_2) \\ \sigma_{01}^2 \rho_0(\vec{x}_2, \vec{x}_1) + \sigma_1^2 \rho_1(\vec{x}_2, \vec{x}_1) & \sigma_{01}^2 \rho_0(\vec{x}_2, \vec{x}_2) + \sigma_1^2 \rho_1(\vec{x}_2, \vec{x}_2) & \sigma_{01} \sigma_{02} \rho_0(\vec{x}_2, \vec{x}_1) & \sigma_{01} \sigma_{02} \rho_0(\vec{x}_2, \vec{x}_2) \\ \sigma_{01} \sigma_{02} \rho_0(\vec{x}_1, \vec{x}_1) & \sigma_{01} \sigma_{02} \rho_0(\vec{x}_1, \vec{x}_2) & \sigma_{02}^2 \rho_0(\vec{x}_1, \vec{x}_1) + \sigma_2^2 \rho_2(\vec{x}_1, \vec{x}_1) & \sigma_{02}^2 \rho_0(\vec{x}_1, \vec{x}_2) + \sigma_2^2 \rho_2(\vec{x}_1, \vec{x}_2) \\ \sigma_{01} \sigma_{02} \rho_0(\vec{x}_2, \vec{x}_1) & \sigma_{01} \sigma_{02} \rho_0(\vec{x}_2, \vec{x}_2) & \sigma_{02}^2 \rho_0(\vec{x}_2, \vec{x}_1) + \sigma_2^2 \rho_2(\vec{x}_2, \vec{x}_1) & \sigma_{02}^2 \rho_0(\vec{x}_2, \vec{x}_2) + \sigma_2^2 \rho_2(\vec{x}_2, \vec{x}_2) \end{bmatrix}.$$

A estimação assim como a avaliação da incerteza sobre os parâmetros do modelo segue numa extensão da apresentada na Subseção 2.1.3.

2.5.2 Cokrigagem

A predição espacial no contexto multivariado é uma extensão da teoria apresentada na Subseção 2.2.2. Considera-se a predição espacial de $\vec{Y}_0(\vec{x})$ em localizações não amostradas $\vec{x}_0 = (\vec{x}_{10}, \vec{x}_{20}, \dots, \vec{x}_{n_20})$. O vetor de médias correspondentes às variáveis Y_1 e Y_2 para todas as localizações de predição e a matriz de covariância são dadas pelo seguinte resultado da distribuição gaussiana multivariada:

Teorema 2.2. *Seja $\vec{Y} = (\vec{Y}_0, \vec{Y})'$ um vetor bivariado com distribuição gaussiana multivariada conjunta com vetor de médias $\vec{\mu} = (\mu_{\vec{Y}_0}, \mu_{\vec{Y}})'$ e matriz de covariância*

$$\Sigma = \begin{bmatrix} \Sigma_{\vec{Y}_0 \vec{Y}_0} & \Sigma_{\vec{Y}_0 \vec{Y}} \\ \Sigma_{\vec{Y} \vec{Y}_0} & \Sigma_{\vec{Y} \vec{Y}} \end{bmatrix}$$

isto é, $\vec{Y} \sim N(\vec{\mu}; \Sigma)$. Então, a distribuição condicional de \vec{Y}_0 dado \vec{Y} é também gaussiana multivariada,

$$\vec{Y}_0 | \vec{Y} \sim N(\vec{\mu}_{\vec{Y}_0 | \vec{Y}}; \Sigma_{\vec{Y}_0 | \vec{Y}})$$

onde

$$\vec{\mu}_{\vec{Y}_0 | \vec{Y}} = \vec{\mu}_{\vec{Y}_0} + \Sigma_{\vec{Y}_0 \vec{Y}} \Sigma_{\vec{Y} \vec{Y}}^{-1} (\vec{Y} - \vec{\mu}_{\vec{Y}}) \quad e \quad \Sigma_{\vec{Y}_0 | \vec{Y}} = \Sigma_{\vec{Y}_0 \vec{Y}_0} - \Sigma_{\vec{Y}_0 \vec{Y}} \Sigma_{\vec{Y} \vec{Y}}^{-1} \Sigma_{\vec{Y} \vec{Y}_0}. \quad (2.40)$$

Sendo desconhecidos os valores de $\vec{\mu}_{\vec{Y}_0}$, estes são substituídos pelo vetor de médias estimadas obtidas no processo de otimização. A matriz Σ , de onde se extrai as matrizes $\Sigma_{\vec{Y}_0 \vec{Y}_0}$, $\Sigma_{\vec{Y}_0 \vec{Y}}$, $\Sigma_{\vec{Y} \vec{Y}_0}$ e $\Sigma_{\vec{Y} \vec{Y}}$, é calculada substituindo-se os valores estimados, restantes no processo de otimização, na matriz \mathbf{V} multiplicada por $\hat{\sigma}_1^2$.

As Equações (2.23) e (2.24) para o caso univariado, no sentido de um único atributo espacial, também podem ser obtidas desse resultado.

3 METODOLOGIA PROPOSTA

Como visto no capítulo anterior, a literatura contempla trabalhos sobre a teoria de dados composicionais, geoestatística e geoestatística aplicada a dados composicionais. Aitchison (1986) apresenta a teoria de dados composicionais considerando a independência das observações; Pawlowsky-Glahn e Olea (2004) fazem análise geoestatística de dados composicionais seguindo a teoria clássica da geoestatística e Lark e Bishop (2007) apresentam um estudo sobre cokrigagem de frações de partículas do solo concluindo que a predição de dados composicionais pode ser feita por cokrigagem ALR com transformação de volta das predições, para a escala das composições, calculadas por quadratura de Gauss-Hermite na aproximação da esperança condicional. Por outro lado, Diggle e Ribeiro Jr. (2007), Schmidt e Sansó (2006), Banerjee, Carlin e Gelfand (2004), Finley, Banerjee e Carlin (2007), Wackernagel (1998) e outros explicitam modelos geoestatísticos uni e multivariados através da especificação de uma função de correlação em que a covariância é função da distância entre pares de localizações. Sob o paradigma bayesiano, tem-se o trabalho de Obage (2007) que faz inferência bayesiana de dados composicionais sem considerar o efeito espacial; já Tjelmeland e Lund (2003) o consideram, mas não especificam uma forma fechada para a função de covariância. Para esta, os autores adotam uma função de correlação exponencial generalizada e uma distribuição *a priori* do tipo Wishart para a estimação dos parâmetros de variância. O que se apresenta a seguir é o desenvolvimento de uma metodologia para análise geoestatística de dados composicionais que compreende a definição do modelo para composições de três componentes, estimação dos parâmetros via verossimilhança e predição espacial tanto clássica como sob o paradigma bayesiano.

O trabalho foi realizado utilizando recursos de *software* livre em ambiente operacional GNU/Linux, no ambiente estatístico R (R development Core Team, 2008), utilizando o pacote *geoR* (RIBEIRO JR.; DIGGLE, 2001), *compositions* (BOOGAART; TOLOSANA; BREN, 2009), *statmod* (SMYTH; HU; DUNN, 2009), *MCMCpack* (MARTIN; QUINN; PARK, 2009), *coda* (PLUMMER et al., 2006) e rotinas desenvolvidas especificamente para o desenvolvimento deste trabalho. Essas rotinas foram organizadas e dispostas em forma de pacote *geoComp* disponível em <http://www.leg.ufpr.br>. As funções, respectivos manuais e os *scripts* adicionais ao pacote referentes às análises realizadas encontram-se nos Anexos.

3.1 Modelo Geoestatístico Composicional

Para $\vec{X} = (X_1, \dots, X_B)'$ sendo uma composição com B componentes e $\vec{Y} = \left(\ln\left(\frac{X_1}{X_B}\right), \dots, \ln\left(\frac{X_{B-1}}{X_B}\right) \right)'$ um vetor com $B - 1$ elementos, o modelo geoestatístico com componente comum pode ser obtido seguindo a formulação dada em Diggle e Ribeiro Jr. (2007). Neste trabalho, considerou-se composições de três componentes, (X_1, X_2, X_3) , que como exemplo de aplicação à fração granulométrica do solo, poderiam corresponder a (Areia, Silte, Argila).

A partir do modelo geoestatístico bivariado com componente espacial comum (2.38), reescrito como (2.39), propõe-se o seguinte modelo geoestatístico bivariado composicional:

$$\begin{cases} Y_1(\vec{x}_i) &= \mu_1(\vec{x}_i) + S_1(\vec{x}_i) + Z_1(\vec{x}_i), \\ Y_2(\vec{x}_{i'}) &= \mu_2(\vec{x}_{i'}) + S_2(\vec{x}_{i'}) + Z_2(\vec{x}_{i'}), \end{cases}$$

em que $\vec{x}_i, \vec{x}_{i'} \in \mathbb{R}^2$, são as localizações amostrais $i, i' = 1, \dots, n_1$, sendo n_1 o tamanho da amostra; $Y_1 = \ln(X_1/X_3)$, $Y_2 = \ln(X_2/X_3)$ são as variáveis resposta do modelo de modo que $\vec{Y}_{n \times 1} = (Y_1(\vec{x}_1), Y_2(\vec{x}_1), \dots, Y_1(\vec{x}_{n_1}), Y_2(\vec{x}_{n_1}))'$, ou seja, as observações são “empilhadas” por variável; $S_j(\vec{x}) \sim N(0; \sigma_j^2)$ e $Z_j(\vec{x}) \sim N(0; \tau_j^2)$, $j = 1, 2$.

Nesse modelo, os efeitos aleatórios com estrutura espacial S_1 e S_2 foram substituídos por um efeito aleatório padronizado U . Supôs-se que este efeito tem distribuição gaussiana multivariada com vetor de médias iguais a zero e matriz de covariâncias, com variâncias unitárias e covariâncias dadas pela função de correlação exponencial, ρ_U . Esta função é caracterizada pelo parâmetro de alcance, ϕ , que controla o decaimento da correlação como função da separação espacial entre duas localizações. No modelo bivariado geral, as unidades de medida são preservadas nas constantes padronizadoras σ_1 e σ_2 , enquanto que, no contexto considerado aqui, são adimensionais. Os efeitos aleatórios Z_1 e Z_2 capturam a variabilidade não espacial incluindo a correlação, ρ , induzida pela estrutura composicional. O modelo porposto pode então ser reescrito como:

$$\begin{cases} Y_1(\vec{x}_i) &= \mu_1(\vec{x}_i) + \sigma_1 U(\vec{x}_i; \phi) + Z_1(\vec{x}_i), \\ Y_2(\vec{x}_{i'}) &= \mu_2(\vec{x}_{i'}) + \sigma_2 U(\vec{x}_{i'}; \phi) + Z_2(\vec{x}_{i'}). \end{cases} \quad (3.1)$$

Sendo assim, $\vec{Y} \sim N(\vec{\mu}; \Sigma)$, com a matriz de covariâncias Σ composta pelos elementos

$$\begin{aligned} \text{Cov}(Y_1(\vec{x}_i); Y_1(\vec{x}_i)) &= \sigma_1^2 + \tau_1^2, & \text{Cov}(Y_1(\vec{x}_i); Y_1(\vec{x}_{i'})) &= \sigma_1^2 \rho_U(\vec{x}_i; \vec{x}_{i'}), \\ \text{Cov}(Y_2(\vec{x}_i); Y_2(\vec{x}_i)) &= \sigma_2^2 + \tau_2^2, & \text{Cov}(Y_2(\vec{x}_i); Y_2(\vec{x}_{i'})) &= \sigma_2^2 \rho_U(\vec{x}_i; \vec{x}_{i'}), \end{aligned}$$

e

$$\text{Cov}(Y_1(\vec{x}_i); Y_2(\vec{x}_{i'})) = \sigma_1 \sigma_2 I_2(i, i') + \tau_1 \tau_2 I_3(i, i'),$$

com as funções indicadoras I_2 e I_3 definidas como:

$$I_2(i, i') = \begin{cases} 1 & , \text{ se } i = i', \\ \rho_U(\vec{x}_i; \vec{x}_{i'}) & , \text{ se } i \neq i', \end{cases} \quad I_3(i, i') = \begin{cases} \rho & , \text{ se } i = i', \\ 0 & , \text{ se } i \neq i', \end{cases}$$

ou seja, a matriz de covariância é da forma,

$$\Sigma = \begin{bmatrix} \sigma_1^2 + \tau_1^2 & \sigma_1\sigma_2 + \tau_1\tau_2\rho & \sigma_1^2\rho_U(\vec{x}_1; \vec{x}_2) & \sigma_1\sigma_2\rho_U(\vec{x}_1; \vec{x}_2) \cdots \sigma_1^2\rho_U(\vec{x}_1; \vec{x}_{n_1}) & \sigma_1\sigma_2\rho_U(\vec{x}_1; \vec{x}_{n_1}) \\ \sigma_1\sigma_2 + \tau_1\tau_2\rho & \sigma_2^2 + \tau_2^2 & \sigma_1\sigma_2\rho_U(\vec{x}_1; \vec{x}_2) & \sigma_2^2\rho_U(\vec{x}_1; \vec{x}_2) \cdots \sigma_1\sigma_2\rho_U(\vec{x}_1; \vec{x}_{n_1}) & \sigma_2^2\rho_U(\vec{x}_1; \vec{x}_{n_1}) \\ \cdot & \cdot & \sigma_1^2 + \tau_1^2 & \sigma_1\sigma_2 + \tau_1\tau_2\rho \cdots \sigma_1^2\rho_U(\vec{x}_2; \vec{x}_{n_1}) & \sigma_1\sigma_2\rho_U(\vec{x}_2; \vec{x}_{n_1}) \\ \cdot & \cdot & \sigma_1\sigma_2 + \tau_1\tau_2\rho & \sigma_2^2 + \tau_2^2 \cdots \sigma_1\sigma_2\rho_U(\vec{x}_2; \vec{x}_{n_1}) & \sigma_2^2\rho_U(\vec{x}_2; \vec{x}_{n_1}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdot & \cdot & \cdot & \cdot & \cdots & \sigma_1^2 + \tau_1^2 & \sigma_1\sigma_2 + \tau_1\tau_2\rho \\ \cdot & \cdot & \cdot & \cdot & \cdots & \sigma_1\sigma_2 + \tau_1\tau_2\rho & \sigma_2^2 + \tau_2^2 \end{bmatrix}.$$

3.1.1 Estimação dos Parâmetros

O vetor de parâmetros do modelo é $\vec{\theta} = (\mu_1, \mu_2, \sigma_1, \sigma_2, \tau_1, \tau_2, \phi, \rho)'$ e sua função de verossimilhança é obtida a partir da função de densidade da distribuição normal multivariada:

$$L(\vec{\theta}; \vec{Y}) = (2\pi)^{-n/2} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2} (\vec{Y} - \vec{\mu}_{\vec{Y}})' \Sigma^{-1} (\vec{Y} - \vec{\mu}_{\vec{Y}}) \right\},$$

em que o termo $\vec{\mu}_{\vec{Y}}$ é função dos parâmetros μ_1 e μ_2 e os demais parâmetros $(\sigma_1, \sigma_2, \tau_1, \tau_2, \phi, \rho)$ definem os termos em Σ .

Fazendo a reparametrização: $\eta = \sigma_2/\sigma_1$; $v_1 = \tau_1/\sigma_1$; $v_2 = \tau_2/\sigma_1$, pode-se escrever

$$\Sigma = \sigma_1^2 \mathbf{R} + \tau_1^2 \mathbf{I}_b = \sigma_1^2 \mathbf{V},$$

em que

$$\mathbf{R} = \begin{bmatrix} 1 & \eta & \rho_U(\vec{x}_1, \vec{x}_2) & \eta\rho_U(\vec{x}_1, \vec{x}_2) & \cdots & \rho_U(\vec{x}_1, \vec{x}_{n_1}) & \eta\rho_U(\vec{x}_1, \vec{x}_{n_1}) \\ \eta & \eta^2 & \eta\rho_U(\vec{x}_1, \vec{x}_2) & \eta^2\rho_U(\vec{x}_1, \vec{x}_2) & \cdots & \eta\rho_U(\vec{x}_1, \vec{x}_{n_1}) & \eta^2\rho_U(\vec{x}_1, \vec{x}_{n_1}) \\ \rho_U(\vec{x}_2, \vec{x}_1) & \eta\rho_U(\vec{x}_2, \vec{x}_1) & 1 & \eta & \cdots & \rho_U(\vec{x}_2, \vec{x}_{n_1}) & \eta\rho_U(\vec{x}_2, \vec{x}_{n_1}) \\ \eta\rho_U(\vec{x}_2, \vec{x}_1) & \eta^2\rho_U(\vec{x}_2, \vec{x}_1) & \eta & \eta^2 & \cdots & \eta\rho_U(\vec{x}_2, \vec{x}_{n_1}) & \eta^2\rho_U(\vec{x}_2, \vec{x}_{n_1}) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_U(\vec{x}_{n_1}, \vec{x}_1) & \eta\rho_U(\vec{x}_{n_1}, \vec{x}_1) & \rho_U(\vec{x}_{n_1}, \vec{x}_2) & \eta\rho_U(\vec{x}_{n_1}, \vec{x}_2) & \cdots & 1 & \eta \\ \eta\rho_U(\vec{x}_{n_1}, \vec{x}_1) & \eta^2\rho_U(\vec{x}_{n_1}, \vec{x}_1) & \eta\rho_U(\vec{x}_{n_1}, \vec{x}_2) & \eta^2\rho_U(\vec{x}_{n_1}, \vec{x}_2) & \cdots & \eta & \eta^2 \end{bmatrix},$$

$$\mathbf{I}_b = \begin{bmatrix} v_1^2 & v_1 v_2 \rho \\ v_1 v_2 \rho & v_2^2 \end{bmatrix} \otimes \mathbf{I}_n = \begin{bmatrix} v_1^2 & v_1 v_2 \rho & 0 & 0 & \cdots & 0 & 0 \\ v_1 v_2 \rho & v_2^2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & v_1^2 & v_1 v_2 \rho & \cdots & 0 & 0 \\ 0 & 0 & v_1 v_2 \rho & v_2^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & v_1^2 & v_1 v_2 \rho \\ 0 & 0 & 0 & 0 & \cdots & v_1 v_2 \rho & v_2^2 \end{bmatrix}$$

e, conseqüentemente,

$$\mathbf{V} = \begin{bmatrix} 1 + v_1^2 & \eta + v_1 v_2 \rho & \rho_U(\bar{x}_1, \bar{x}_2) & \eta \rho_U(\bar{x}_1, \bar{x}_2) & \cdots & \rho_U(\bar{x}_1, \bar{x}_{n_1}) & \eta \rho_U(\bar{x}_1, \bar{x}_{n_1}) \\ \eta + v_1 v_2 \rho & \eta^2 + v_2^2 & \eta \rho_U(\bar{x}_1, \bar{x}_2) & \eta^2 \rho_U(\bar{x}_1, \bar{x}_2) & \cdots & \eta \rho_U(\bar{x}_1, \bar{x}_{n_1}) & \eta^2 \rho_U(\bar{x}_1, \bar{x}_{n_1}) \\ \rho_U(\bar{x}_2, \bar{x}_1) & \eta \rho_U(\bar{x}_2, \bar{x}_1) & 1 + v_1^2 & \eta + v_1 v_2 \rho & \cdots & \rho_U(\bar{x}_2, \bar{x}_{n_1}) & \eta \rho_U(\bar{x}_2, \bar{x}_{n_1}) \\ \eta \rho_U(\bar{x}_2, \bar{x}_1) & \eta^2 \rho_U(\bar{x}_2, \bar{x}_1) & \eta + v_1 v_2 \rho & \eta^2 + v_2^2 & \cdots & \rho_U(\bar{x}_2, \bar{x}_{n_1}) & \eta^2 \rho_U(\bar{x}_2, \bar{x}_{n_1}) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_U(\bar{x}_{n_1}, \bar{x}_1) & \eta \rho_U(\bar{x}_{n_1}, \bar{x}_1) & \rho_U(\bar{x}_{n_1}, \bar{x}_2) & \eta \rho_U(\bar{x}_{n_1}, \bar{x}_2) & \cdots & 1 + v_1^2 & \eta + v_1 v_2 \rho \\ \eta \rho_U(\bar{x}_{n_1}, \bar{x}_1) & \eta^2 \rho_U(\bar{x}_{n_1}, \bar{x}_1) & \eta \rho_U(\bar{x}_{n_1}, \bar{x}_2) & \eta^2 \rho_U(\bar{x}_{n_1}, \bar{x}_2) & \cdots & \eta + v_1 v_2 \rho & \eta^2 + v_2^2 \end{bmatrix}.$$

A função de log-verossimilhança reparametrizada é dada por

$$l(\vec{\theta}; \vec{Y}) = -\frac{1}{2} \left(n \ln(2\pi) + 2n \ln(\sigma_1) + \ln(|\mathbf{V}|) + \frac{1}{\sigma_1^2} Qe \right). \quad (3.2)$$

Considerando $\vec{\mu}_{\vec{Y}} = \mathbf{D}\vec{\mu}$, em que \mathbf{D} é a matriz do delineamento de ordem $n \times 2$, ou seja,

$$\mathbf{D}' = \begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 1 & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 & 1 \end{bmatrix},$$

tem-se que $Qe = (\vec{Y} - \vec{\mu}_{\vec{Y}})' \mathbf{V}^{-1} (\vec{Y} - \vec{\mu}_{\vec{Y}})$ pode ser reescrita como

$$Qe = \vec{Y}' \mathbf{V}^{-1} \vec{Y} - 2(\vec{Y}' \mathbf{V}^{-1} \mathbf{D}) \vec{\mu} + \vec{\mu}' (\mathbf{D}' \mathbf{V}^{-1} \mathbf{D}) \vec{\mu}.$$

Expressões analíticas fechadas podem ser obtidas para os estimadores de máxima verossimilhança de $\vec{\mu} = (\mu_1, \mu_2)'$ e σ_1 diferenciando a função (3.2) em relação aos respectivos parâmetros, e estes são dados por

$$\hat{\vec{\mu}} = (\mathbf{D}' \mathbf{V}^{-1} \mathbf{D})^{-1} (\mathbf{D}' \mathbf{V}^{-1} \vec{Y}) \quad \text{e} \quad \hat{\sigma}_1 = \sqrt{\hat{Q}e/n}. \quad (3.3)$$

É importante notar que $\hat{Q}e$ pode ser escrita como

$$\hat{Q}e = \vec{Y}' \mathbf{V}^{-1} \vec{Y} - (\vec{Y}' \mathbf{V}^{-1} \mathbf{D}) (\mathbf{D}' \mathbf{V}^{-1} \mathbf{D})^{-1} (\mathbf{D}' \mathbf{V}^{-1} \vec{Y}).$$

Ao substituir as expressões (3.3) em (3.2) obtém-se a função de log-verossimilhança concentrada

$$l(\vec{\theta}^*; \vec{Y}) = -\frac{1}{2} [\ln(|\mathbf{V}|) + n (\ln(2\pi) + \ln(\hat{Q}e) - \ln(n) + 1)],$$

que é uma função do vetor de parâmetros desconhecidos $\vec{\theta}^* = (\eta, v_1, v_2, \phi, \rho)'$, e pode ser maximizada numericamente.

Os algoritmos de otimização empregados no processo de maximização foram “L-BFGS-B”, método de Byrd et al. (1995), que permite informar os limites inferior e superior de busca no espaço paramétrico; “Nelder-Mead”, uma implementação do método de Nelder e Mead (1965); “Gradiente Conjugado”, baseado no método de Fletcher e Reeves (1964) e “BFGS”, um método quasi-Newton conforme implementação na função *optim()* no ambiente estatístico R (R Development Core Team, 2008).

3.1.2 Incerteza Sobre os Parâmetros

Do processo de maximização obtém-se $\hat{\theta}^* = (\hat{\eta}, \hat{v}_1, \hat{v}_2, \hat{\phi}, \hat{\rho})'$ e as respectivas variâncias através da matriz Hessiana numérica dada pela derivada segunda do logaritmo da função de verossimilhança em relação aos parâmetros em $\vec{\theta}^*$. A matriz Informação de Fisher observada é definida como o negativo da matriz Hessiana e é dada por

$$\mathbf{I}_F(\hat{\theta}^*) = -\frac{\partial^2 l(\hat{\theta}^*)}{\partial \hat{\theta}^* \partial (\hat{\theta}^*)'} = - \begin{bmatrix} \frac{\partial^2 l(\hat{\theta}^*)}{\partial \eta^2} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial \eta \partial v_1} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial \eta \partial v_2} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial \eta \partial \phi} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial \eta \partial \rho} \\ \frac{\partial^2 l(\hat{\theta}^*)}{\partial v_1 \partial \eta} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial v_1^2} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial v_1 \partial v_2} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial v_1 \partial \phi} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial v_1 \partial \rho} \\ \frac{\partial^2 l(\hat{\theta}^*)}{\partial v_2 \partial \eta} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial v_2 \partial v_1} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial v_2^2} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial v_2 \partial \phi} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial v_2 \partial \rho} \\ \frac{\partial^2 l(\hat{\theta}^*)}{\partial \phi \partial \eta} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial \phi \partial v_1} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial \phi \partial v_2} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial \phi^2} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial \phi \partial \rho} \\ \frac{\partial^2 l(\hat{\theta}^*)}{\partial \rho \partial \eta} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial \rho \partial v_1} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial \rho \partial v_2} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial \rho \partial \phi} & \frac{\partial^2 l(\hat{\theta}^*)}{\partial \rho^2} \end{bmatrix}.$$

Para se obter $\hat{\mu}_1$, $\hat{\mu}_2$, e $\hat{\sigma}_1$, basta substituir $\hat{\theta}^*$ nas Equações (3.3).

Como o interesse está na obtenção de $\hat{\theta} = (\hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1, \hat{\sigma}_2, \hat{\tau}_1, \hat{\tau}_2, \hat{\phi}, \hat{\rho})'$ e suas respectivas variâncias, o método Delta (Subseção 2.1.3) é aplicado para obter uma aproximação da distribuição de $\hat{\theta}$. Maiores detalhes podem ser encontrados em DeGroot e Schervish (2002), Cox e Hinkley (1974), Azzalini (1996) e Pawitan (2001). Assintoticamente, a distribuição de $\hat{\theta}$ será aproximadamente multivariada gaussiana com vetor de médias $\hat{\theta} = g(\hat{\theta}^*)$ e variância

$$\text{Var}(\hat{\theta}) \geq \nabla g(\hat{\theta}^*)' \mathbf{I}_{Fe}(\hat{\theta}^*)^{-1} \nabla g(\hat{\theta}^*),$$

em que \mathbf{I}_{Fe} é a matriz Informação de Fisher esperada e

$$\nabla g(\hat{\boldsymbol{\theta}}^*) = \left(\frac{\partial g(\hat{\boldsymbol{\theta}}^*)}{\partial \eta}, \frac{\partial g(\hat{\boldsymbol{\theta}}^*)}{\partial v_1}, \frac{\partial g(\hat{\boldsymbol{\theta}}^*)}{\partial v_2}, \frac{\partial g(\hat{\boldsymbol{\theta}}^*)}{\partial \phi}, \frac{\partial g(\hat{\boldsymbol{\theta}}^*)}{\partial \rho} \right)'$$

é a função escore $U(\hat{\boldsymbol{\theta}}^*)$. Assim, dado que:

- $\eta = \frac{\sigma_2}{\sigma_1} \Rightarrow \sigma_2 = \eta \sigma_1$, se $g(\hat{\boldsymbol{\theta}}^*) = \sigma_2$, então $\frac{\partial g(\hat{\boldsymbol{\theta}}^*)}{\partial \eta} = (\sigma_1, 0, 0, 0, 0)'$;
- $v_1 = \frac{\tau_1}{\sigma_1} \Rightarrow \tau_1 = v_1 \sigma_1$, se $g(\hat{\boldsymbol{\theta}}^*) = \tau_1$, então $\frac{\partial g(\hat{\boldsymbol{\theta}}^*)}{\partial v_1} = (0, \sigma_1, 0, 0, 0)'$;
- $v_2 = \frac{\tau_2}{\sigma_1} \Rightarrow \tau_2 = v_2 \sigma_1$, se $g(\hat{\boldsymbol{\theta}}^*) = \tau_2$, então $\frac{\partial g(\hat{\boldsymbol{\theta}}^*)}{\partial v_2} = (0, 0, \sigma_1, 0, 0)'$;
- $\phi = \phi$, se $g(\hat{\boldsymbol{\theta}}^*) = \phi$, então $\frac{\partial g(\hat{\boldsymbol{\theta}}^*)}{\partial \phi} = (0, 0, 0, 1, 0)'$;
- $\rho = \rho$, se $g(\hat{\boldsymbol{\theta}}^*) = \rho$, então $\frac{\partial g(\hat{\boldsymbol{\theta}}^*)}{\partial \rho} = (0, 0, 0, 0, 1)'$,

tem-se

$$\nabla g(\hat{\boldsymbol{\theta}}^*) = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_1 & 0 & 0 & 0 \\ 0 & 0 & \sigma_1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

e a matriz Informação de Fisher esperada para $\hat{\boldsymbol{\theta}}^*$, baseada nos dados \vec{Y} , é substituída pela matriz $\mathbf{I}_F(\hat{\boldsymbol{\theta}}^*)$ que é assintoticamente equivalente, de modo que

$$\text{Var}(\hat{\boldsymbol{\theta}}) \geq \nabla g(\hat{\boldsymbol{\theta}}^*)' \mathbf{I}_F(\hat{\boldsymbol{\theta}}^*)^{-1} \nabla g(\hat{\boldsymbol{\theta}}^*).$$

Para encontrar as variâncias para $\hat{\mu}$ e $\hat{\sigma}_1$, através da função (3.2), obtém-se

$$\mathbf{I}_F(\hat{\mu}) = -\frac{\partial^2 l(\vec{\theta})}{\partial \hat{\mu}^2} = \frac{1}{\sigma_1^2} (\mathbf{D}' \mathbf{V}^{-1} \mathbf{D})' \quad \text{e} \quad \mathbf{I}_F(\hat{\sigma}_1) = -\frac{\partial^2 l(\vec{\theta})}{\partial \sigma_1^2} = -\frac{n}{\sigma_1} + \frac{3Qe}{\sigma_1^3},$$

e, portanto,

$$\begin{aligned} \text{Var}(\hat{\mu}) &= \mathbf{I}_F(\hat{\mu})^{-1} = \hat{\sigma}_1^2 (\mathbf{D}' \hat{\mathbf{V}}^{-1} \mathbf{D})^{-1}, \\ \text{Var}(\hat{\sigma}_1) &= \mathbf{I}_F(\hat{\sigma}_1)^{-1} = \frac{\hat{\sigma}_1^3}{3\hat{Q}e - n\hat{\sigma}_1}. \end{aligned}$$

Na continuação do trabalho, a predição espacial, ou melhor, a cokrigagem foi feita como explicado e conforme o Sistema de Equações (2.40) na Subseção 2.5.2.

3.1.3 Transformação Dos Valores Preditos Para o Espaço Amostral Simplex

Uma vez que a transformação ALR foi aplicada aos dados originais e o procedimento de estimação e cokrigagem foi realizada com os dados transformados em \mathbb{R}^2 , deve-se fazer a transformação de volta do vetor de médias e da matriz de covariância para o espaço amostral original, o simplex \mathbb{S}^3 .

O objetivo é calcular para cada localização uma estimativa de

$$\vec{\mu}_{\vec{X}} = E(\vec{X}) = \int_{\mathbb{S}^B} \vec{X} f(\vec{X}) d\vec{X} \quad (3.4)$$

e

$$\Sigma_{\vec{X}} = Cov(\vec{X}, \vec{X}) = \int_{\mathbb{S}^B} (\vec{X} - \vec{\mu}_{\vec{X}})(\vec{X} - \vec{\mu}_{\vec{X}})' f(\vec{X}) d\vec{X}. \quad (3.5)$$

Sabe-se que

$$f(\vec{Y}) = (2\pi)^{-\frac{B-1}{2}} |\Sigma_{\vec{Y}}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\vec{Y} - \vec{\mu}_{\vec{Y}})' \Sigma_{\vec{Y}}^{-1} (\vec{Y} - \vec{\mu}_{\vec{Y}}) \right\},$$

segue uma distribuição gaussiana multivariada e $\vec{Y} = \text{alr}(\vec{X})$. Assim, para voltar a escala original dos dados, $\vec{X} = \text{agl}(\vec{Y})$, é usual utilizar o método do Jacobiano (JAMES, 2004) cuja transformação é dada por

$$J_{\text{alr}(\vec{X})} = \left| \frac{\partial \vec{Y}}{\partial \vec{X}} \right| = \left(\prod_{i=1}^B X_i \right)^{-1} \quad (3.6)$$

de modo que $f(\vec{X})$ se torna

$$f(\vec{X}) = (2\pi)^{-\frac{B-1}{2}} |\Sigma_{\vec{Y}}|^{-\frac{1}{2}} \left(\prod_{i=1}^B X_i \right)^{-1} \exp \left\{ -\frac{1}{2} (\text{alr}(\vec{X}) - \vec{\mu}_{\vec{Y}})' \Sigma_{\vec{Y}}^{-1} (\text{alr}(\vec{X}) - \vec{\mu}_{\vec{Y}}) \right\}.$$

Aitchison (1986), Pawlowsky-Glahn e Olea (2004) afirmam que para se resolver

as integrais (3.4) e (3.5) estas devem ser expressas como

$$\vec{\mu}_{\vec{X}} = \int_{\mathbb{R}^{B-1}} g_1(\vec{Z}) f(-\vec{Z}'\vec{Z}) d\vec{Z} \quad (3.7)$$

e

$$\Sigma_{\vec{X}} = \int_{\mathbb{R}^{B-1}} g_2(\vec{Z}) f(-\vec{Z}'\vec{Z}) d\vec{Z} \quad (3.8)$$

em que \vec{Z} é a transformação

$$\vec{Z} = \frac{1}{\sqrt{2}}(\mathbf{R}')^{-1}(\text{alr}(\vec{X}) - \vec{\mu}_{\vec{Y}}), \quad (3.9)$$

e \mathbf{R} é a decomposição Cholesky (BURDEN; FAIRES, 2003) de $\Sigma_{\vec{Y}}$, uma matriz triangular superior.

Desta forma, as integrais (3.7) e (3.8) podem, então, ser aproximadas pela integração de Gauss-Hermite multivariada de ordem k :

$$\int_{\mathbb{R}^{B-1}} g(\vec{Z}) f(-\vec{Z}'\vec{Z}) d\vec{Z} \approx \sum_{i_1=1}^k \sum_{i_2=1}^k \cdots \sum_{i_{B-1}=1}^k \omega_{i_1} \omega_{i_2} \cdots \omega_{i_{B-1}} g(Z_{i_1}, Z_{i_2}, \dots, Z_{i_{B-1}}), \quad (3.10)$$

em que os pesos $\omega_{i_1} \omega_{i_2} \cdots \omega_{i_{B-1}}$ e as abscissas $Z_{i_1}, Z_{i_2}, \dots, Z_{i_{B-1}}$ são conhecidos e seus valores podem ser encontrados, por exemplo, em Abramowitz e Stegun (1972, p. 924). Segundo Gamerman (1997) ordens de quadratura de 6 a 8 são suficientes para aproximar a integral.

Explicitando o procedimento, observa-se que a integral (3.4) pode ser reescrita como

$$\vec{\mu}_{\vec{X}} = \int_{\mathbb{S}^B} g(\vec{X}) \left((2\pi)^{-\frac{B-1}{2}} |\Sigma_{\vec{Y}}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \left(\text{alr}(\vec{X}) - \vec{\mu}_{\vec{Y}} \right)' \Sigma_{\vec{Y}}^{-1} \left(\text{alr}(\vec{X}) - \vec{\mu}_{\vec{Y}} \right) \right\} \right) d\vec{X},$$

em que

$$g(\vec{X}) = \vec{X} \left(\prod_{i=1}^B X_i \right)^{-1}. \quad (3.11)$$

Considere que $\Sigma_{\vec{Y}}^{-1} = (\mathbf{R}'\mathbf{R})^{-1} = \mathbf{R}^{-1}(\mathbf{R}')^{-1} = \mathbf{R}^{-1}(\mathbf{R}^{-1})'$. Da transformação (3.9), tem-se

$$\vec{X} = \text{agl}(\vec{\mu}_{\vec{Y}} + \sqrt{2}\mathbf{R}'\vec{Z}) = \text{agl}(\vec{Y})$$

e de (3.6),

$$\partial \vec{X} = \left(\prod_{i=1}^B X_i \right) \partial \vec{Y}. \quad (3.12)$$

Além disso,

$$J = \left| \frac{\partial \vec{Y}}{\partial \vec{Z}} \right| = \left| \frac{\partial(\vec{\mu}_{\vec{Y}} + \sqrt{2}\mathbf{R}'\vec{Z})}{\partial \vec{Z}} \right| = |\sqrt{2}\mathbf{R}'| = (\sqrt{2})^{B-1} |\mathbf{R}'|;$$

sendo \mathbf{R} a decomposição Cholesky de $\Sigma_{\vec{Y}}$,

$$|\mathbf{R}'| = |\mathbf{R}| = |\Sigma_{\vec{Y}}|^{\frac{1}{2}},$$

e

$$J = \left| \frac{\partial \vec{Y}}{\partial \vec{Z}} \right| = 2^{\frac{B-1}{2}} |\Sigma_{\vec{Y}}|^{\frac{1}{2}} \Rightarrow \partial \vec{Y} = 2^{\frac{B-1}{2}} |\Sigma_{\vec{Y}}|^{\frac{1}{2}} \partial \vec{Z}. \quad (3.13)$$

Substituindo a equação (3.13) em (3.12) tem-se

$$\partial \vec{X} = \left(\prod_{i=1}^B X_i \right) 2^{\frac{B-1}{2}} |\Sigma_{\vec{Y}}|^{\frac{1}{2}} \partial \vec{Z} \Rightarrow \partial \vec{Z} = 2^{-\frac{B-1}{2}} \left(\prod_{i=1}^B X_i \right)^{-1} |\Sigma_{\vec{Y}}|^{-\frac{1}{2}} \partial \vec{X}.$$

Por outro lado, de (3.11) tem-se $g(\text{agl}(\vec{\mu}_{\vec{Y}} + \sqrt{2}\mathbf{R}'\vec{Z})) = \text{agl}(\vec{\mu}_{\vec{Y}} + \sqrt{2}\mathbf{R}'\vec{Z}) \left(\prod_{i=1}^B X_i \right)^{-1}$ que substituída em (3.7) produz

$$\vec{\mu}_{\vec{X}} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \pi^{-\frac{B-1}{2}} \text{agl}(\vec{\mu}_{\vec{Y}} + \sqrt{2}\mathbf{R}'\vec{Z}) \exp\{\vec{Z}'\vec{Z}\} d\vec{Z} \quad (3.14)$$

com $g_1(\vec{Z}) = \pi^{-\frac{B-1}{2}} \text{agl}(\vec{\mu}_{\vec{Y}} + \sqrt{2}\mathbf{R}'\vec{Z})$. Logo, a aproximação de Gauss-Hermite (3.10) de ordem 3, por exemplo, para $\vec{\mu}_{\vec{X}}$ é

$$\begin{aligned} \vec{\mu}_{\vec{X}} &\approx \sum_{i_1=1}^3 \sum_{i_2=1}^3 \omega_{i_1} \omega_{i_2} g_1(\mathbf{Z}_{i_1}, \mathbf{Z}_{i_2}) \\ &\approx \omega_1^2 g_1(\mathbf{Z}_1, \mathbf{Z}_1) + \omega_1 \omega_2 [g_1(\mathbf{Z}_1, \mathbf{Z}_2) + g_1(\mathbf{Z}_2, \mathbf{Z}_1)] + \omega_1 \omega_3 [g_1(\mathbf{Z}_1, \mathbf{Z}_3) + g_1(\mathbf{Z}_3, \mathbf{Z}_1)] + \\ &\quad + \omega_2^2 g_1(\mathbf{Z}_2, \mathbf{Z}_2) + \omega_2 \omega_3 [g_1(\mathbf{Z}_2, \mathbf{Z}_3) + g_1(\mathbf{Z}_3, \mathbf{Z}_2)] + \omega_3^2 g_1(\mathbf{Z}_3, \mathbf{Z}_3). \end{aligned}$$

Lembrando que $\vec{\mu}_{\vec{X}}$ representa a média da composição em cada localização, esta é um vetor trivariado. Então na função g_1 , $\vec{\mu}_{\vec{Y}}$ é um vetor bivariado extraído de $\vec{\mu}_{\vec{Y}_0|\vec{Y}}$, cujo primeiro elemento corresponde à média da cokrigagem para a variável Y_1 , o segundo para a variável Y_2 e \mathbf{R}' de ordem 2×2 será a correspondente matriz de covariâncias de Y_1, Y_2 , extraída do bloco diagonal da matriz $\Sigma_{\vec{Y}_0|\vec{Y}}$. Assim, em cada localização a resolução de (3.14) implica na resolução de 3 integrais.

O mesmo procedimento é aplicado para a obtenção de $\Sigma_{\vec{X}}$ dada pela integral (3.8), agora reescrita como

$$\Sigma_{\vec{X}} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \pi^{-\frac{B-1}{2}} \left(\text{agl}(\vec{\mu}_{\vec{Y}} + \sqrt{2}\mathbf{R}'\vec{Z}) - \vec{\mu}_{\vec{X}} \right) \left(\text{agl}(\vec{\mu}_{\vec{Y}} + \sqrt{2}\mathbf{R}'\vec{Z})' - \vec{\mu}_{\vec{X}}' \right) \exp\{\vec{Z}'\vec{Z}\} d\vec{Z} \quad (3.15)$$

com g_2 dada por $g_2(\vec{Z}) = \pi^{-\frac{b-1}{2}} \left(\text{agl}(\vec{\mu}_Y + \sqrt{2}\mathbf{R}'\vec{Z}) - \vec{\mu}_X \right) \left(\text{agl}(\vec{\mu}_Y + \sqrt{2}\mathbf{R}'\vec{Z}) - \vec{\mu}_X \right)'$ e, neste caso, sendo Σ_X uma matriz de ordem 3×3 , a resolução da integral (3.15) para cada localização implica na resolução de 9 integrais. Portanto, a estimação do vetor de médias e variâncias em n_2 localizações implica na resolução de $n_2 \times 12$ integrais.

Em resumo, uma análise geoestatística de dados composicionais pode ser feita seguindo os passos:

- a. escrever o modelo geoestatístico bivariado para dados composicionais;
- b. escrever a função de verossimilhança do modelo;
- c. maximizar a função de verossimilhança, obtendo-se as estimativas dos parâmetros;
- d. fazer a cokrigagem obtendo-se o vetor de valores esperados preditos e a matriz de covariância em \mathbb{R}^2 ;
- e. transformar de volta os valores preditos para a escala original, \mathbb{S}^3 . Isto pode ser feito por:
 - e.1 integração de Gauss-Hermite;
 - e.2 simulação onde para cada localização deve-se:
 - e.2.1 gerar dados de uma distribuição gaussiana multivariada com vetor de médias e matriz de covariância iguais ao vetor de médias e matriz de covariância obtidos por cokrigagem (item (d));
 - e.2.2 aplicar a transformação AGL ao vetor de valores esperados obtidos no item (e.2.1);
 - e.2.3 calcular a média de cada componente ou funcionais destes.
- f. construir mapas dos valores esperados preditos para cada componente da composição.

Neste trabalho, foram feitas 1000 simulações condicionadas aos dados observados de areia, silte e argila (GONÇALVES, 1997) tendo argila como denominador (Seção 4.1). Essas simulações correspondem a 1000 prováveis realizações do processo na área considerando uma grade de 2500 pontos. Estas simulações possibilitam a obtenção de vários funcionais e, a título de exemplificação, considerou-se inicialmente o cálculo de alguns funcionais de interesse como frações máximas e mínimas de areia, silte e argila. A fração máxima de areia na primeira simulação, por exemplo, correspondeu ao valor máximo de areia observado nos 2500 pontos e a fração mínima, ao menor valor observado de areia. Isto foi feito para os outros dois componentes e para cada uma das 1000 simulações. Com estes resultados foram construídos mapas que representam a distribuição das frações máximas (mínimas) na área de estudo. O mesmo procedimento foi aplicado no cálculo de outros funcionais como as frações medianas que separaram as frações de

areia, silte e argila nas 50% inferiores ao valor mediano de cada componente. Ou seja, calculou-se a fração mediana dos componentes em cada simulações e os resultados foram apresentados em mapas. Também foram calculadas aquelas frações que separaram os dados nas 5% e 95% menores frações de areia, silte e argila. Além disso, calculou-se para cada simulação os percentuais na área de cada categoria de classificação (Figura 1.2). Com os resultados obtidos para cada categoria, nas 1000 simulações, foram construídos histogramas das densidades das percentagens observadas.

3.2 Análise de Dados Simulados

Em um estudo da metodologia proposta na Seção 3.1, foram simulados três conjuntos de dados com diferentes padrões composicionais com o objetivo de ilustrar a análise nesses cenários. Cada conjunto contém 100 valores de (X_1, X_2, X_3) , cujas localizações foram obtidas ao gerar um campo aleatório gaussiano com média zero, variância um e parâmetro de alcance $\phi = 0,6$ (escolha arbitrária), considerando-se um quadrado de lado unitário como área de estudo. A disposição de cada composição simulada nessa área, representada por pontos, pode ser vista na Figura 3.1.

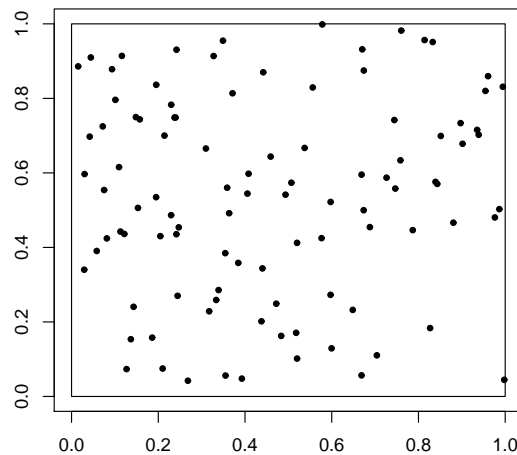


Figura 3.1: Distribuição das localizações no quadrado unitário.

3.2.1 Conjunto de Dados 1

O primeiro conjunto de dados, denominado Configuração 1, foi gerado pelo modelo 3.1 com vetor de parâmetros $\vec{\theta}=(\mu_1, \mu_2, \sigma_1, \sigma_2, \tau_1, \tau_2, \phi, \rho) = (-0,2; -0,5; 1; 1,5; 0,3; 0,3; 0,6; 0,9)$. A atribuição de valores próximos e de mesmo sinal para as médias fez com que a concentração dos valores das composições se situasse na parte central do triângulo (Figura 3.2). O fato de estarem pouco dispersos deve-se ao alto valor de ρ : quanto menor, mais espalhados. O fato das variâncias dos efeitos espaciais serem próximas, com $\sigma_1 < \sigma_2$, fez com que os pontos se aproximassem mais do vértice X_2 do que do X_1 . Os valores iguais para as variâncias τ_1 e τ_2 justifica os pontos no interior do diagrama.

Em cada uma das localizações, tem-se os valores de X_1 , X_2 e X_3 e as 100 composições representadas por círculos em um diagrama ternário podem ser vistas na Figura 3.2.

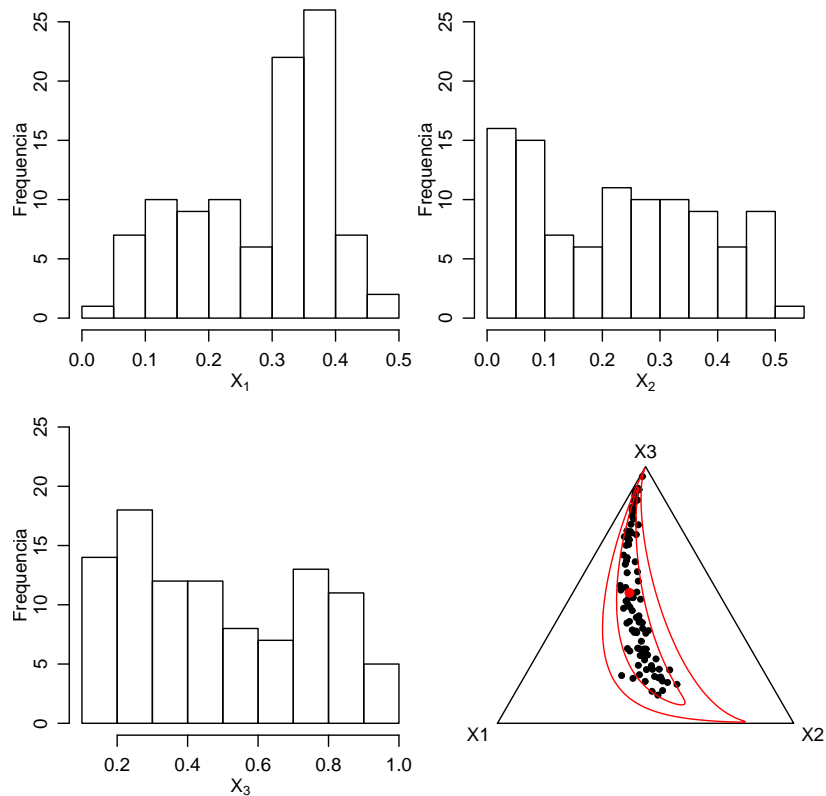


Figura 3.2: Distribuição de X_1 , X_2 e X_3 e diagrama ternário das composições para a Configuração 1.

Composições próximas a um vértice têm altas proporções do componente correspondente àquele vértice, de modo que esta figura indica que as maiores percentagens na amostra ocorreram para o componente X_3 . Também, pode-se observar, por exemplo, que a variabilidade da razão X_3/X_1 foi maior que a variabilidade da razão X_2/X_1 , já que a amplitude da intersecção das linhas partindo do vértice X_2 , passando por todos os pontos, com o lado $\overline{X_1X_3}$ é maior que

a amplitude da intersecção das linhas com o lado $\overline{X_1X_2}$. Por outro lado, as variabilidades das razões X_1/X_3 , X_2/X_3 foram similares. Além disso, uma região de confiança de 4-desvios-padrão contemplou todas as composições e o centro da distribuição dado pelo fechamento da média geométrica dos três componentes está representado pelo ponto vermelho no diagrama. Pelos histogramas nota-se que as percentagens dos três componentes apresentam desvios nítido em relação a distribuição gaussiana.

Ao fazer a transformação ALR nos dados simulados obteve-se as variáveis $Y_1 = \ln(X_1/X_3)$ e $Y_2 = \ln(X_2/X_3)$ e as distribuições dos dados transformados podem ser vistas na Figura 3.3. O diagrama de dispersão, por sua vez, mostra pontos alinhados em forma linear crescente indicando uma correlação linear positiva entre Y_1 e Y_2 .

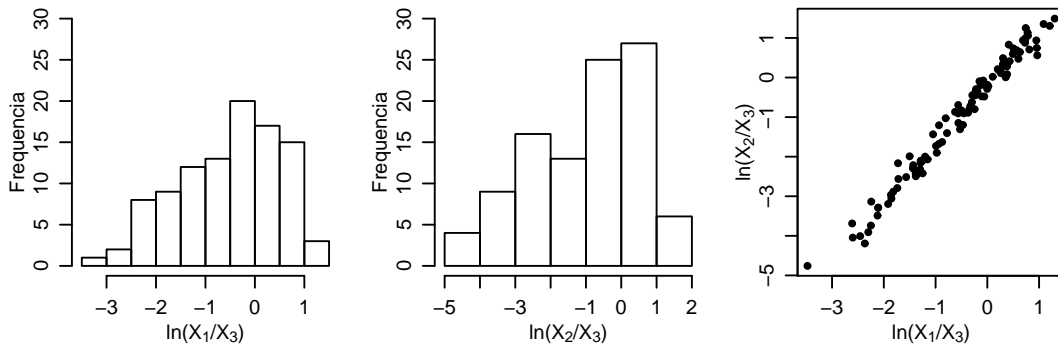


Figura 3.3: Distribuição das log razões e correspondente diagrama de dispersão para a Configuração 1.

Em seguida, fez-se o ajuste do modelo. Dentre os métodos de otimização testados adotou-se o método L-BFGS-B por não apresentar problemas de convergência. As estimativas obtidas para os parâmetros bem como os respectivos intervalos de confiança calculados pelo método Delta são apresentados na Tabela 3.1. Observa-se, em geral, estimativas não muito próximas aos valores atribuídos aos parâmetros mas com mesmos sinais e que três de oito intervalos contêm os valores atribuídos.

Tabela 3.1: Estimativas, erros padrão e intervalos de confiança para a Configuração 1, pelo método Delta via método de otimização “L-BFGS-B”.

Parâmetro	Valor	Estimativa	Erro Padrão	LI. Delta	LS. Delta
μ_1	-0,2	-0,993	1,607	-2,599	0,614
μ_2	-0,5	-1,689	1,983	-3,672	0,294
σ_1	1	1,153	0,102	1,051	1,255
σ_2	1,5	1,758	0,054	1,704	1,812
τ_1	0,3	0,400	0,029	0,371	0,430
τ_2	0,3	0,427	0,040	0,387	0,468
ϕ	0,6	0,953	0,513	0,440	1,466
ρ	0,9	0,957	0,026	0,931	0,984

No processo de otimização os valores iniciais para o vetor de médias foram considerados como as médias dos valores observados para Y_1 e Y_2 . Metade da variância calculada para Y_1 foi atribuída para o efeito espacial e a outra metade para o efeito composicional. Da mesma forma, procedeu-se com Y_2 . O valor inicial para ρ foi calculado como o coeficiente de correlação de Pearson e $\phi = \min + 0,2(\max - \min)$, onde “min” e “max” foram, respectivamente, a menor e maior distância entre duas localizações. Então, estes valores indicaram uma log-verossimilhança concentrada de $-301,35132$. Após o processo de otimização, a log-verossimilhança maximizada foi $59,64009$.

Como resultado da cokrigagem obteve-se, para cada localização, o vetor de médias e a matriz de covariância no espaço \mathbb{R}^2 . A volta dos valores preditos para o simplex \mathbb{S}^3 foi feita em uma grade constituída de 1156 pontos usando aproximação de Gauss-Hermite com ordem de quadratura $k = 7$. Na Figura 3.4 tem-se os mapas dos três componentes conforme a Configuração 1 e, assim como revelou o diagrama ternário, as maiores proporções ocorreram para o componente X_3 . A Figura 3.5, por sua vez, mostra boa concordância entre os valores observados e preditos para os três componentes.

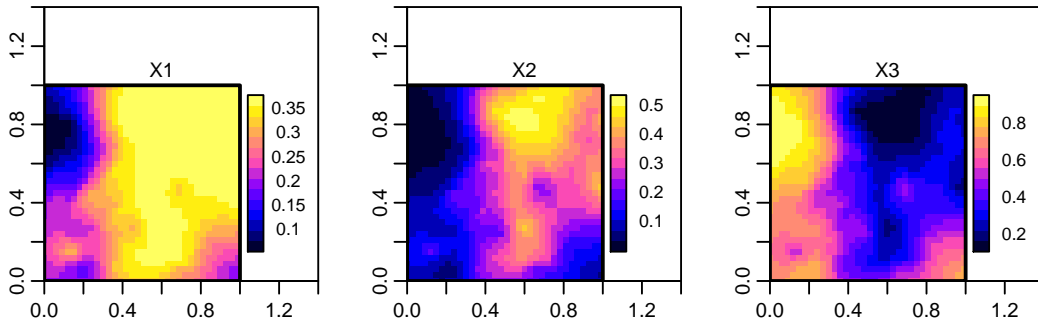


Figura 3.4: Mapas das percentagens de X_1 , X_2 e X_3 para dados da Configuração 1.

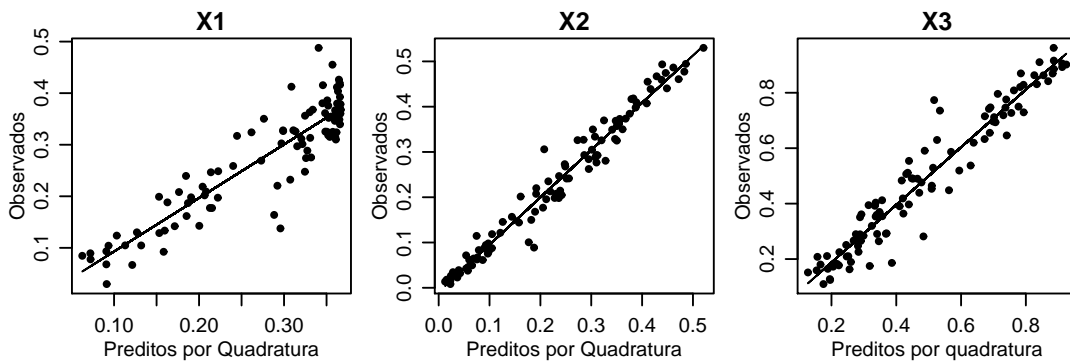


Figura 3.5: Valores observados versus preditos de X_1 , X_2 e X_3 para a Configuração 1.

3.2.2 Conjunto de Dados 2

O segundo conjunto de dados, Configuração 2, foi gerado considerando-se o vetor de parâmetros $(\mu_1, \mu_2, \sigma_1, \sigma_2, \tau_1, \tau_2, \phi, \rho) = (1; 1; 1, 2; 1, 5; 0, 9; 1; 0, 6; 0, 5)$. Assim como na Configuração 1, em relação às médias, buscou-se fazer com que a dispersão dos valores das composições continuasse na parte central do diagrama; a proximidade dos pontos ao vértice X_3 justificada pelo baixo valor de ϕ e o espalhamento pelo baixo valor de ρ . Também aqui, $\sigma_1 < \sigma_2$ fez com que os pontos se aproximassem mais de X_2 do que de X_1 e a diferença entre τ_1 e τ_2 forçou os pontos a se aproximarem mais do lado $\overline{X_1 X_2}$, com maior aproximação para X_2 por τ_1 ser o menor valor.

Para este vetor de parâmetros a configuração das composições no diagrama ternário da Figura 3.6 apresentou-se mais espalhada em relação à anterior com maiores percentagens relativas aos componentes X_2 e X_3 . Ao mesmo tempo, observa-se muitas composições com percentagens muito baixas de X_3 o que se confirma pelo respectivo histograma.

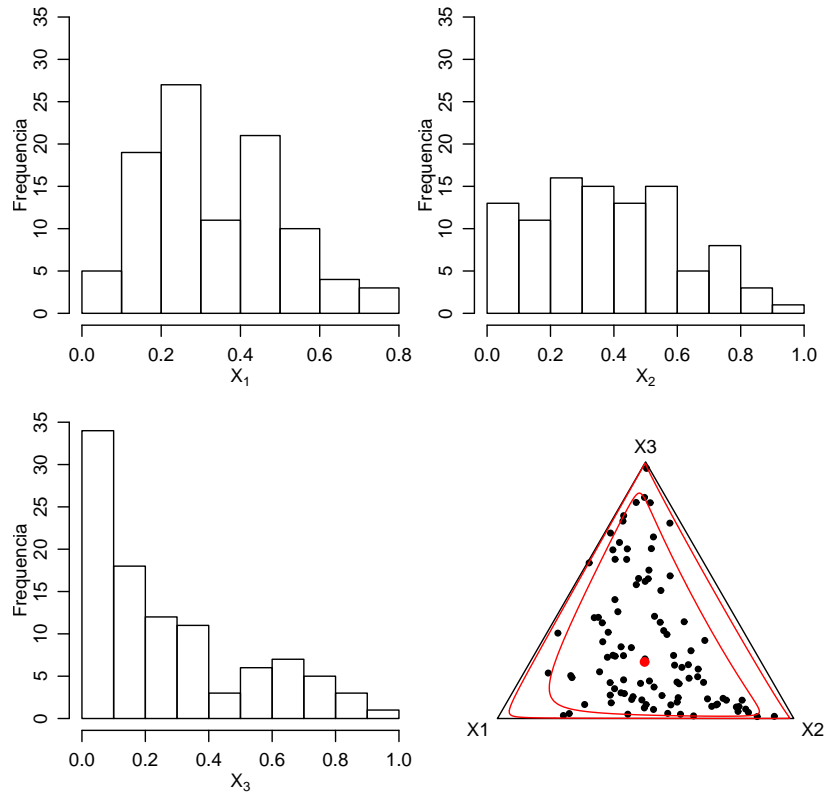


Figura 3.6: Distribuição de X_1 , X_2 e X_3 e diagrama ternário das composições para a Configuração 2.

A Figura 3.7 mostra a distribuição dos dados transformados e a dispersão de pontos no diagrama de dispersão também se apresenta de forma linear crescente mas com um espalhamento maior em relação ao da Configuração 1.

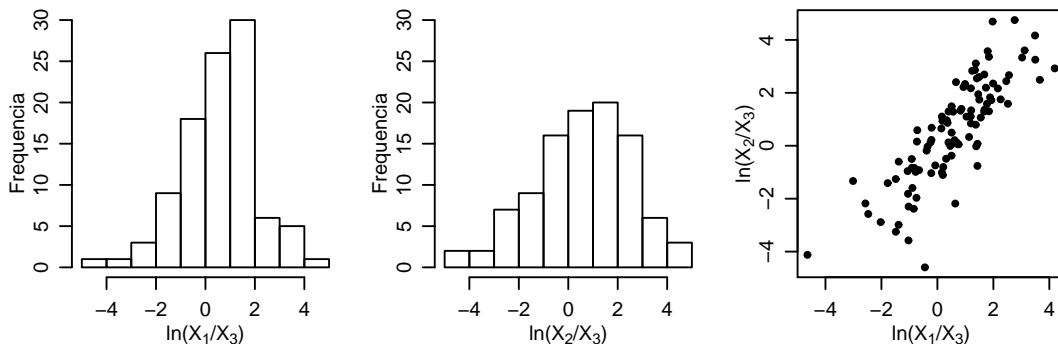


Figura 3.7: Distribuição das log razões e correspondente diagrama de dispersão para a Configuração 2.

Na Tabela 3.2 tem-se os resultados do ajuste do modelo para a Configuração 2 e o que se observa é que as estimativas permanecem com mesmos sinais; para as médias não são

boas mas os respectivos intervalos de confiança contêm os valores atribuídos para os parâmetros. Somente o intervalo para σ_2 não conteve o valor do parâmetro apesar de estar próximo ao limite inferior. Os valores das log-verossimilhanças concentrada e otimizada foram iguais a $-330,3923$ e $298,63124$, respectivamente.

Tabela 3.2: Estimativas, erros padrão e intervalos de confiança para a Configuração 2, pelo método Delta via método de otimização “L-BFGS-B”.

Parâmetro	Valor	Estimativa	Erro Padrão	LI. Delta	LS. Delta
μ_1	1	0,388	1,463	-1,075	1,852
μ_2	1	0,284	1,756	-1,471	2,040
σ_1	1,2	1,258	0,105	1,154	1,363
σ_2	1,5	1,831	0,265	1,566	2,096
τ_1	0,9	0,982	0,181	0,802	1,163
τ_2	1	1,005	0,213	0,792	1,218
ϕ	0,6	0,523	0,443	0,081	0,967
ρ	0,5	0,540	0,156	0,384	0,696

Na Figura 3.8 tem-se os mapas onde se observam poucas mudanças em relação aos mapas obtidos para a Configuração 1 e a Figura 3.9 mostra que concordância já não é tão boa para X_1 quanto para X_2 e X_3 .

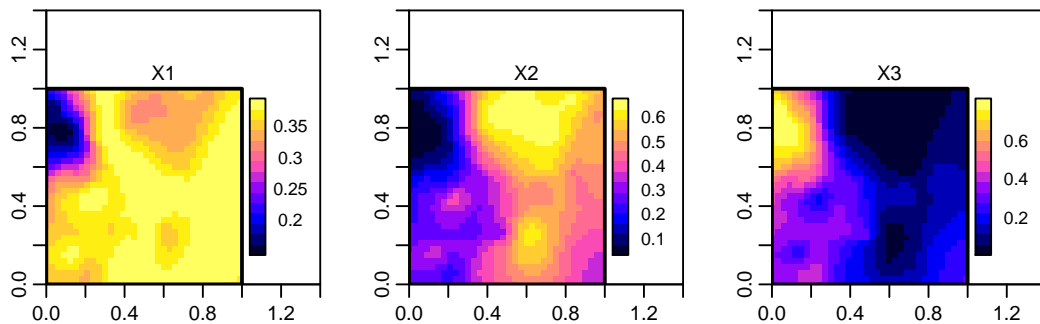


Figura 3.8: Mapas das percentagens de X_1 , X_2 e X_3 para dados da Configuração 2.

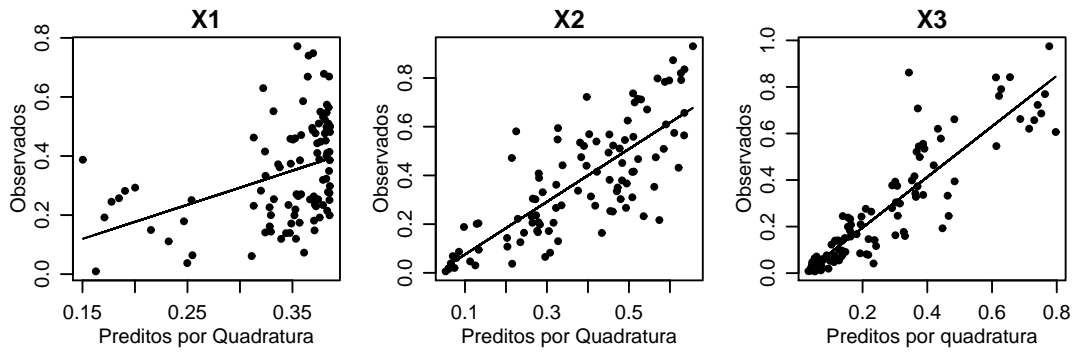


Figura 3.9: Valores observados versus preditos de X_1 , X_2 e X_3 para a Configuração 2.

3.2.3 Conjunto de Dados 3

A última configuração deste estudo caracterizou-se pelo conjunto de parâmetros $(\mu_1, \mu_2, \sigma_1, \sigma_2, \tau_1, \tau_2, \phi, \rho) = (-0,2; -1; 0,45; 0,13; 0,3; 0,3; 0,6; 0,95)$. O baixo valor absoluto de μ_1 fez com que os dados se aproximassem do lado esquerdo do diagrama com proximidade a X_1 ; caso contrário, a aproximação seria para o lado direito. Também neste caso, o baixo valor de ϕ aproximou os pontos ao vértice X_3 enquanto o alto valor de ρ tornou-os concentrados novamente. Como nesta configuração $\sigma_1 > \sigma_2$, os pontos se afastaram do vértice X_2 e continuaram no interior do diagrama pelos valores iguais e baixos para τ_1 e τ_2 .

Como pode ser visto na Figura 3.10 tem-se as distribuições dos componentes X_1 , X_2 e X_3 e no diagrama ternário observa-se baixos percentuais do componente X_2 .

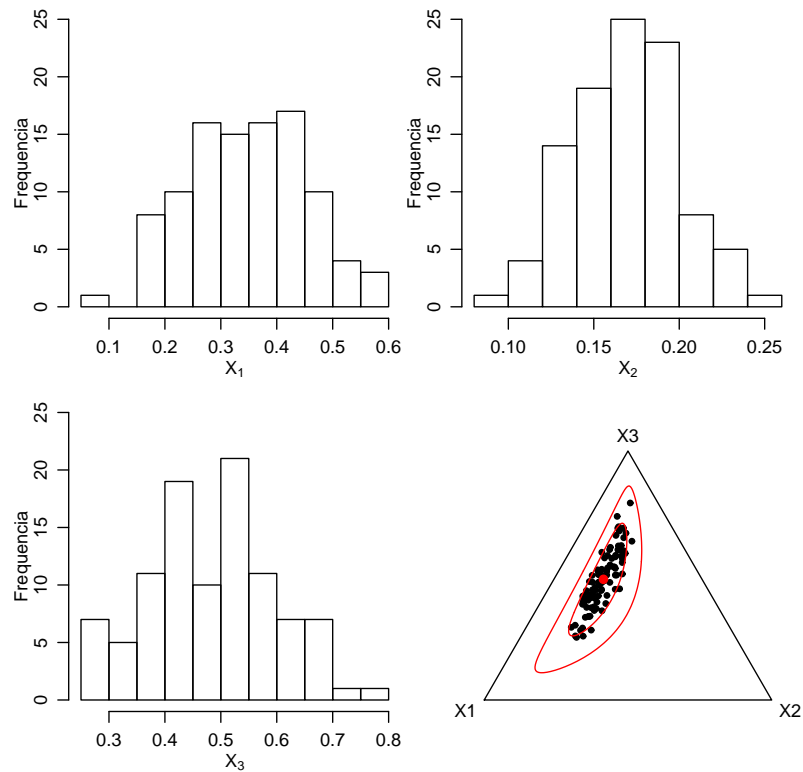


Figura 3.10: Distribuição de X_1 , X_2 e X_3 e diagrama ternário das composições para Configuração 3.

A distribuição dos dados após transformação é mostrada na Figura 3.11 e os pontos no diagrama de dispersão também apresentam-se em forma linear crescente mas com espalhamento maior que o da Configuração 1.

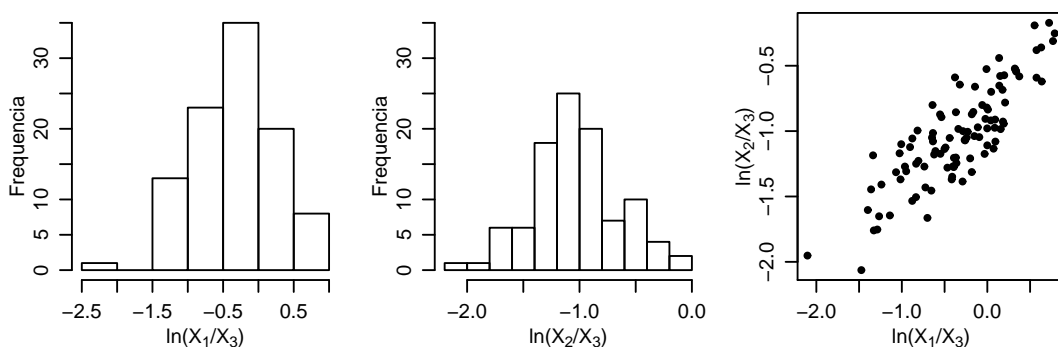


Figura 3.11: Distribuição das log razões e correspondente diagrama de dispersão para a Configuração 3.

Pela Tabela 3.3, também neste caso, as estimativas conservaram os mesmos sinais dos valores verdadeiros dos parâmetros e observa-se que os intervalos de confiança para as variâncias dos efeitos aleatórios Z_1 e Z_2 não contiveram os valores atribuídos para os parâmetros τ_1 e τ_2 .

Neste caso, o valor da log-verossimilhança concentrada foi $-92,32910$ e o da log-verossimilhança otimizada foi $14,34055$.

Tabela 3.3: Estimativas, erros padrão e intervalos de confiança para a Configuração 3, pelo método Delta via método de otimização “L-BFGS-B”.

Parâmetro	Valor	Estimativa	Erro Padrão	LI. Delta	LS. Delta
μ_1	-0,2	-0,589	0,691	-1,281	0,102
μ_2	-1	-1,115	0,402	-1,517	-0,712
σ_1	0,45	0,531	0,096	0,435	0,626
σ_2	0,13	0,177	0,054	0,123	0,230
τ_1	0,3	0,349	0,036	0,313	0,385
τ_2	0,3	0,345	0,031	0,314	0,375
ϕ	0,6	0,740	0,520	0,219	1,261
ρ	0,95	0,938	0,037	0,901	0,976

Na Figura 3.12 tem-se os mapas referentes à Configuração 3 podendo-se verificar uma mudança das previsões de X_1 e X_2 em relação às outras configurações. Neste caso, a pior concordância entre os valores observados e os preditos ocorreu para o segundo componente (Figura 3.13).

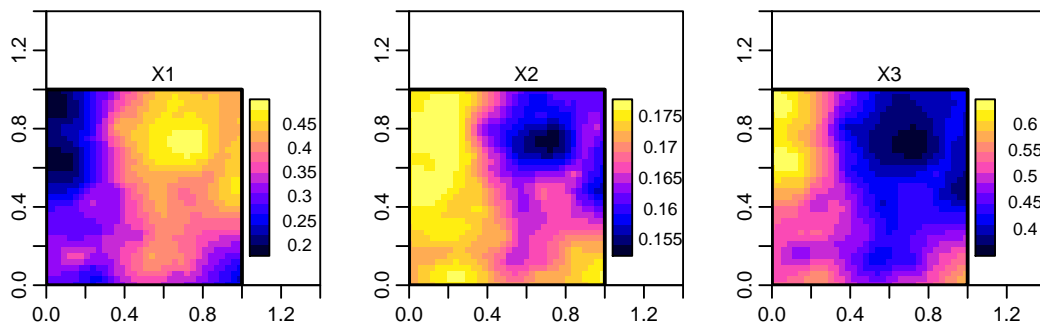


Figura 3.12: Mapas das percentagens de X_1 , X_2 e X_3 para dados da Configuração 3.

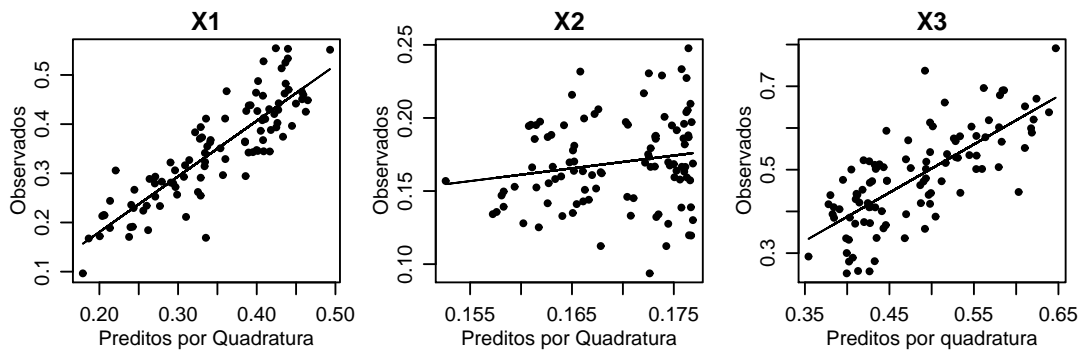


Figura 3.13: Valores observados versus preditos de X_1 , X_2 e X_3 para a Configuração 3.

3.2.4 Conclusões

Analisando o comportamento do modelo nas três configurações observa-se, de forma geral, que o componente X_1 apresentou-se o mais similar dentre as configurações com percentuais de médios para baixos. Com relação a X_2 , a Configuração 2 evidenciou a maior variação nas percentagens e as maiores percentagens para X_3 apareceram na Configuração 1.

A Configuração 1 foi a que mais se aproximou de uma forma linear, foi a que apresentou as piores estimativas, enquanto que a Configuração 3 que esperaria-se pior não resultou a pior em termos de qualidade de ajuste. As melhores estimativas foram obtidas com os dados da Configuração 2. Os intervalos construídos via aproximação quadrática pelo método Delta não foram bons devido a baixa cobertura, principalmente para os parâmetros de variância.

Em relação aos mapas de valores preditos considerando-se as três configurações, ressalta-se que estes confirmam os resultados observados no diagrama ternário. As duas primeiras se mostram mais semelhantes por componente dada a dispersão dos pontos na parte central do triângulo. As mudanças observadas nos mapas correspondentes à Configuração 3 se justificam pelo deslocamento da dispersão dos pontos para o lado esquerdo implicando em baixos valores de X_2 . As maiores concordâncias entre valores observados e preditos resultaram da Configuração 1.

Observa-se que se a volta dos valores preditos para o simplex for feita por simulação, para cada localização, os passos a serem seguidos são:

- a. gerar dados bivariados de uma distribuição gaussiana multivariada com vetor de médias e matriz de covariância obtidos por cokrigança;
- b. aplicar a transformação AGL nos dados obtidos no item (a);
- c. construir os mapas de médias de predição para cada componente.

3.2.5 Intervalos de Cobertura

Na avaliação da incerteza sobre os parâmetros do modelo fez-se um estudo de 1000 simulações para verificar a cobertura dos intervalos de confiança. O vetor de parâmetros considerado foi o do modelo da Equação (3.1) $\vec{\theta} = (\mu_1, \mu_2, \sigma_1, \sigma_2, \tau_1, \tau_2, \phi, \rho) = (50; 70; 2,5; 3,5; 2,5; 3,5; 0,6; 0,5)'$.

Inicialmente foi gerada uma amostra de 300 localizações de dados bivariados do modelo proposto sendo o efeito aleatório composicional gerado de uma normal bivariada com vetor de médias iguais a zero e matriz de covariância calculada com os devidos valores em $\vec{\theta}$. O campo aleatório gaussiano foi gerado com média zero, variância um, parâmetro de alcance ϕ dado em $\vec{\theta}$, e função de covariância exponencial.

Em seguida, foi realizado o processo de estimação (algoritmo L-BFGS-B) obtendo-se as estimativas dos parâmetros e foram construídos os respectivos intervalos de confiança via método Delta. Esse procedimento foi repetido num total de 1000 simulações.

Para cada parâmetro verificou-se quantos dos respectivos 1000 intervalos contiveram o seu verdadeiro valor e dividindo-se o resultado por 1000, obteve-se a cobertura dos intervalos. Pela Tabela 3.4, observa-se que a cobertura dos intervalos foi baixa exceto para τ_1 , τ_2 e ρ , o que sugere a construção por verossimilhança perfilhada ou inferência bayesiana. Tem-se também, as médias das 1000 estimativas com respectivos intervalos de 95% confiança construídos pelo método clássico.

Tabela 3.4: Parâmetros, média e intervalo de 95% de confiança das estimativas dos parâmetros, e cobertura (%) dos intervalos via método Delta para 1000 simulações de amostras de tamanho 300.

Parâmetro	Valor	Média	LI.Clássico	LS.Clássico	Cobertura
μ_1	50	50,042	49,947	50,137	69,2
μ_2	70	70,063	69,931	70,196	63,4
σ_1	2,5	2,026	1,989	2,062	5,4
σ_2	3,5	2,841	2,789	2,892	24,2
τ_1	2,5	2,466	2,458	2,474	94,6
τ_2	3,5	3,457	3,445	3,469	94,1
ϕ	0,6	0,387	0,367	0,407	43,1
ρ	0,5	0,489	0,486	0,492	91,6

Nas Figuras 3.14 e 3.15 pode-se observar as distribuições e os boxplots das estimativas.

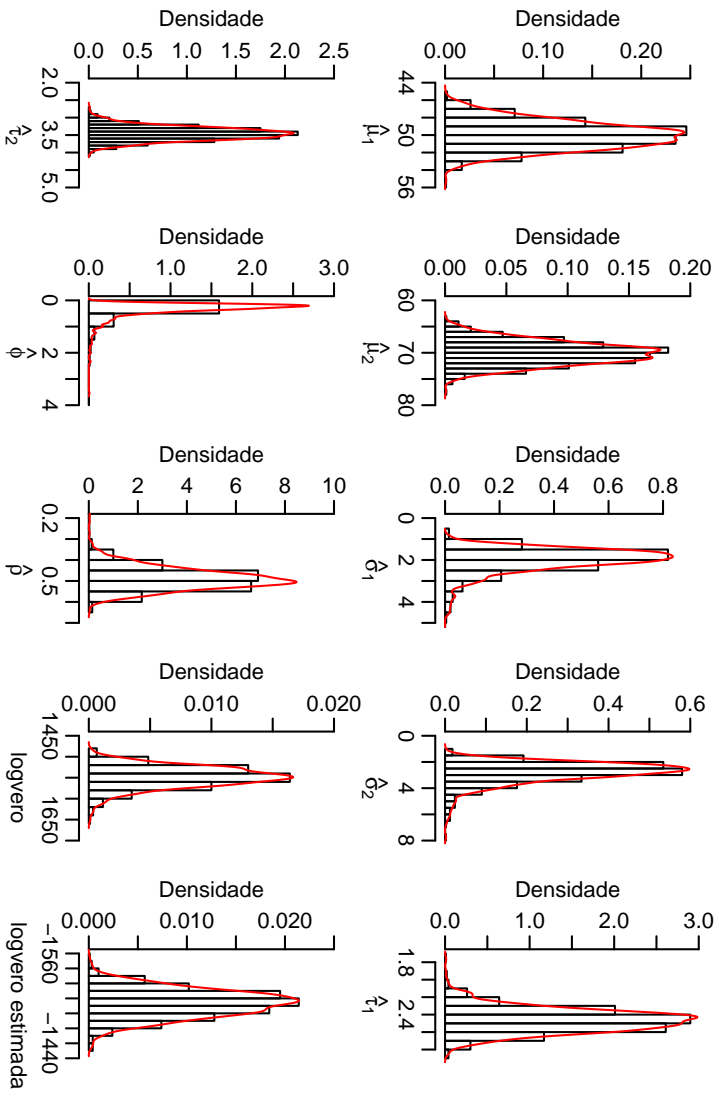


Figura 3.14: Distribuições das estimativas dos parâmetros para 1000 simulações e amostras de tamanho 300.

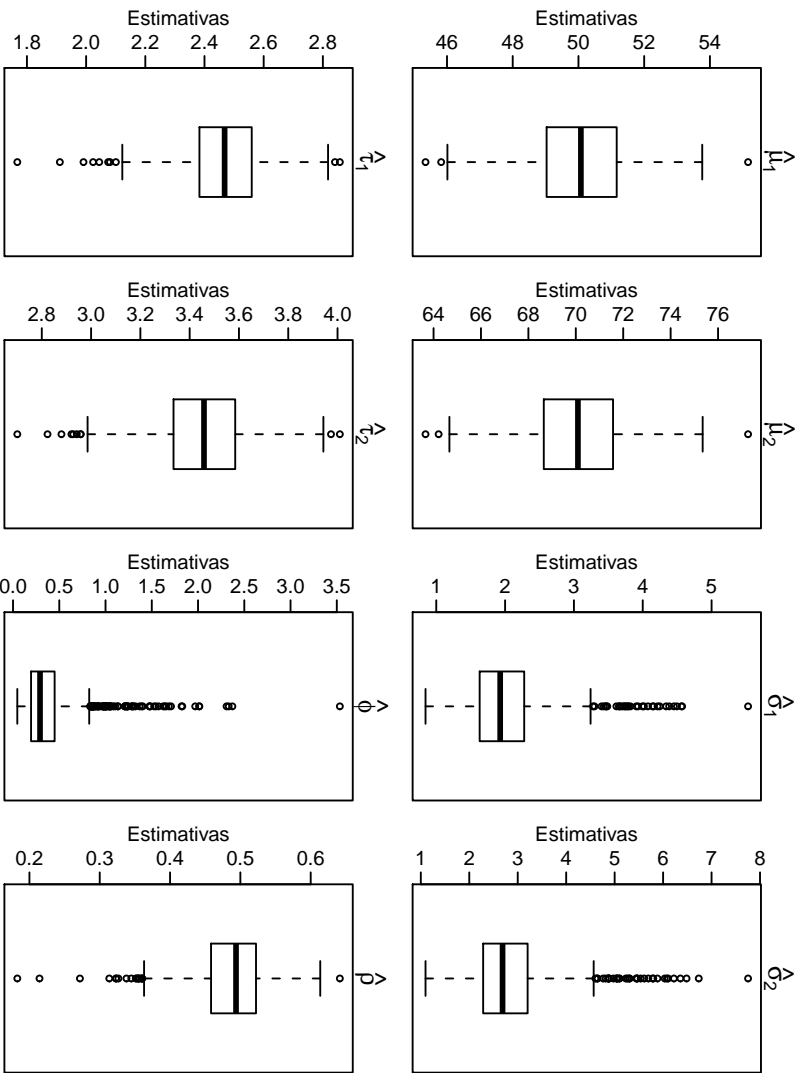


Figura 3.15: Boxplot das estimativas dos parâmetros para 1000 simulações e amostras de tamanho 300.

3.3 Inferência Bayesiana Para o Modelo Geoestatístico Composicional

Uma análise geoestatística de dados composicionais também pode ser feita por inferência bayesiana. Como visto na Subseção 2.3.1, sob o paradigma bayesiano, o vetor de parâmetros $\vec{\theta} = (\vec{\mu}, \sigma_1, \sigma_2, \tau_1, \tau_2, \phi, \rho)'$ é considerado um vetor aleatório, a função de log-verossimilhança do modelo é a mesma dada na Equação (2.26) e, para se fazer inferência bayesiana, é preciso construir a distribuição conjunta dos vetores aleatórios \vec{Y} e $\vec{\theta}$, que, substituída na Equação (2.27), fornece a distribuição *a posteriori* para $\vec{\theta}$

$$P(\vec{\theta}|\vec{Y}) = \frac{L(\vec{\theta}, \vec{Y})P(\vec{\theta})}{P(\vec{Y})} \propto L(\vec{\theta}, \vec{Y})P(\vec{\theta})$$

ou

$$P(\vec{\mu}, \sigma_1, \sigma_2, \tau_1, \tau_2, \phi, \rho|\vec{Y}) \propto P(\vec{Y}|\vec{\mu}, \sigma_1, \sigma_2, \tau_1, \tau_2, \phi, \rho)P(\vec{\mu}, \sigma_1, \sigma_2, \tau_1, \tau_2, \phi, \rho).$$

Observa-se que, para o modelo geoestatístico bivariado composicional, não é possível deduzir analiticamente as distribuições marginais *a posteriori* para todos os parâmetros do modelo. Neste caso, considerando a reparametrização apresentada na Subseção 3.1.1, $\vec{\theta}^* = (\eta, v_1, v_2, \phi, \rho)$ e, supondo independência entre as distribuições de $\vec{\theta}^*$ e $(\vec{\mu}, \sigma_1^2)$, pode-se escrever

$$P(\vec{\mu}, \sigma_1^2, \vec{\theta}^*|\vec{Y}) \propto P(\vec{Y}|\vec{\mu}, \sigma_1^2, \vec{\theta}^*)P(\vec{\mu}, \sigma_1^2)P(\vec{\theta}^*), \quad (3.16)$$

e então é possível obter expressões fechadas para as distribuições marginais para $\vec{\mu}$ e σ_1^2 . Para isto, observa-se que integrando ambos os lados da Equação (3.16) em relação a $\vec{\theta}^*$ tem-se

$$\int_{\vec{\theta}^*} P(\vec{\mu}, \sigma_1^2, \vec{\theta}^*|\vec{Y})d\vec{\theta}^* \propto \int_{\vec{\theta}^*} P(\vec{Y}|\vec{\mu}, \sigma_1^2, \vec{\theta}^*)P(\vec{\mu}, \sigma_1^2)P(\vec{\theta}^*)d\vec{\theta}^*.$$

e supondo $1/\sigma_1^2$ como distribuição *a priori* para $P(\vec{\mu}, \sigma_1^2)$ obtém-se

$$P(\vec{\mu}, \sigma_1^2|\vec{Y}) \propto P(\vec{Y}|\vec{\mu}, \sigma_1^2, \vec{\theta}^*) \frac{1}{\sigma_1^2} \int_{\vec{\theta}^*} P(\vec{\theta}^*)d\vec{\theta}^*. \quad (3.17)$$

Não sendo possível deduzir analiticamente a distribuição *a posteriori* para $\vec{\theta}^*$ e tampouco calcular analiticamente a integral em (3.17), esta é resolvida numericamente usando o algoritmo de Metropolis-Hastings. As distribuições *a priori* utilizadas para η , v_1 e v_2 foram lognormais com parâmetros correspondentes ao logaritmo das razões das estimativas de máxima verossimilhança obtidas no processo de maximização. Para ϕ utilizou-se uma distribuição Gama com parâmetros (66; 1) em que o valor 66 é a estimativa de máxima verossimilhança de ϕ e para ρ uma distribuição *a priori* vaga definida no intervalo $[-1, 1]$. Os valores estimados obtidos foram

então substituídos em (3.17) de modo que:

$$\begin{aligned} P(\vec{\mu}, \sigma_1^2 | \vec{\theta}^*, \vec{Y}) &\propto (\sigma_1^2)^{-1} P(\vec{Y} | \vec{\mu}, \sigma_1^2, \vec{\theta}^*) \\ &\propto (\sigma_1^2)^{-\left(\frac{n}{2}+1\right)} \exp \left\{ -\frac{1}{2} (\vec{Y} - \vec{\mu}_{\vec{Y}})' \mathbf{V}^{-1} (\vec{Y} - \vec{\mu}_{\vec{Y}}) \right\}. \end{aligned}$$

Da definição de probabilidade condicional tem-se

$$P(\vec{\mu} | \sigma_1^2, \vec{\theta}^*, \vec{Y}) = \frac{P(\vec{Y} | \vec{\mu}, \sigma_1^2, \vec{\theta}^*) P(\vec{\mu} | \sigma_1^2, \vec{\theta}^*) P(\sigma_1^2, \vec{\theta}^*)}{P(\vec{Y} | \sigma_1^2, \vec{\theta}^*) P(\sigma_1^2, \vec{\theta}^*)},$$

e conseqüentemente,

$$P(\vec{\mu} | \sigma_1^2, \vec{\theta}^*, \vec{Y}) \propto P(\vec{\mu} | \sigma_1^2, \vec{\theta}^*) P(\vec{Y} | \vec{\mu}, \sigma_1^2, \vec{\theta}^*).$$

Assim, supondo uma distribuição *a priori* vaga para $P(\vec{\mu} | \sigma_1^2, \vec{\theta}^*)$, ou seja $[\vec{\mu} | \sigma_1^2, \vec{\theta}^*] \propto 1$ vem

$$\begin{aligned} P(\vec{\mu} | \sigma_1^2, \vec{\theta}^*, \vec{Y}) &\propto P(\vec{Y} | \vec{\mu}, \sigma_1^2, \vec{\theta}^*) \\ &\propto (\sigma_1^2)^{-\frac{n}{2}} \exp \left\{ -\frac{1}{2\sigma_1^2} (\vec{\mu} - \hat{\vec{\mu}})' (\mathbf{D}' \mathbf{V}^{-1} \mathbf{D}) (\vec{\mu} - \hat{\vec{\mu}}) \right\} \times \\ &\quad \exp \left\{ -\frac{1}{2\sigma_1^2} (\vec{Y} - \mathbf{D} \hat{\vec{\mu}})' \mathbf{V}^{-1} (\vec{Y} - \mathbf{D} \hat{\vec{\mu}}) \right\} \\ &\propto (\sigma_1^2)^{-\frac{n}{2}} \exp \left\{ -\frac{1}{2\sigma_1^2} (\vec{\mu} - \hat{\vec{\mu}})' (\mathbf{D}' \mathbf{V}^{-1} \mathbf{D}) (\vec{\mu} - \hat{\vec{\mu}}) \right\}, \end{aligned}$$

e a distribuição *a posteriori* marginal para $\vec{\mu} | \sigma_1^2, \vec{\theta}^*, \vec{Y}$ é

$$[\vec{\mu} | \sigma_1^2, \vec{\theta}^*, \vec{Y}] \sim N(\hat{\vec{\mu}}; \sigma_1^2 (\mathbf{D}' \mathbf{V}^{-1} \mathbf{D})^{-1}). \quad (3.18)$$

Como

$$P(\vec{\mu}, \sigma_1^2 | \vec{\theta}^*, \vec{Y}) = P(\vec{\mu} | \sigma_1^2, \vec{\theta}^*, \vec{Y}) P(\sigma_1^2 | \vec{\theta}^*, \vec{Y}) \Rightarrow P(\sigma_1^2 | \vec{\theta}^*, \vec{Y}) = \frac{P(\vec{\mu}, \sigma_1^2 | \vec{\theta}^*, \vec{Y})}{P(\vec{\mu} | \sigma_1^2, \vec{\theta}^*, \vec{Y})}, \quad (3.19)$$

substituindo (3.18) em (3.19) tem-se:

$$\begin{aligned} P(\sigma_1^2 | \vec{\theta}^*, \vec{Y}) &\propto (\sigma_1^2)^{-\left(\frac{n-n_{\vec{\mu}}}{2}+1\right)} \exp \left\{ -\frac{1}{2\sigma_1^2} (\vec{Y} - \mathbf{D} \hat{\vec{\mu}})' \mathbf{V}^{-1} (\vec{Y} - \mathbf{D} \hat{\vec{\mu}}) \right\} \\ &\propto (\sigma_1^2)^{-\left(\frac{n-n_{\vec{\mu}}}{2}+1\right)} \exp \left\{ -\frac{1}{2\sigma_1^2} n \hat{\sigma}_1^2 \right\} \\ &\propto (\sigma_1^2)^{-\left(\frac{n-n_{\vec{\mu}}}{2}+1\right)} \exp \left\{ -\frac{1}{2\sigma_1^2} \frac{(n-n_{\vec{\mu}})}{(n-n_{\vec{\mu}})} n \hat{\sigma}_1^2 \right\} \\ &\propto (\sigma_1^2)^{-\left(\frac{n-n_{\vec{\mu}}}{2}+1\right)} \exp \left\{ -\frac{1}{2\sigma_1^2} (n-n_{\vec{\mu}}) S^2 \right\}, \end{aligned}$$

e a distribuição marginal *a posteriori* para $\sigma_1^2 | \vec{\theta}^*, \vec{Y}$ é

$$[\sigma_1^2 | \vec{\theta}^*, \vec{Y}] \sim \chi_{\text{SInv}}^2(n-n_{\vec{\mu}}; S^2), \quad (3.20)$$

em que χ_{Sinv}^2 é a distribuição qui-quadrado escalonada inversa e

$$S^2 = \frac{n}{n - n_{\bar{\mu}}} \hat{\sigma}_1^2 = \frac{(\vec{Y} - \mathbf{D}\hat{\bar{\mu}})' \mathbf{V}^{-1} (\vec{Y} - \mathbf{D}\hat{\bar{\mu}})}{n - n_{\bar{\mu}}}. \quad (3.21)$$

Por fim, com o desenvolvimento descrito pode-se resumir os passos para uma análise bayesiana de dados composicionais espacializados, segundo o modelo proposto, da seguinte forma:

- a. deduzir analiticamente as distribuições *a posteriori* para $\bar{\mu}|\sigma_1^2, \vec{\theta}^*, \vec{Y}$ e $\sigma_1^2|\vec{\theta}^*, \vec{Y}$;
- b. executar o algoritmo Metropolis-Hastings no vetor $\vec{\theta}^*$;
- c. com o vetor $\vec{\theta}^*$ estimado, calcular a matriz de covariância \mathbf{V} do modelo;
- d. calcular $\hat{\bar{\mu}}$ dado em (3.3);
- e. calcular S^2 dado em (3.21);
- f. executar um passo Gibbs que significa amostrar um valor de $\sigma_1^2|\vec{\theta}^*, \vec{Y}$ da distribuição (3.20);
- g. com $\hat{\bar{\mu}}$ e $\text{Var}(\bar{\mu}|\sigma_1^2, \vec{\theta}^*, \vec{Y})$, executar um passo Gibbs amostrando um valor de $\bar{\mu}|\sigma_1^2, \vec{\theta}^*, \vec{Y}$ de (3.18);
- h. repetir o procedimento até o número de simulações desejado;
- i. fazer a cokrigagem com cada conjunto de parâmetros simulados;
- j. gerar uma amostra de uma distribuição normal multivariada com cada resultado obtido no item (i), vetor de médias e matriz de covariância;
- k. aplicar a transformação AGL nos resultados obtidos no item (j);
- l. construir mapa de predição para cada componente da composição.

As análises devem ser acompanhadas de diagnósticos de convergência como descrito na Subseção 2.3.3.

4 ANÁLISE DE FRAÇÕES GRANULOMÉTRICAS DE UM SOLO

4.1 Análise Descritiva

Neste estudo de caso, analisou-se um conjunto de dados obtido no trabalho de Gonçalves (1997), que foi conduzido em uma área irrigada por sistema pivô-central na Fazenda Areão (Figura 4.1), pertencente ao campus da Escola Superior de Agricultura - “Luiz de Queiroz” (ESALQ-USP). Nessa área foi demarcado um quadrante na porção mais elevada (topo da encosta), no qual foram obtidas 82 amostras de solo na profundidade entre 0 e 0,20 m em uma malha regular quadrada de amostragem, de lado igual a 20 metros. Em cada amostra, foram medidos os valores das frações granulométricas de areia, silte e argila.

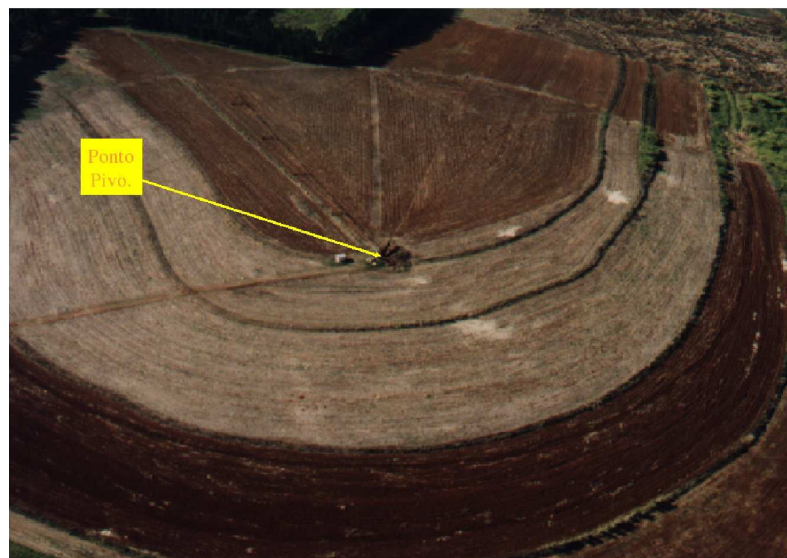


Figura 4.1: Foto aérea do campo experimental de irrigação da ESALQ-USP com área de estudo correspondente ao quadrante irrigado por um sistema pivô-central.
FONTE: Gonçalves (1997).

Na Figura 4.2 está representado o quadrante da área com as localizações amostradas. A coordenada mínima foi igual a (0,0) e máxima igual a (180,180) metros. Os círculos representam os 25% menores valores das frações granulométricas observados na amostra; os triângulos representam os 25 a 50% menores valores seguintes; as diagonais verticais, 50 a 75% menores valores seguintes e, as diagonais cruzadas, correspondem aos 75 a 100% menores valores. Pode-se observar, assim, em qual classe de valores pertence cada componente da composição e a respectiva localização na área.

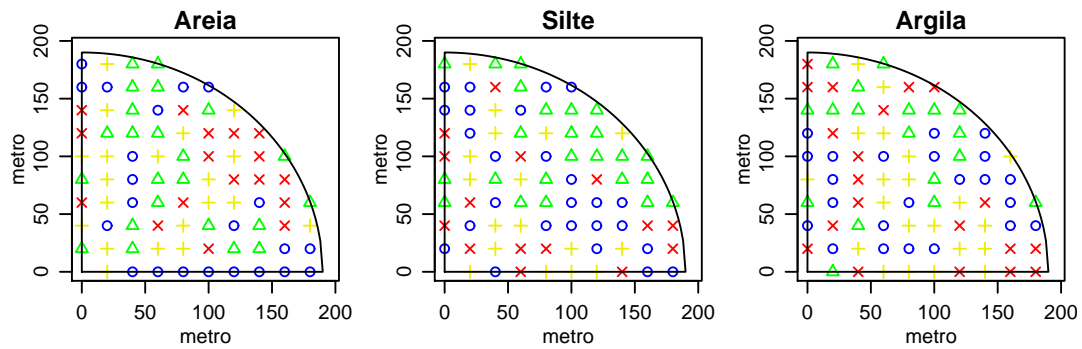


Figura 4.2: Percentagem de areia (à esquerda), silte (meio) e argila (à direita) segundo os 25% (círculos), 25 a 50% (triângulos), 50 a 75% (diagonais verticais), 75 a 100% (diagonais cruzadas) menores valores e a localização amostral.

A disposição dos valores das composições no diagrama de classificação textural pode ser vista na Figura 4.3. Observa-se que, segundo essa classificação, o solo varia de muito argiloso (60 a 70%), franco-argiloso (30 a 40%), com a maior frequência em argiloso (40 a 60%, aproximadamente).

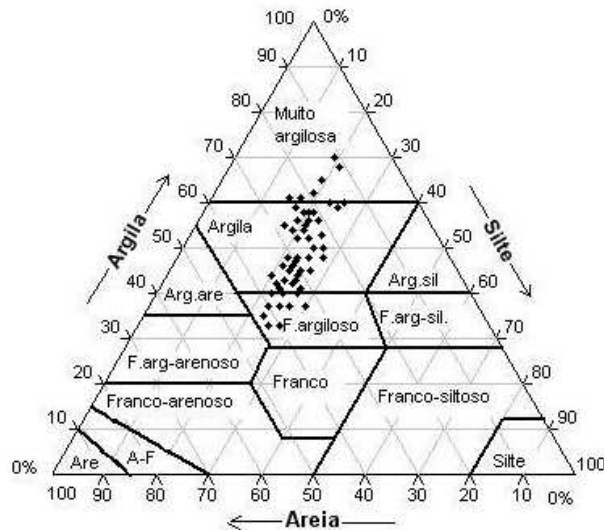


Figura 4.3: Diagrama de classificação textural.

Mais especificamente, a variação dos percentuais dos componentes é mostrada nas Figuras 4.4 e 4.5, podendo-se notar que existe uma separação clara entre os valores de silte e argila e, areia e silte apresentam um intervalo de valores comuns variando de 15 a 30%. Quanto às distribuições marginais dos conteúdos, os componentes areia e silte têm distribuições levemente assimétricas negativas com coeficientes de assimetria iguais a $-0,40$ e $-0,27$, respectivamente, enquanto silte tem distribuição assimétrica positiva ($As = 0,42$).

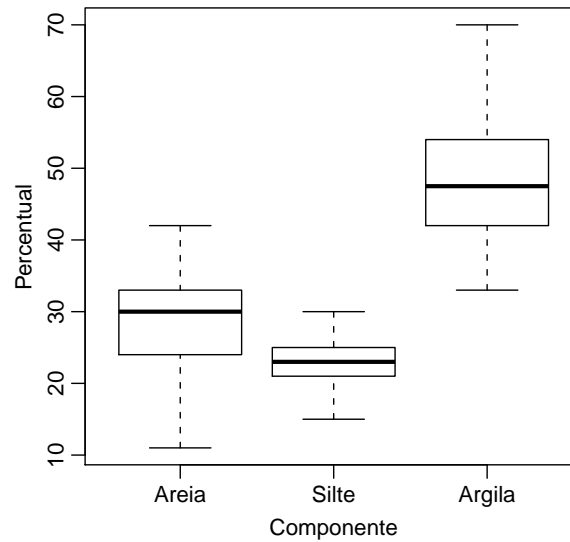


Figura 4.4: Boxplot dos percentuais de areia, silte e argila.

Para a construção do diagrama ternário, considerou-se a classe “acomp” do pacote *compositions*, do R (R development Core Team, 2008), na qual os dados são analisados na geometria relativa. Observa-se que uma região de confiança de 4-desvios padrão contempla todas as composições. O ponto vermelho no diagrama representa o centro da distribuição e é igual a $(0,2814;0,2330;0,4856)$.

Tabela 4.1: Estatísticas descritivas dos percentuais de areia, silte e argila.

Estatística	Areia	Silte	Argila
Mínimo	11,00	15,00	33,00
Quartil 1	24,00	21,00	42,00
Mediana	30,00	23,00	47,50
Média	28,57	23,06	48,37
Quartil 3	33,00	25,00	54,00
Máximo	42,00	30,00	70,00
Desvio padrão	6,87	2,96	8,37
Variância	47,21	8,75	69,99

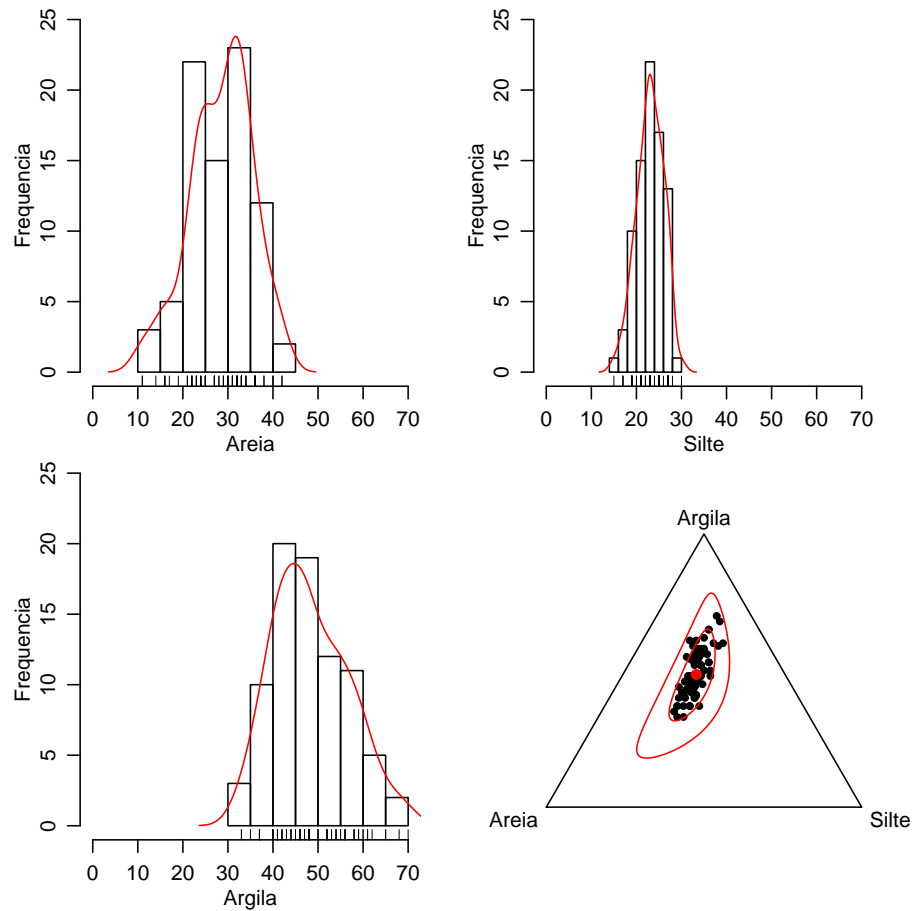


Figura 4.5: Distribuição dos percentuais de areia, silte e argila e diagrama ternário das composições.

Os resultados apresentados na Tabela 4.1, assim como as figuras apresentadas, revelam que argila apresenta os maiores valores, enquanto silte os menores. Os valores dos desvios-padrão (variância) indicam que argila apresentou maior variabilidade. A matriz de correlação mostra que os componentes areia e argila são os mais correlacionados, negativamente ($r = -0,9433$),

$$\begin{array}{l}
 \textit{Areia} \\
 \textit{Silte} \\
 \textit{Argila}
 \end{array}
 \begin{bmatrix}
 \textit{Areia} & \textit{Silte} & \textit{Argila} \\
 \begin{bmatrix}
 1 & 0,3451 & -0,9433 \\
 0,3451 & 1 & -0,6370 \\
 -0,9433 & -0,6370 & 1
 \end{bmatrix}
 \end{bmatrix}$$

Como argila determina a retenção de água no solo, pela sua alta superfície de penetração, essa foi escolhida como denominador das log razões. Assim, ao fazer a transformação ALR nos dados originais obteve-se as variáveis $Y_1 = \ln\left(\frac{\textit{Areia}}{\textit{Argila}}\right)$, $Y_2 = \ln\left(\frac{\textit{Silte}}{\textit{Argila}}\right)$, cujas distribuições são mostradas nos histogramas da Figura 4.6. O diagrama de dispersão na mesma figura evidencia uma correlação linear positiva entre Y_1 e Y_2 ($r = 0,82$).

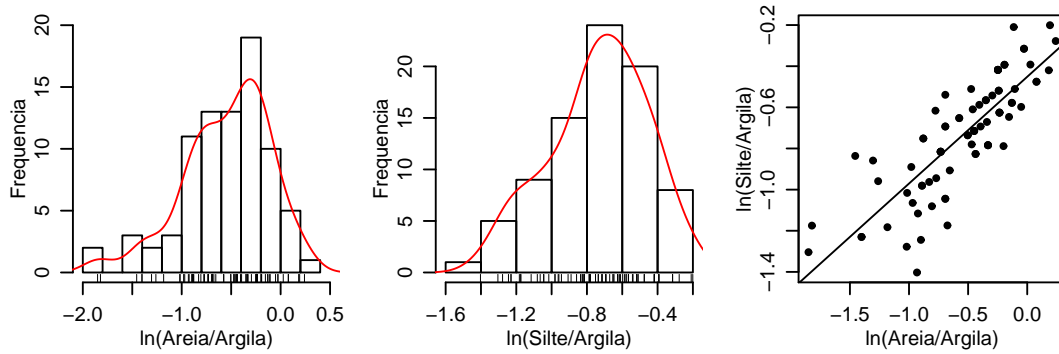


Figura 4.6: Distribuição e diagrama de dispersão das log razões.

A matriz das covariâncias estimadas das log razões, matriz variação, é dada por:

$$Var(ALR) = \begin{bmatrix} 0 & 0,0689 & 0,1916 \\ 0,0689 & 0 & 0,0762 \\ 0,1916 & 0,0762 & 0 \end{bmatrix}.$$

Observa-se que a maior variabilidade ocorreu para $\ln(\text{Areia}/\text{Argila})$, enquanto $\ln(\text{Areia}/\text{Silte})$ e $\ln(\text{Silte}/\text{Argila})$ tem variabilidades menores e similares. A variabilidade composicional, dada pela Equação (2.36), é igual a:

$$\begin{aligned} \text{Totvar}(\vec{X}) &= \frac{1}{B} \sum_{i < j} \text{Var} \left(\ln \left(\frac{X_i}{X_j} \right) \right) \\ &= \frac{1}{B} \left[\sum_{\substack{i=1 \\ j=2,3}} \text{Var} \left(\ln \left(\frac{X_i}{X_j} \right) \right) + \sum_{\substack{i=2 \\ j=3}} \text{Var} \left(\ln \left(\frac{X_i}{X_j} \right) \right) \right] \\ &= \frac{1}{3} \left[\text{Var} \left(\ln \left(\frac{X_1}{X_2} \right) \right) + \text{Var} \left(\ln \left(\frac{X_1}{X_3} \right) \right) + \text{Var} \left(\ln \left(\frac{X_2}{X_3} \right) \right) \right] \\ &= \frac{1}{3} (0,0689 + 0,1916 + 0,0762) \\ &= 0,1123. \end{aligned}$$

4.2 Resultados da Modelagem e Predição Geoestatística

Dentre os métodos de otimização testados, adotou-se o método “L-BFGS-B” por não apresentar problemas de convergência. As estimativas obtidas para os parâmetros do modelo proposto são dados na Tabela 4.2.

Tabela 4.2: Estimativas, erros-padrão e limites inferior e superior dos intervalos de confiança para os parâmetros do modelo, obtidos pelo método Delta via, método de otimização “L-BFGS-B”.

Parâmetros	Estimativas	Erro Padrão	LI. Delta	LS. Delta
μ_1	-0,756	0,432	-1,188	-0,324
μ_2	-0,787	0,221	-1,008	-0,566
σ_1	0,443	0,118	0,325	0,561
σ_2	0,110	0,048	0,062	0,158
τ_1	0,279	0,040	0,239	0,320
τ_2	0,262	0,029	0,233	0,290
ϕ	66,909	38,329	28,579	105,238
ρ	0,966	0,062	0,904	1,000

O valor da log-verossimilhança concentrada foi de $-41,5141$, calculado considerando-se as estimativas obtidas através da amostra com ρ estimado pelo valor amostral do coeficiente de correlação de Pearson que é $r = 0,82$ e ϕ estimado por $\min + 0.2(\max - \min)$, onde “min” e “max” são, respectivamente, a menor e maior distância entre duas localizações. O valor da log-verossimilhança otimizada foi $4,9982$.

A Figura 4.7 apresenta os mapas de predições para os três componentes, considerando-se a volta dos valores preditos em \mathbb{R}^2 para o simplex \mathbb{S}^3 por aproximação de Gauss-Hermite (gráficos (A), (B) e (C)) e por simulação (gráficos (D), (E) e (F)). Foram realizadas 1000 simulações. No caso da aproximação de Gauss-Hermite, a ordem de quadratura foi igual a 7. Em ambos os casos a grade foi constituída de 2500 pontos.

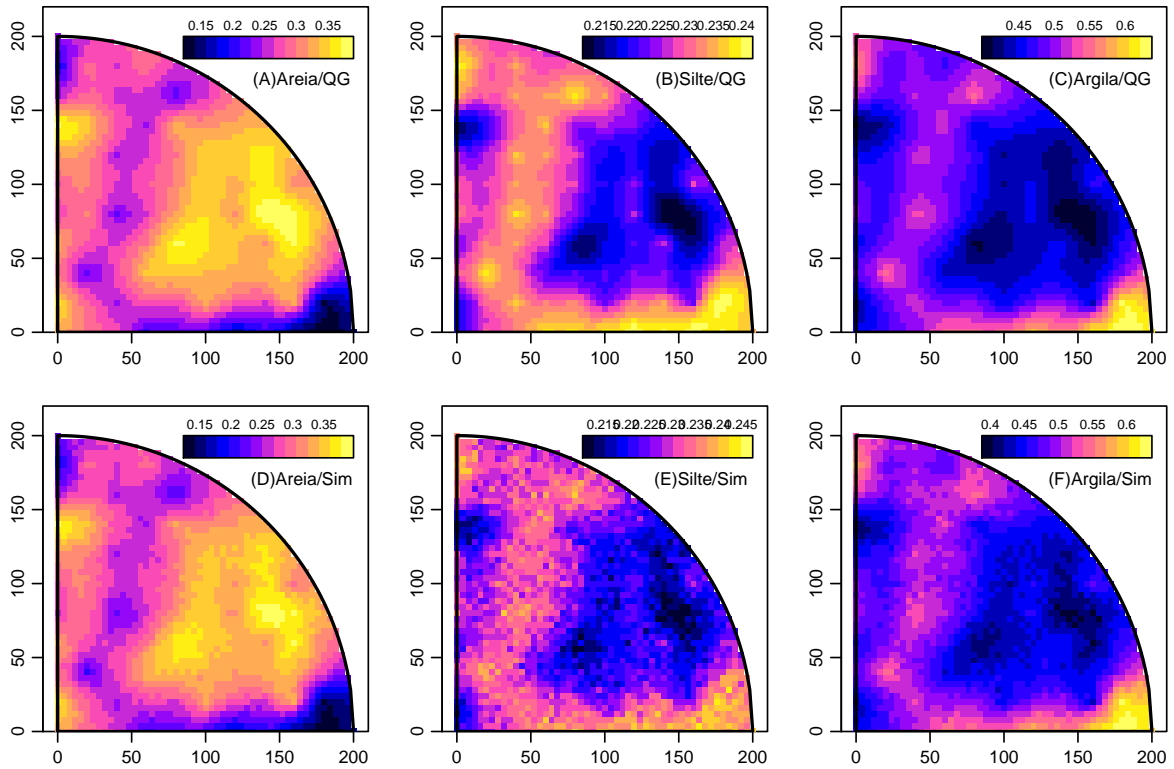


Figura 4.7: Mapas das predições das percentagens de areia, silte e argila obtidas por quadratura de Gauss-Hermite (A-C) e por simulação (D-F).

Diante dos resultados mostrados na Figura 4.7, conclui-se que os mapas pelos dois métodos são muito similares e confirmam os resultados já observados nas Figuras 4.4 e 4.5. Os componentes areia e argila se complementam na área de estudo enquanto areia e silte são concorrentes. Uma comparação dos métodos mostra que os resultados obtidos para areia e argila são os mais próximos (Figura 4.10).

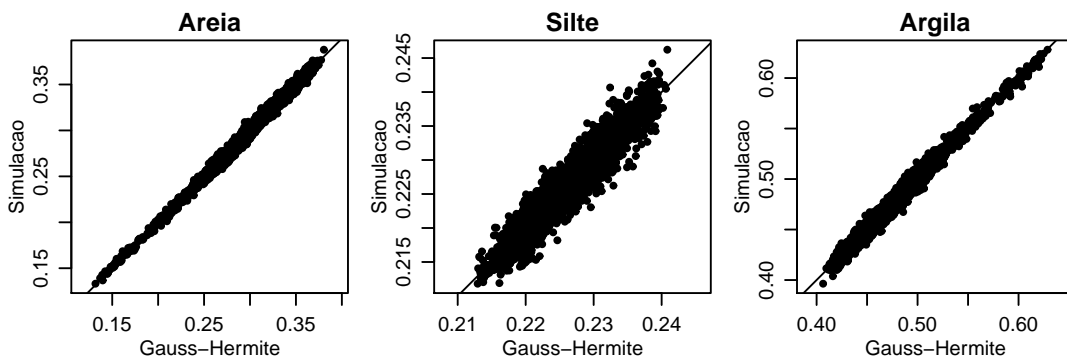


Figura 4.8: Comparação da frações de areia, silte e argila obtidos por quadratura de Gauss-Hermite e simulação.

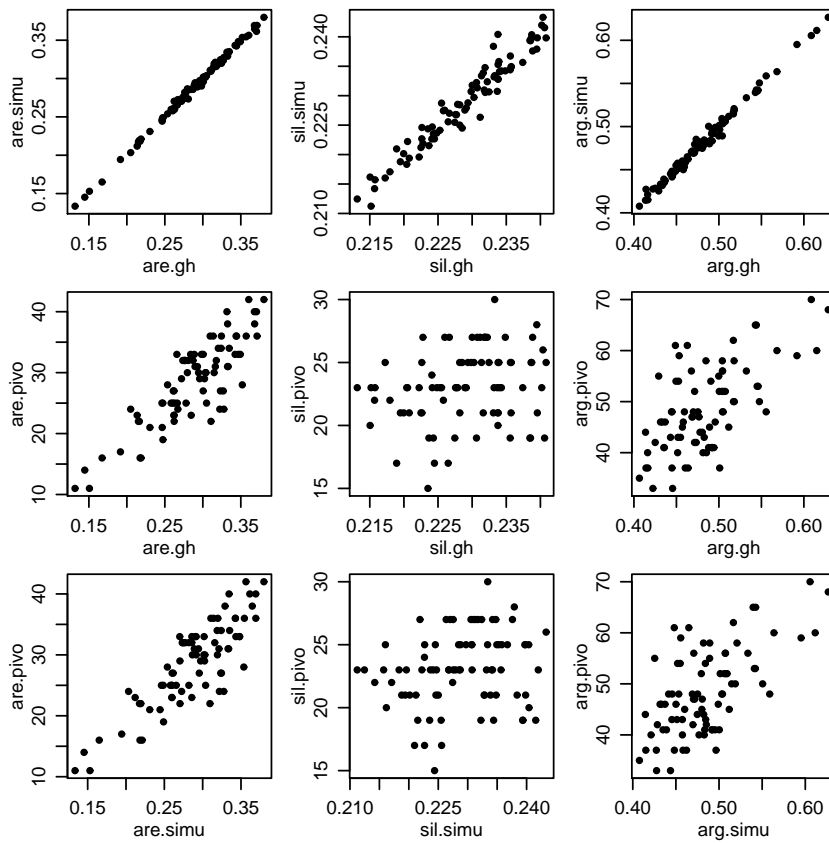


Figura 4.9: Comparação dos dados do pivô com os valores preditos por quadratura de Gauss-Hermite e por simulação.

Na Figura 4.10, embora os valores preditos de silte utilizando quadratura de Gauss-Hermite, assim como os obtidos por simulação, não pareçam correlacionados com os dados originais de silte, nota-se que a faixa dos valores preditos é pequena enquanto a faixa dos valores originais é maior.

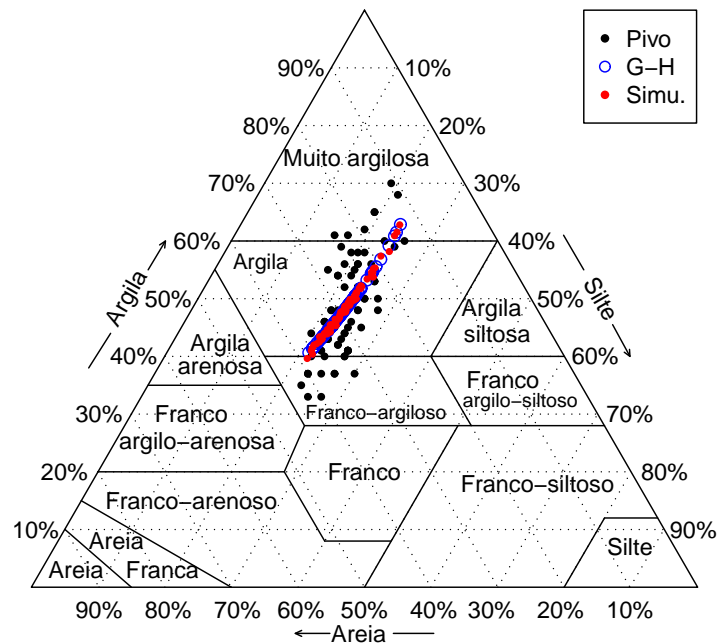


Figura 4.10: Comparação da frações de areia, silte e argila obtidos por quadratura de Gauss-Hermite e simulação.

A Figura 4.10 mostra que os resultados obtidos pelos métodos de predição apresentaram configuração como uma linha reta enquanto os dados originais mostram a variabilidade maior do argila.

A Figura 4.11 mostra a distribuição dos percentuais máximos e mínimos em 1000 simulações. Pode-se notar que a maior densidade dos valores máximos em 1000 simulações, aproximadamente 7,5%, ocorreram para o componente areia na faixa de 70 a 75% correspondendo a 37,5%. Por outro lado, as maiores densidades dos valores mínimos ocorreram para os componentes areia e silte nas faixas de 2 a 2,5% e 5,5 a 6%, respectivamente. Os mapas das frações máximas e mínimas em 2500 localizações são apresentados na Figura 4.12. Por exemplo, observa-se que os valores mínimos, em torno de 45%, na distribuição dos máximos de areia, ocorreram no canto direito inferior do mapa (A), onde se encontra também, os valores máximos na distribuição dos mínimos de argila (mapa (F)).

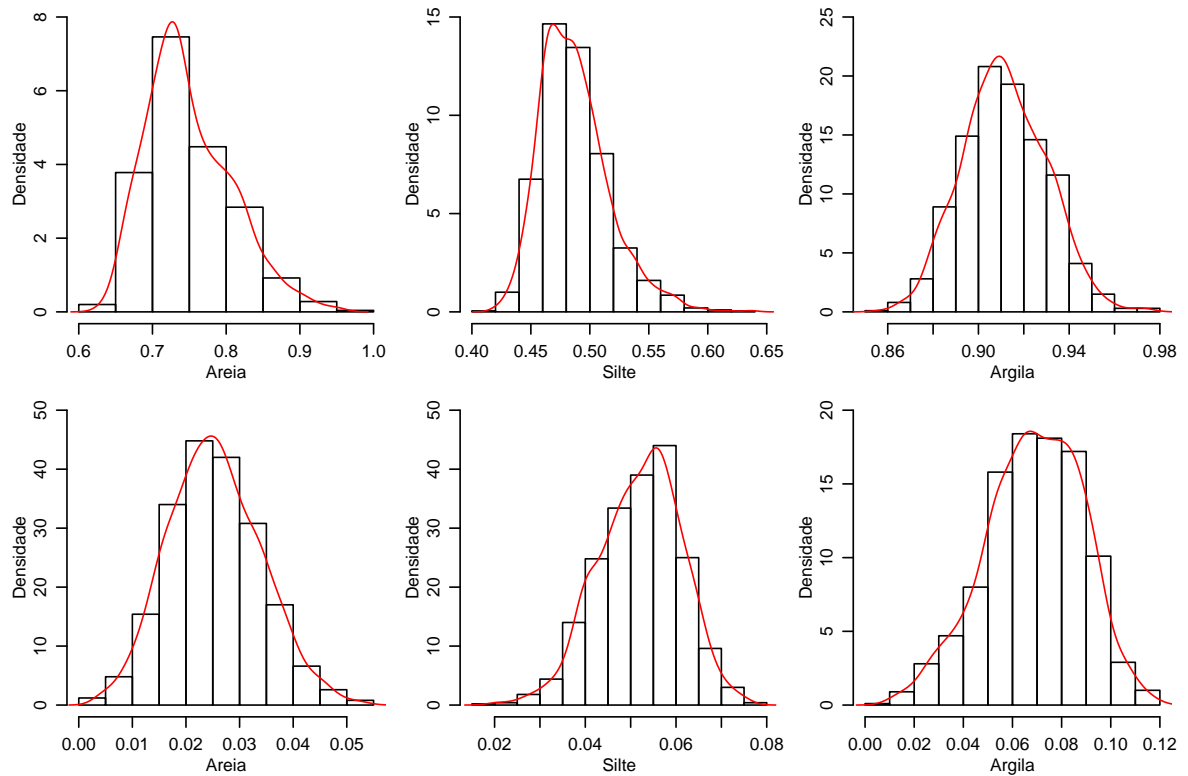


Figura 4.11: Distribuição das frações máximas (1^a linha) e mínimas (2^a linha) na área, por componente.

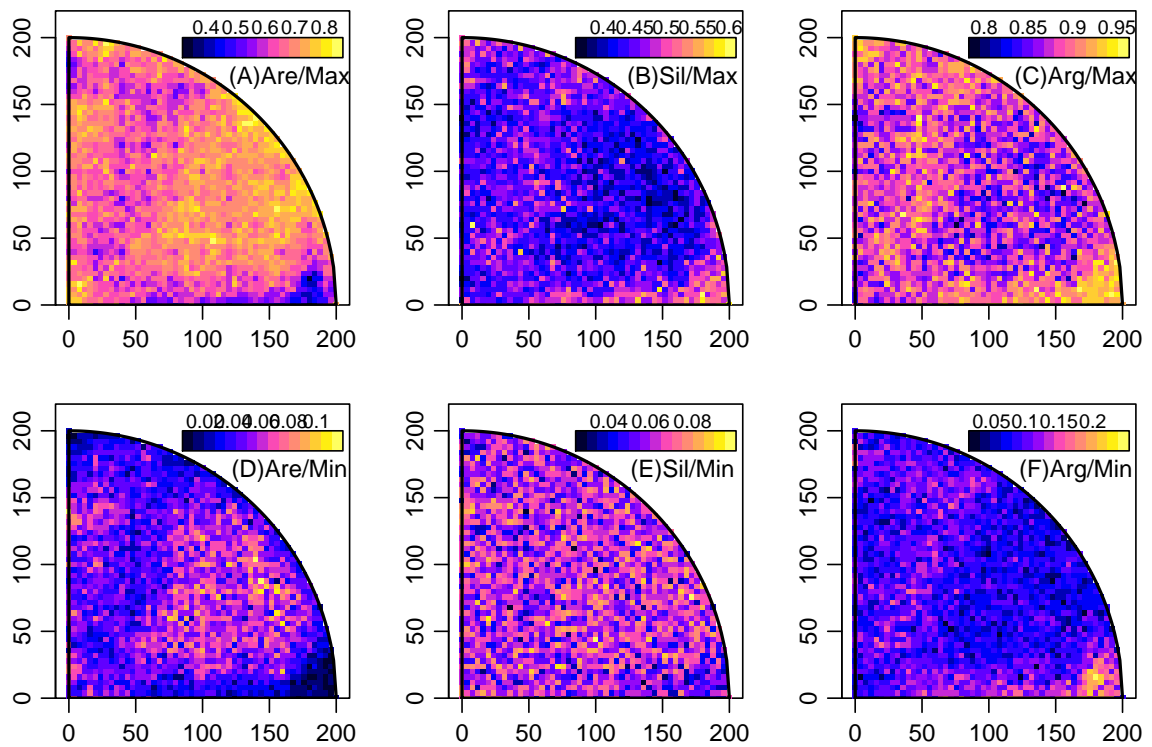


Figura 4.12: Frações máximas e mínimas por componente, obtidas em 1000 simulações para cada localização.

A Tabela 4.3 apresenta as médias das frações máximas e mínimas encontradas em cada localização.

Tabela 4.3: Médias das frações máximas e mínimas de areia, silte e argila obtidas nas 1000 simulações de Gauss-Hermite.

Componente	Máximo	Mínimo
Areia	0,699	0,043
Silte	0,456	0,060
Argila	0,887	0,088

Na Figura 4.13, pode-se afirmar que 50% das frações simuladas de areia variaram, aproximadamente, no intervalo de 15 a 35% e se distribuem na área conforme o mapa (A), enquanto mais especificamente 5% das frações deste mesmo componente variando de 5 a 20% se distribuem conforme o mapa (D). Leituras semelhantes podem ser feitas para os outros mapas.

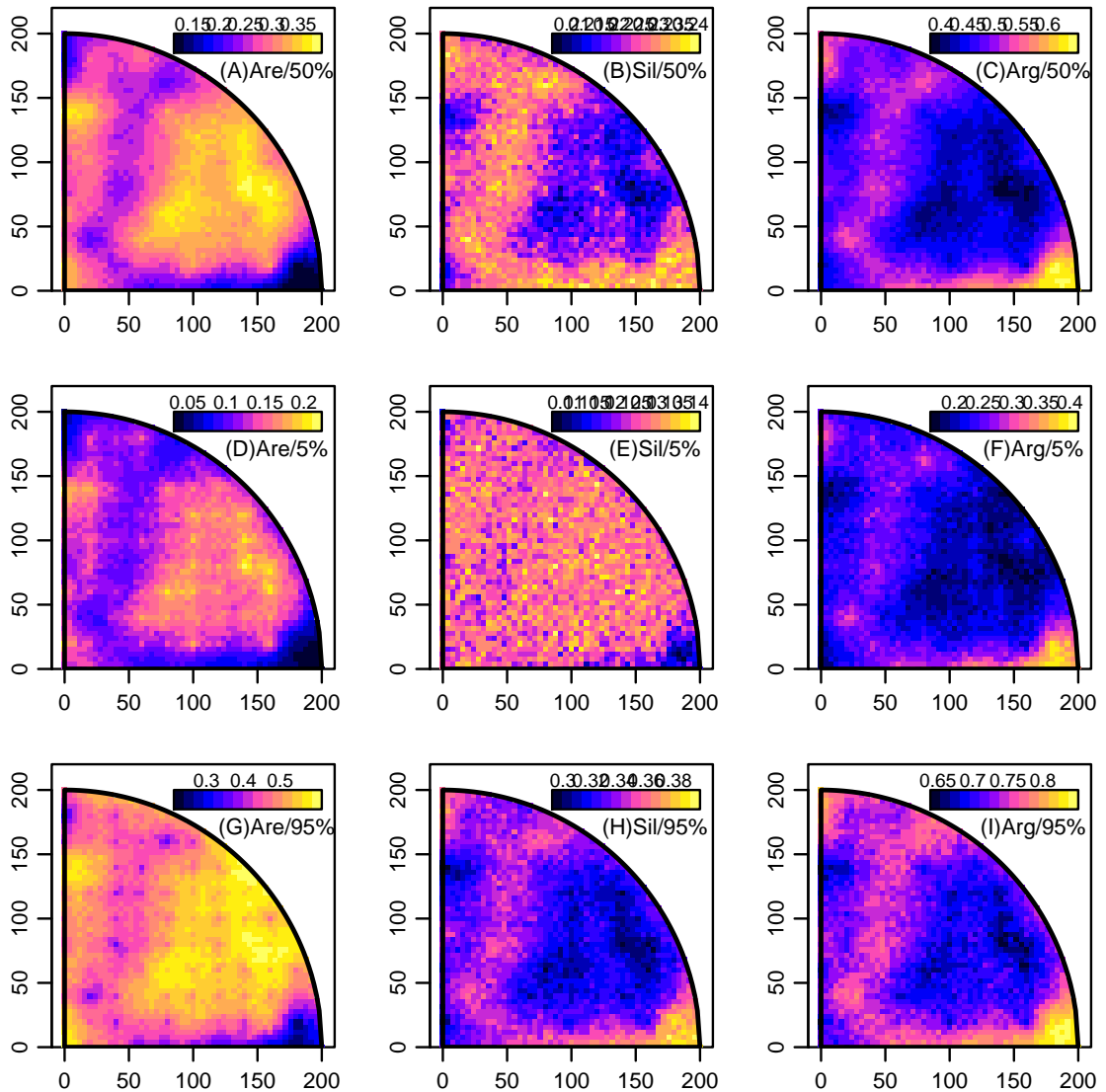


Figura 4.13: Frações medianas, inferiores a 5% e 95% de probabilidade, por componente, obtidos em 1000 simulações para cada localização.

Além dos funcionais apresentados nas Figuras 4.12 e 4.13, a Figura 4.14 mostra que aproximadamente 26% das 1000 simulações apresentaram percentual em torno de 25% na classe “Muito argilosa” e aproximadamente 28% das simulações apresentaram percentual entre 43 e 44%.

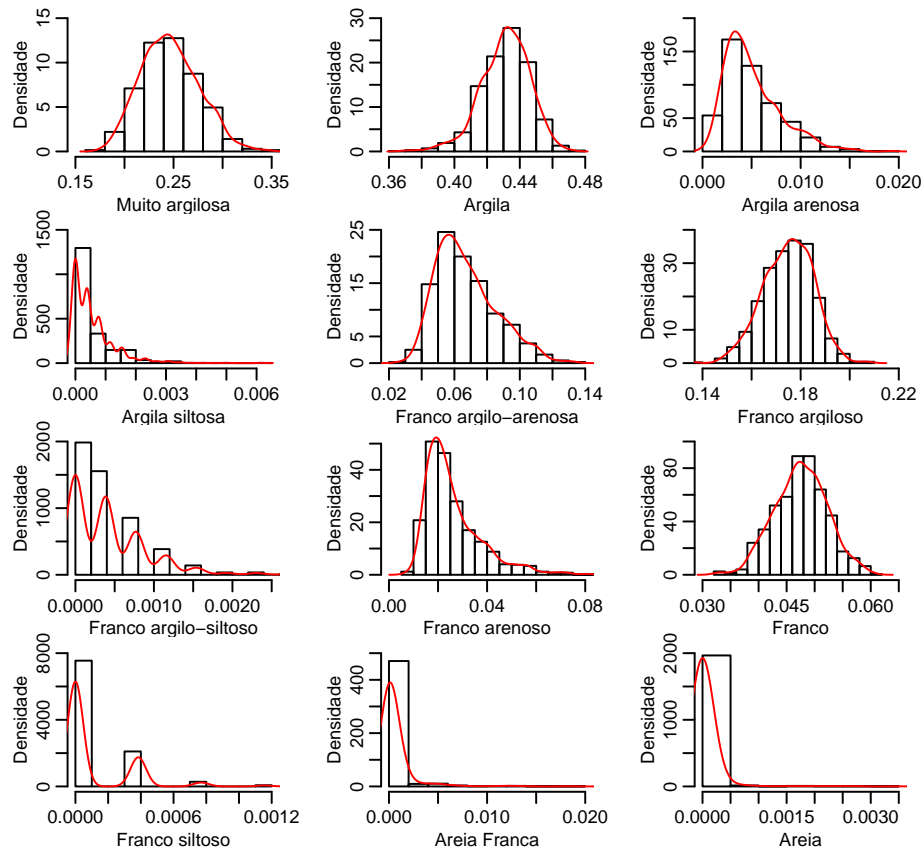


Figura 4.14: Probabilidades *a posteriori* de classificação nas diferentes classes de solo.

Os estudos de simulação da Subseção 3.2 mostraram que, na avaliação da incerteza sobre os parâmetros do modelo, os intervalos de confiança apresentaram baixa cobertura e uma alternativa é a construção de intervalos baseados na verossimilhança perfilhada. Para exemplificar, os gráficos da log-verossimilhança perfilhada para os parâmetros de alcance ϕ e de correlação ρ são apresentados na Figura 4.15. Os intervalos de confiança são obtidos a partir da *deviance* conforme Subseção 2.1.3 Equação (2.18):

$$IC_{\phi} = \{\phi; D(\phi) - D(\hat{\phi})\} < \chi_{1;0,566}^2 \text{ ou } 37,19 < \phi < 178,90 \quad (4.1a)$$

$$IC_{\phi} = \{\phi; D(\phi) - D(\hat{\phi})\} < \chi_{1;0,835}^2 \text{ ou } 26,57 < \phi < 522,53$$

$$IC_{\rho} = \{\rho; D(\rho) - D(\hat{\rho})\} < \chi_{1;0,95}^2 \text{ ou } 0,84 < \rho < 0,98. \quad (4.1b)$$

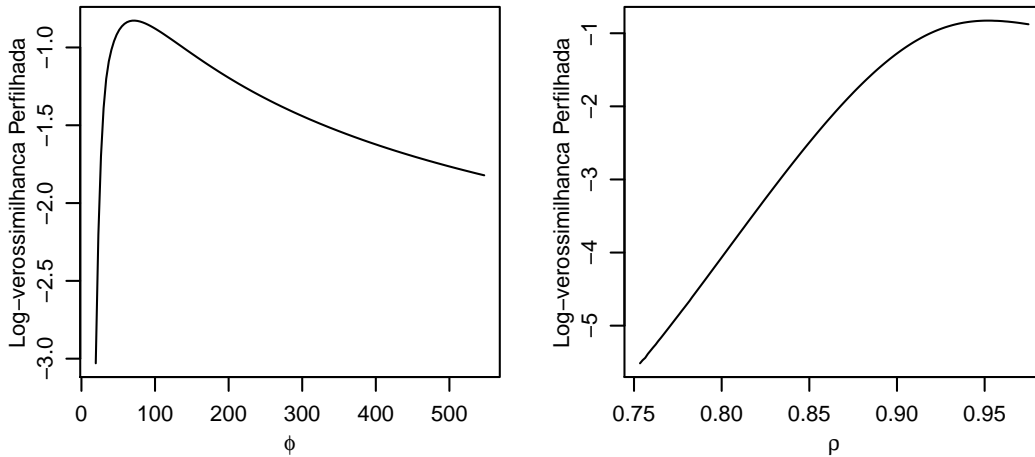


Figura 4.15: Log-verossimilhanças perfilhadas para ϕ e ρ .

Os gráficos da *deviance* para ϕ e ρ são apresentados na Figura 4.16. Primeiramente, observa-se que o nível de confiança máximo para a obtenção de um intervalo para ϕ é de 83,5%, mas com limite superior (522,53m) que ultrapassa o limite da área em estudo. Isso significa que o modelo não foi bem identificado, pois a informação contida nos dados não foi suficiente levando a resultados possivelmente instáveis. Para se obter um intervalo cujos limites pertençam à área, isso é, para se fazer inferências razoáveis para ϕ com as informações disponíveis, é necessário escolher um nível de confiança de 56,6%, cujos limites são dados na Equação (4.1(a)).

Com relação a ρ , o espaço paramétrico impõe um truncamento superior no intervalo. O limite inferior desse intervalo é 95% menos plausível do que a estimativa de máxima verossimilhança, $\hat{\rho} = 0,95$, mas não é possível ter a mesma plausibilidade para o limite superior.

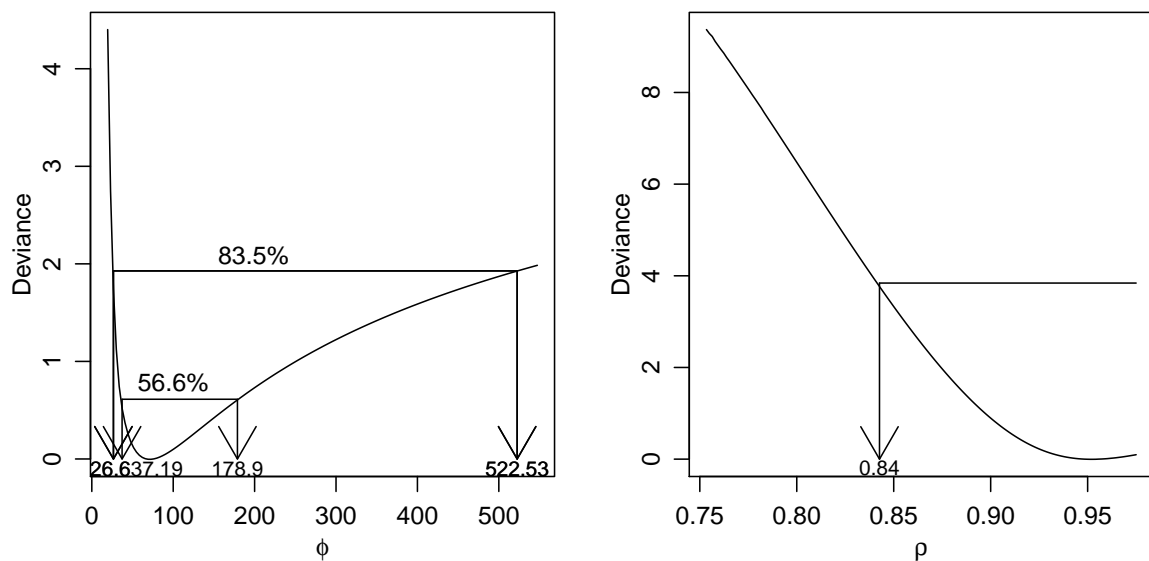


Figura 4.16: Deviance para ϕ , com linhas de corte de 83,5% e 56,6% de confiança (esquerda) e para ρ , com linhas de corte de 95% de confiança (direita).

4.3 Resultados da Análise Bayesiana

Os resultados da inferência via métodos bayesianos foram obtidos utilizando-se o algoritmo Metropolis-Hastings. Foram executadas 12000 simulações, com período de aquecimento de 1000 simulações e um salto igual a 10. A sintonia (*tune*) foi igual a 0,5 para todos os parâmetros resultando em cadeias formadas por 1200 simulações.

As médias dos valores da distribuição *a posteriori* dos parâmetros bem como os respectivos intervalos de credibilidade dos parâmetros são apresentados na Tabela 4.4.

Tabela 4.4: Média dos valores da distribuição *a posteriori* de $\vec{\theta}$ e intervalos de 95% de credibilidade.

Parâmetros	Média	LI. Delta	LS. Delta
μ_1	-0,749	-1,164	-0,330
μ_2	-0,785	-0,916	-0,666
σ_1	0,432	0,328	0,549
σ_2	0,109	0,058	0,182
τ_1	0,284	0,224	0,354
τ_2	0,260	0,225	0,302
ϕ	66,392	51,984	83,714
ρ	0,939	0,845	0,995

Ao comparar as estimativas na Tabela 4.4 com os valores apresentados na Tabela 4.2, observa-se que são aproximadamente iguais com o intervalo de credibilidade para ϕ mais curto.

Além disso, a comparação dos resultados obtidos por inferência bayesiana e não bayesiana é apresentada nos gráficos da Figura 4.17, sendo possível verificar que as densidades dos três componentes são semelhantes.

Os mapas dos valores preditos de areia, silte e argila são apresentados na Figura 4.18, cujo padrão geral assemelha-se aos obtidos por inferência clássica (Figura 4.7).

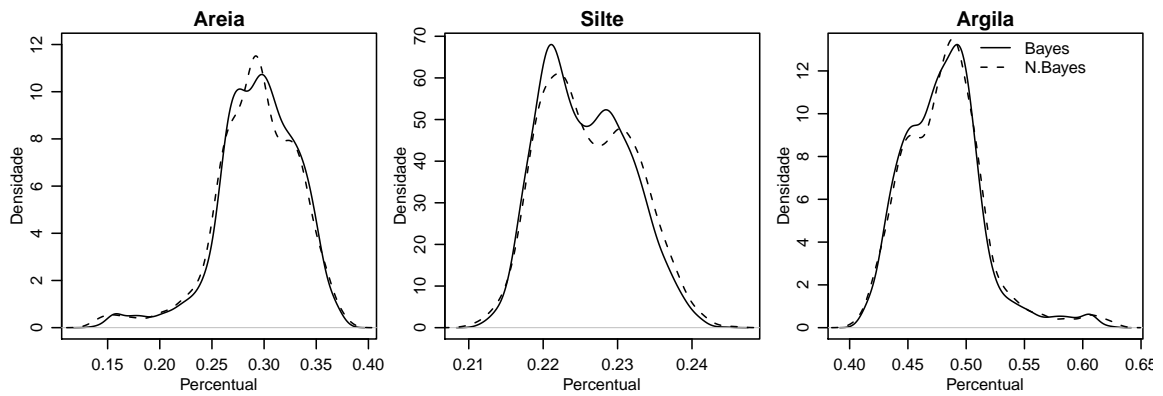


Figura 4.17: Comparação das densidades de areia (à esquerda), silte (centro) e argila (à direita) obtidos por inferência bayesiana e não-bayesiana.

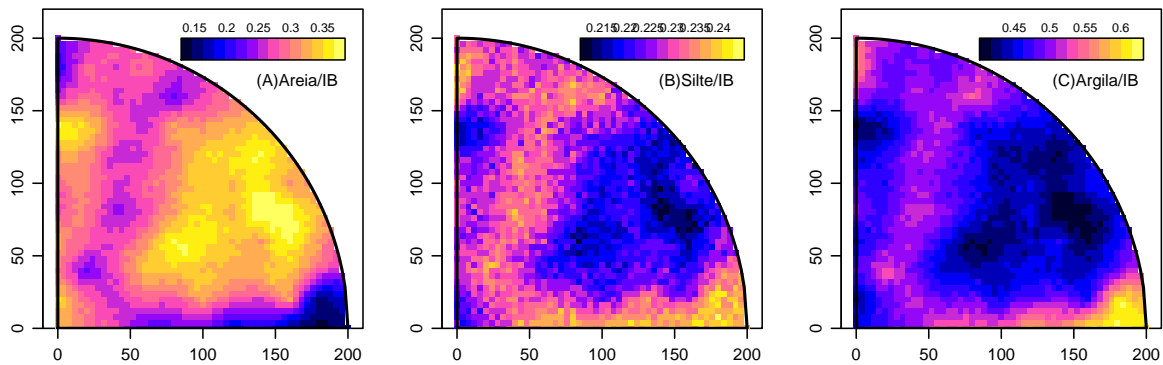


Figura 4.18: Valores preditos de areia (à esquerda), silte (centro) e argila (à direita) por inferência bayesiana.

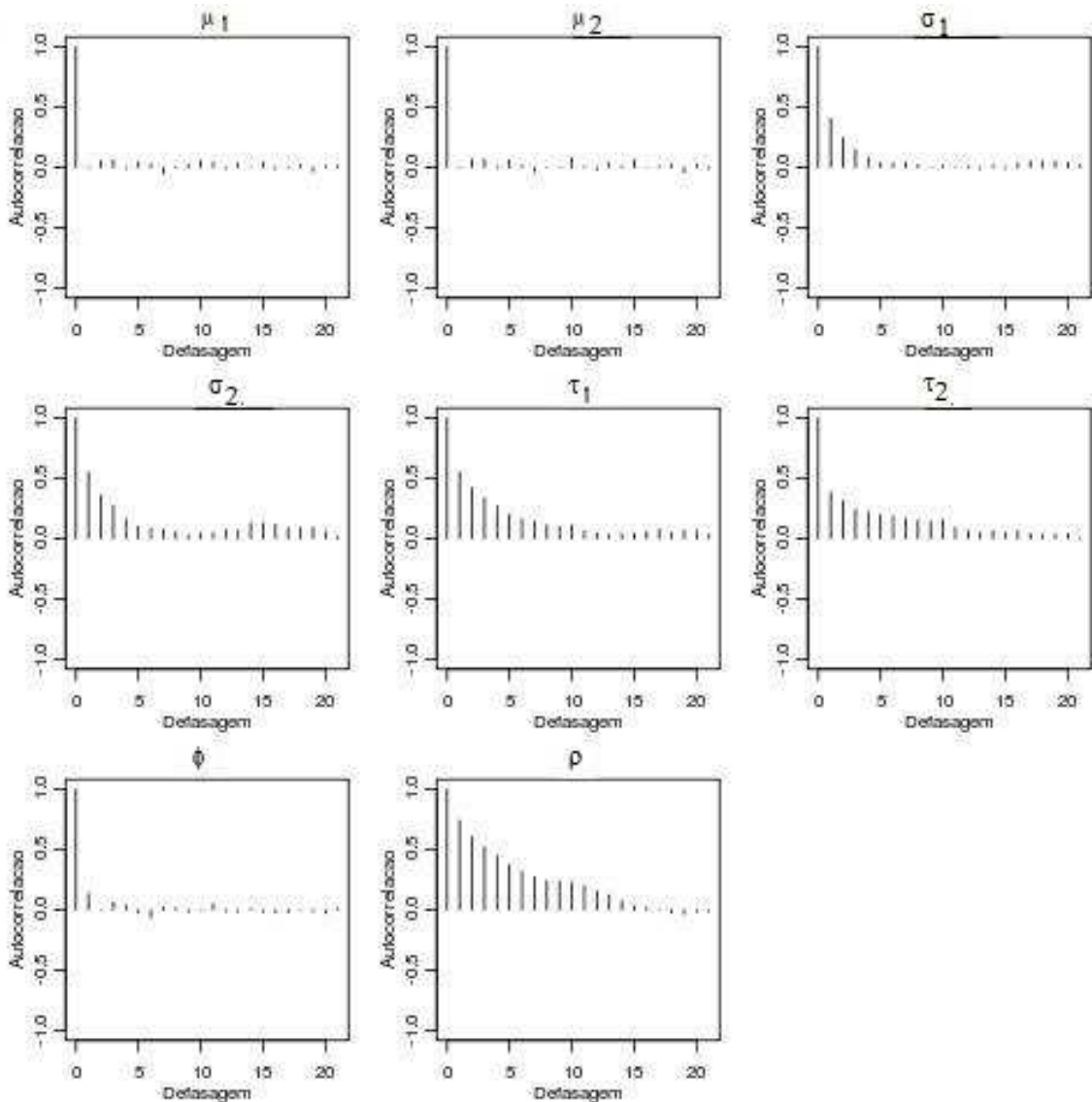
4.3.1 Avaliação da Convergência

Para avaliar a validade dos resultados obtidos por inferência bayesiana apresenta-se, a seguir, os resultados dos diagnósticos de convergência das cadeias simuladas.

A Tabela 4.5 apresenta as autocorrelações *a posteriori*, dos valores dos parâmetros nas cadeias, para diferentes escolhas de defasagens (saltos) entre valores amostrados ao longo da cadeia. Observa-se que até a defasagem 50 os valores apresentam uma redução, característica típica de convergência.

Tabela 4.5: Autocorrelações das *posteriori* dos parâmetros.

Def.	μ_1	μ_2	σ_1	σ_2	τ_1	τ_2	ϕ	ρ
0	1	1	1	1	1	1	1	1
1	0,003	-0,002	0,407	0,554	0,553	0,383	0,133	0,741
5	0,047	0,059	0,040	0,102	0,201	0,197	-0,026	0,374
10	0,057	0,082	0,015	0,046	0,112	0,156	-0,002	0,229
50	-0,033	-0,027	-0,018	-0,033	-0,035	-0,049	-0,017	-0,055

Figura 4.19: Autocorrelações dos parâmetros *a posteriori*.

Quanto aos gráficos da Figura 4.19, as autocorrelações decrescem com o aumento da defasagem e a maior dependência é observada entre os valores da *posteriori* de ρ . As correlações entre os parâmetros da cadeia, por sua vez, são mostradas na Tabela 4.6.

Tabela 4.6: Correlações das *posteriori* dos parâmetros.

<i>Posteriori</i>	μ_1	μ_2	σ_1	σ_2	τ_1	τ_2	ϕ	ρ
μ_1	1							
μ_2	0,907	1						
σ_1	-0,085	-0,126	1					
σ_2	-0,057	-0,211	0,709	1				
τ_1	0,051	0,095	-0,232	-0,265	1			
τ_2	0,020	0,037	0,103	-0,009	0,656	1		
ϕ	-0,059	-0,045	0,205	0,058	0,107	0,031	1	
ρ	-0,060	0,006	0,273	-0,140	0,021	0,368	-0,072	1

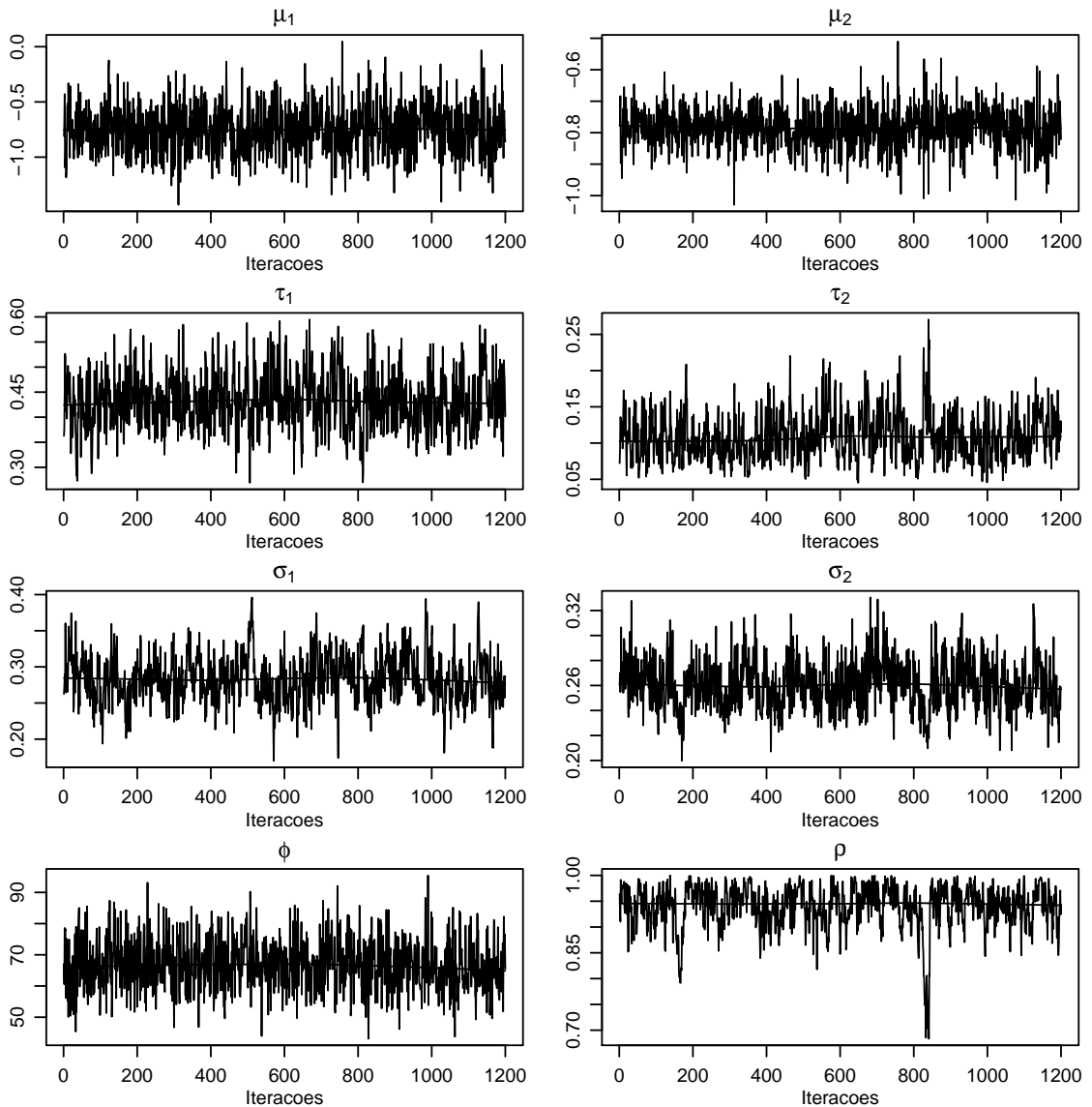
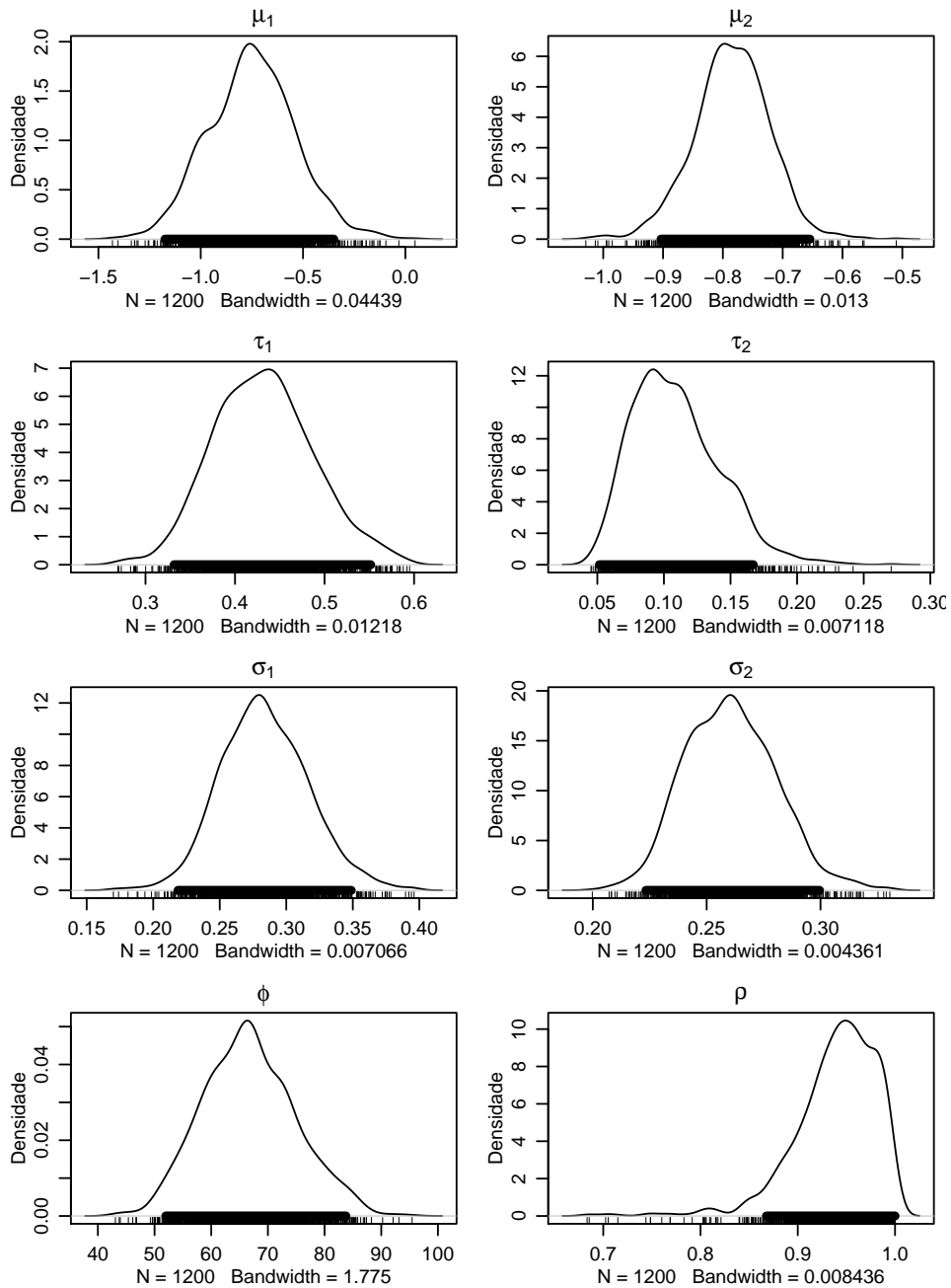
Figura 4.20: Trajetória das *posteriori* dos parâmetros.

Tabela 4.7: Limites dos intervalos de 95% de credibilidade de alta densidade *a posteriori*.

<i>Posteriori</i>	μ_1	μ_2	σ_1	σ_2	τ_1	τ_2	ϕ	ρ
LI	-1,174	-0,903	0,332	0,052	0,219	0,223	51,987	0,868
LS	-0,350	-0,655	0,552	0,167	0,349	0,300	83,757	0,999

Figura 4.21: Densidades estimadas para *as posteriori* dos parâmetros com respectivos intervalos de alta densidade *a posteriori* a 95%.

Pode-se observar, na Figura 4.20, as trajetórias das cadeias nas 1200 simulações e os gráficos da Figura 4.21 mostram as densidades estimadas de cada parâmetro nos quais a multimodalidade seria um indicativo de não convergência, o que não ocorre nesses casos.

Um diagnóstico de convergência adicional é comparar os intervalos de mais alta densidade *a posteriori*, Tabela 4.7, que são construídos com todos os valores da cadeia e o centro da distribuição *a posteriori*. Os intervalos estão representados por segmentos de reta acima dos *rug plot*. Grandes discrepâncias são indicativos que os últimos valores não são típicos da amostra completa e, novamente, essas não são observadas nesses gráficos.

Pode-se observar que todos os valores apresentados na Tabela 4.8 são inferiores a 2 o que implica na não preocupação com a convergência das cadeias.

Tabela 4.8: Valores de z_G do teste de Geweke para *as posteriori* dos parâmetros.

<i>Posteriori</i>	μ_1	μ_2	σ_1	σ_2	τ_1	τ_2	ϕ	ρ
z_G	-0,299	0,427	-0,217	-0,890	-0,067	-0,278	0,723	0,053

Tabela 4.9: Fatores de redução de escala potencial do teste de Gelman e Rubin.

<i>Posteriori</i>	μ_1	μ_2	σ_1	σ_2	τ_1	τ_2	ϕ	ρ
Est.Pontual	1,83	1,49	11,17	3,15	9,70	15,17	14,14	6,20
Quantil 95%	7,02	3,01	56,83	11,70	36,38	50,88	87,24	15,83

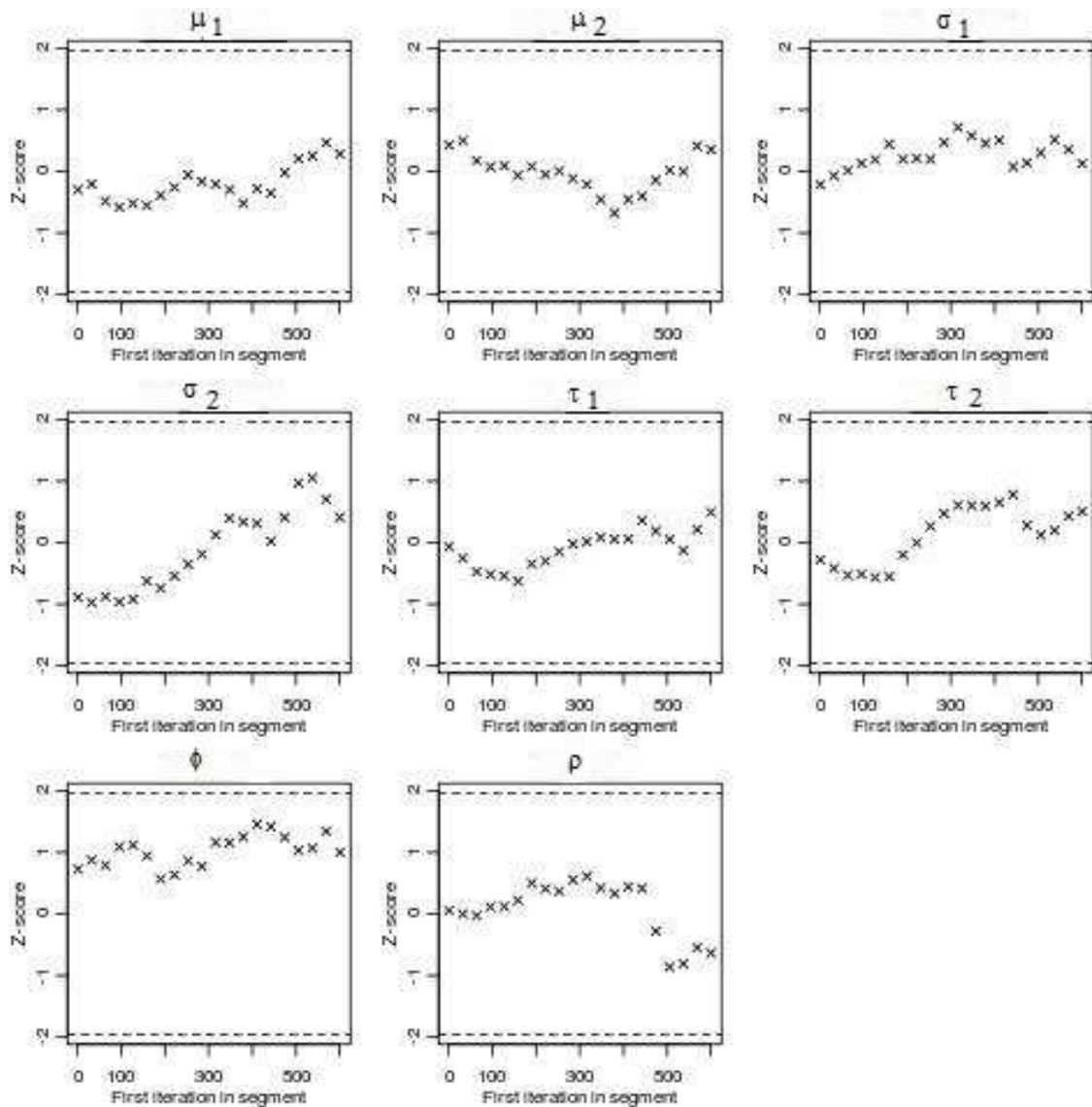


Figura 4.22: Diagnóstico de convergência para o Teste de Geweke.

A Tabela 4.9 apresenta as estimativas pontuais e quantis de 95% de credibilidade para os fatores de redução de escala potencial do diagnóstico de convergência de seqüências múltiplas de Gelman e Rubin. Como os resultados são superiores a 1,2, é recomendável aumentar o número de simulações, bem como o salto, e com isso pode-se obter resultados melhores. Os resultados estão representados graficamente na Figura 4.23.

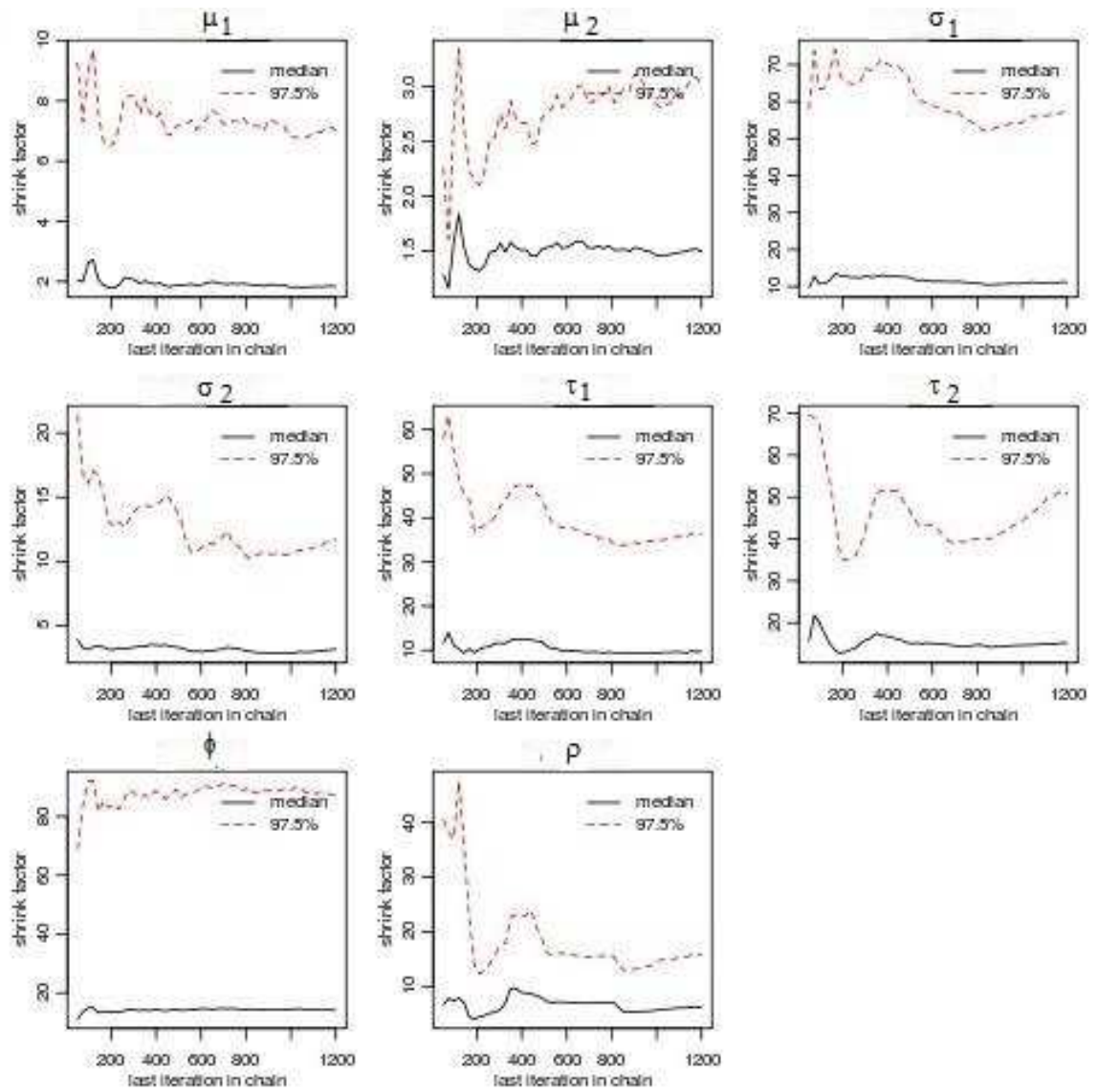


Figura 4.23: Diagnóstico de convergência para o Teste de Gelman e Rubin.

5 CONCLUSÕES E SUGESTÕES DE TRABALHOS FUTUROS

Os estudos de simulação mostraram que o modelo proposto neste trabalho é capaz de capturar as variações espaciais, sejam induzidas pelas composições ou não estruturadas. Os procedimentos adotados permitiram a construção de mapas de areia, silte e argila por uma metodologia que implicitamente garante a restrição de que as frações somem 1, não só nos pontos observados como nos pontos preditos.

A metodologia desenvolvida foi baseada na teoria de verossimilhança e a declaração explícita do modelo permitiu que fossem feitas inferências sobre parâmetros de forma usual. Apresentou-se, também, o ajuste do modelo via inferência bayesiana, a fim de se considerar, nas predições, a incerteza associada à estimação dos parâmetros do modelo.

A transformação do vetor de médias e da matriz de covariância para o espaço amostral simplex \mathbb{S}^3 por quadratura de Gauss-Hermite de ordem igual a 7 não diferiram dos obtidos com ordem 20 e isso confirma o que a literatura cita sobre o assunto.

Os mapas de valores preditos de areia, silte e argila construídos de forma usual, com a volta para o simplex tanto por quadratura de Gauss-Hermite como por simulação, e por inferência bayesiana apresentaram padrão geral semelhante, com diferenças nos detalhes.

Com os recursos computacionais disponíveis no momento da elaboração do trabalho, os tempos computacionais requeridos para a elaboração dos mapas de predição foram, em média, de 480 horas para os dados reais quando utilizou-se da quadratura de Gauss-Hermite, 168 horas considerando a volta dos valores preditos para o simplex por simulação e 168 horas por inferência bayesiana.

Como recomendação para trabalhos futuros que tratem do assunto em questão sugere-se quatro linhas de atuação:

1. extensões do presente trabalho

A princípio poderia-se pensar na incorporação de covariáveis no modelo. Pode-se pensar em covariável como sendo a ocorrência absoluta; a razão desta; a razão entre duas covariáveis; a covariável afetando um componente ou a log-razão. Ainda, a covariável poderia ser considerada como uma outra classe de solo como pedregulhos, como a cota ou altitude do terreno e até mesmo a própria coordenada amostral. Estas são extensões conceitualmente não complicadas mas interessantes de se investigar. Por outro lado, ainda em relação à covariáveis, é possível situações em que se deseja modelar as composições mais alguma(s) variável(veis), ou seja, um modelo multivariado em que parte da resposta seja composicional e a outra não. Neste caso surge a necessidade de se investigar formas de fazer a transformação das predições do espaço real para o espaço amostral simplex.

Na direção do artigo de Lark e Bishop (2007), existe espaço na literatura para estudos sobre avaliação da qualidade do modelo geoestatístico composicional, diagnósticos de convergência das cadeias de Monte Carlo, identificação de situações em que a aproximação normal é adequada.

Além dos tópicos citados acima, existem situações em que as amostras podem apresentar proporções muito baixas dos componentes ou até mesmo, zeros. Assim, estas também são áreas de possíveis investigações.

Quanto a procedimentos computacionais, citam-se a implementação de códigos mais gerais, eficientes, com simulações mais rápidas.

2. extensões para outros modelos

Existe espaço na literatura de dados composicionais para outras formas de especificação do modelo multivariado como por exemplo o apresentado em Schmidt e Sansó (2006). Este modelo sob forma triangular implica na investigação de como induzir correlação negativa que pode ser nas variáveis originais ou nas transformadas.

Em relação ao efeito pepita, no modelo proposto este efeito está incorporado ao efeito dos dados composicionais e nos modelos mais gerais teria que ser investigado.

Neste item, inclui-se ainda a generalização do modelo geoestatístico composicional para mais de três componentes e na implementação computacional, formas de interface com outros pacotes estatísticos como *spBayes*(FINLEY; BANERJEE; CARLIN, 2007).

3. aplicações em outras áreas

Potenciais aplicações ocorrem, por exemplo, quando se trata de poluentes ou proporções de gases que compõe o efeito estufa sendo de interesse o mapeamento destes gases em diferentes regiões de uma cidade, país. Pode-se pensar, também, em proporções ao estudar a diversidade e riqueza de espécies em que as proporções representariam o número de cada espécie em relação ao total analisado. Observa-se que são temas com caracterís-

ticas particulares, que induzem proporções ou subcomposições, de extrema relevância no momento atual, e que encontram espaço na literatura para estudos aprofundados.

4. extensões gerais de estatística espacial

Neste sentido, é muito comum incluir a não estacionariedade no primeiro momento, mas pode-se incluir uma não estacionariedade no segundo momento, na covariância e o estudo de modelos não estacionários torna-se uma possibilidade. Além disso, é viável o estudo de modelos composicionais espaço-temporais em que as proporções podem se modificar ao longo do tempo.

Referências Bibliográficas

- ABRAMOWITZ, M.; STEGUN, I. A. **Handbook of mathematical functions with formulas, graphs, and mathematical tables**. Washington: Milton Abramowitz and Irene A. Stegun, 1972, p. 924.
- AITCHISON, J. The statistical analysis of compositional data. **Royal Statistical Society, Series B**, v. 44, p. 139–177, 1982.
- AITCHISON, J. (Ed.). **The statistical analysis of compositional data**. New Jersey: The Blackburn Press, 1986.
- AITCHISON, J. Logratios and natural laws in compositional data analysis. **Mathematical Geology**, v. 31, p. 563–580, 1999. Subject Collection: Earth and Environment Science.
- AITCHISON, J. et al. Logratios analysis and compositional distance. **Mathematical Geology**, v. 32, p. 563–580, 2000.
- AITCHISON, J.; EGOZCUE, J. J. Compositional data analysis: where are we and where should we be heading? **Mathematical Geology**, v. 37, p. 829–850, October 2005. Subject Collection Earth and Environmental Science.
- AITCHISON, J.; GREENACRE, M. Biplot of compositional data. **Journal of the Royal Statistical Society, Series C**, v. 51, p. 375–392, October 2002.
- ANDREWS, D. F.; GNADESIKAN, R.; WARNER, J. L. Methods for assessing multivariate normality. In: **Multivariate Analysis Survey Research Methods Section, ASA**. New York: [s.n.], 1973.
- AZZALINI, A. (Ed.). **Statistical Inference Based on the Likelihood**. London: Chapman and Hall, 1996.
- BAILEY, T. C.; GATRELL, A. C. **Interactive spatial data analysis**. Harlow: Longman, 1995.
- BANERJEE, S.; CARLIN, B. P.; GELFAND, A. E. **Hierarchical modelling and analysis for spatial data**. Boca Raton: Chapman and Hall, 2004.
- BARCELÓ-VIDAL, C.; MARTÍN-FERNÁNDEZ, J. A.; PAWLOWSKY-GLAHN, V. Mathematical foundations of compositional data analysis. 2001. Acesso em 08/06/08. Disponível em: <http://ima.udg.edu/~jamf/carles_martin_vera_cancun.pdf>.

- BOGNOLA, I. A. et al. Modelagem uni e bivariada da variabilidade espacial de rendimento de pinus taeda l. **Floresta**, v. 38, p. 373–385, 2008.
- BOOGAART, K. G. v. d.; TOLOSANA, R.; BREN, M. compositions package. Mar 2009. Acesso em 06/11/09. Disponível em: <<http://cran.r-project.org/web/packages/compositions/>>.
- BOX, G.; COX, D. An analysis of transformation. **Journal of the Royal Statistical Society, Series B**, p. 211–252, 1964.
- BURDEN, R. L.; FAIRES, J. D. **Análise Numérica**. São Paulo: Pioneira Thomson Learning, 2003.
- BUTLER, A.; GLASBEY, C. A latent gaussian model for compositional data with zeros. **Journal of the Royal Statistical Society, Series C, (in press)**, 2008. Disponível em: <http://www.bioss.ac.uk/staff/adam/documents/ButlerGlasbey_resubmit.pdf>.
- BYRD, R. H. et al. A limited memory algorithm for bound constrained optimization. **SIAM J. Scientific Computing**, v. 16, p. 1190–1208, 1995.
- CARLIN, B. P.; LOUIS, T. A. **Bayesian Methods for Data Analysis**. Third. Boca Raton: Chapman and Hall, 2009.
- CHILÈS, J. P.; DELFINER, P. (Ed.). **Geostatistics: modeling spatial uncertainty**. New York: John Wiley & Sons, 1999.
- COX, D. R.; HINKLEY, D. V. **Theoretical Statistics**. London: Chapman & Hall, 1974.
- CRESSIE, N. A. C. (Ed.). **Statistics spatial data**. New York: Wiley, 1993.
- DEGROOT, M. H.; SCHERVISH, M. J. **Probability and Statistics**. Third. Boston: Addison Wesley, 2002.
- DIGGLE, P.; MENEZES, R.; SU, T.-I. Geostatistical inference under preferential sampling. **Journal of the Royal Statistical Society: Series C**, v. 59, 2010.
- DIGGLE, P. J.; RIBEIRO JR., P. J. **Model-based geostatistics**. USA: Springer Series in Statistics, 2007.
- DIGGLE, P. J.; RIBEIRO JR., P. J.; CHRISTENSEN, O. F. An introduction to model-based geostatistics. In: MØLLER, J. (Ed.). **Spatial Statistics and Computational Methods**. New York: Springer, 2003. p. 43–86.
- DIGGLE, P. J.; TAWN, J. A.; MOYEED, R. A. Model-based geostatistics. **Applied Statistics**, v. 47, n. 3, p. 299–350, 1998. Disponível em: <citeseer.ist.psu.edu/diggle98modelbased.html>.

- EGUCHI, E. S.; SILVA, E. L.; OLIVEIRA, M. S. Variabilidade espacial da textura e da densidade de partículas em um solo aluvial no município de lavras, mg. **Revista Brasileira de Engenharia Agrícola e Ambiental**, v. 6, p. 242–246, 2002.
- EHLERS, R. S. *Introdução à Inferência Bayesiana*. Curitiba: [s.n.], 2006. Disponível em: <<http://www.leg.ufpr.br/~ehlers>>.
- EYNATTEN, H. v.; BARCELÓ-VIDAL, C.; PAWLOWSKY-GLAHN, V. Modelling compositional change: the example of chemical weathering of granitoid rocks. **Mathematical Geology**, v. 35, p. 231–251, April 2003. Subject Collection Earth and Environmental Science.
- FALEIROS, A. C.; YONEYAMA, T. (Ed.). **Teoria Matemática de Sistemas**. São Paulo: Editora Arte & Ciência, 2002.
- FINLEY, A. O.; BANERJEE, S.; CARLIN, B. P. spbayes: an r package for univariate and multivariate hierarchical point-referenced spatial models. **Journal of Statistical Software**, v. 19, p. 1–24, April 2007.
- FLETCHER, R.; REEVES, C. M. Function minimization by conjugate gradients. **Computer Journal**, v. 7, p. 148–154, 1964.
- GAMERMAN, D. **Markov chain Monte Carlo: stochastic simulation for Bayesian inference**. 1.ed.. ed. Londres: Chapman & Hall/CRC, 1997.
- GAMERMAN, D.; LOPES, H. F. **Markov chain Monte Carlo: stochastic simulation for Bayesian inference**. 2a. ed. Londres: Chapman & Hall/CRC, 2006.
- GELMAN, A. et al. (Ed.). **Bayesian Data Analysis**. Second. Boca Raton: Chapman and Hall, 2003.
- GILKS, W. R.; RICHARDSON, S.; SPIEGELHALTER, D. J. **Markov Chain Monte Carlo in Practice**. London: Chapman and Hall, 1996.
- GILL, J. **Bayesian Methods For The Social And Behavioral Sciences**. London: Chapman and Hall, 2002.
- GONÇALVES, A. C. A. (Ed.). **Variabilidade espacial de propriedades físicas do solo para fins de manejo da irrigação**. Tese (Doutorado em Agronomia) - Escola Superior de Agricultura “Luiz de Queiroz”. Universidade de São Paulo: Piracicaba, 1997.
- GONÇALVES, A. C. A.; FOLEGATTI, M. V.; MATA, J. D. V. Análise exploratória e geostatística da variabilidade de propriedades físicas de um Argissolo Vermelho. **Acta Scientiarum**, v. 23, p. 1149–1157, June 2001.

- GOOVAERTS, P. **Geostatistics for Natural Resources Evaluation**. New York: Oxford University Press, 1997, Seção 4.2.3.
- GRAF, M. Precision of compositional data in a stratified two-stage cluster sample: comparison of the swiss earnings structure survey 2002 and 2004. In: **Survey Research Methods Section, ASA**. [S.l.: s.n.], 2006. p. 3066–3072.
- HOEL, P. G.; PORT, S. C.; STONE, C. J. (Ed.). **Introduction to Stochastic Processes**. Boston: Houghton Mifflin Company, 1972.
- ISAAKS, E. H.; SRISVASTAVA, R. M. **An introduction to applied geostatistics**. New York: Oxford University Press, 1989.
- JAMES, B. R. (Ed.). **Probabilidade: um curso em nível intermediário**. 3.ed.. ed. IMPA: Rio de Janeiro, 2004.
- JOHNSON, R. A.; WICHERN, D. W. **Applied statistical analysis**. Fourth. USA: Prentice Hall, 1998.
- KITANIDIS, P. **Introduction to geostatistics: applications in hydrogeology**. Third. New York: Cambridge University Press, 1997.
- LABUS, M. Compositional data analysis as a tool for interpretation of rock porosity parameters. **Geological Quarterly**, v. 49, p. 347–354, 2005.
- LAKE, J. V.; BOCK, G. R.; GOODE, J. A. (Ed.). **Precision agriculture: spatial and temporal variability of environmental quality (norvatis foundation symposia)**. USA: John wiley & Sons Ltd, 1997.
- LARK, R. M.; BISHOP, T. F. A. Cokriging particle size fractions of the soil. **European Journal of Soil Science**, v. 58, p. 763–774, june 2007.
- LEE, P. M. **Bayesian Statistics**. Third. New York: Arnold, 2004.
- LEMOS, R. C.; SANTOS, R. D. **Manual de descrição e coleta de solo no campo**. 3.ed.. ed. Campinas: Sociedade Brasileira de Ciência do Solo, 1996.
- LINDSEY, J. K. **Parametric statistical Inference**. New York: Oxford University Press, Inc., 2001.
- MARTIN, A. D.; QUINN, K. M.; PARK, J. H. Markov chain monte carlo (mcmc) package. p. 1–103, September 2009. Acesso em 01/10/09. Disponível em: <<http://mcmcpack.wustl.edu>>.
- MATÉRN, B. **Spatial Variation**. Stockholm, 1960.
- MATHERON, G. Principles of geostatistics. **Economic Geology**, v. 58, p. 1246–1266, 1963.

MCBRATNEY, A. B.; DE GRUIJTER, J. J.; BRUS, D. J. Spatial prediction and mapping of continuous soil classes. **Geoderma**, p. 39–64, 1992.

MOLIN, J. P. Agricultura de precisão. parte 1: o que é e estado da arte em sensoriamento. **Engenharia Agrícola**, v. 17, p. 97–107, 1997.

MOLIN, J. P. Agricultura de precisão. parte 2: diagnóstico, aplicação localizada e considerações econômicas. **Engenharia Agrícola**, v. 17, p. 108–121, 1997.

MOOD, A. M.; GRAYBILL, F. A.; BOES, D. C. (Ed.). **Introduction to the Theory of Statistics**. Third. USA: McGraw-Hill, 1974.

NELDER, J. A.; MEAD, R. A simplex algorithm for function minimization. **Computer Journal**, v. 7, p. 308–313, 1965.

NOTT, D. J.; FIELDING, M.; LEONTE, D. On a generalization of the Laplace approximation. **Statistics and Probability Letters**, v. 79, p. 1397–1403, 2009.

OBAGE, S. C. (Ed.). **Uma análise bayesiana para dados composicionais**. Dissertação (Mestrado em Estatística). Universidade Federal de São Carlos: São Carlos, 2007.

ODEH, I. O. A.; TOOD, A. J.; TRIANTAFILIS, J. Spatial prediction of soil particle-size fractions as compositional data. **Soil Science**, v. 168, n. 7, p. 501–515, July 2003.

PAULINO, C. D.; TURKMAN, M. A. A.; MURTEIRA, B. **Estatística Bayesiana**. Lisboa: Fundação Calouste Gulbenkian, 2003.

PAWITAN, Y. (Ed.). **In All Likelihood**. New York: Oxford University Press, Inc., 2001.

PAWLOWSKY-GLAHN, V.; EGOZCUE, J. J.; TOLOSANA-DELGADO, R. **Lecture Notes on Compositional Data Analysis**. [s.n.], 2007. Disponível em: <<http://www.sediment.uni-goettingen.de/staff/tolosana/extra/CoDa.pdf>>.

PAWLOWSKY-GLAHN, V.; OLEA, R. A. (Ed.). **Geostatistical Analysis of Compositional Data**. New York: Oxford University Press, Inc., 2004. (Studies in Mathematical Geology;7).

PAWLOWSKY-GLAHN, V.; OLEA, R. A. (Ed.). **Geostatistical Analysis of Compositional Data**. New York: Oxford University Press, Inc., 2004, Seção 6.1. (Studies in Mathematical Geology;7).

PAWLOWSKY-GLAHN, V.; OLEA, R. A.; DAVIS, J. C. Estimation of regionalized compositions: a comparison of three methods. **Mathematical Geology**, v. 27, n. 1, p. 105–127, 1995.

PLUMMER, M. et al. CODA: Convergence diagnosis and output analysis for MCMC. **R News**, v. 6, n. 1, p. 7–11, March 2006. Disponível em: <<http://CRAN.R-project.org/doc/Rnews/>>.

- REICHARDT, K.; TIMM, L. C. **Solo, planta e atmosfera: conceitos, processos e aplicações**. Barueri, SP: Manole, 2004.
- REIS, E. **Estatística Multivariada Aplicada**. Lisboa: Edições Sílabo, LDA, 1997.
- REYMENT, R. A.; SAVAZZI, E. (Ed.). **Aspects of Multivariate Statistical Analysis in Geology**. [S.l.]: Elsevier, 1999.
- RIBEIRO JR., P. J.; DIGGLE, P. J. **Bayesian inference in Gaussian model-based geostatistics**. Lancaster, 1999. Relatório Técnico.
- RIBEIRO JR., P. J.; DIGGLE, P. J. geoR: a package from geostatistical analysis. **R-NEWS**, v. 1, p. 15–18, June 2001. ISSN 1609-3631. Disponível em: <<http://cran.R-project.org/doc/Rnews>>.
- RUE, H.; HELD, L. (Ed.). **Gaussian Markov Random Fields**. Boca Raton: Chapman and Hall, 2005.
- SCHABENBERGER, O.; GOTWAY, C. A. **Statistical Methods for Spatial Data Analysis**. Boca Raton: Chapman and Hall, 2005.
- SCHABENBERGER, O.; PIERCE, F. J. (Ed.). **Contemporary Statistical Models for the Plant and Soil Sciences**. Boca Raton: Taylor and Francis, 2001.
- SCHMIDT, A. M.; GELFAND, A. E. A bayesian coregionalization approach for multivariate pollutant data. **Journal of Geophysical Research**, v. 108, p. 10–1–10–8, 2003.
- SCHMIDT, A. M.; SANSÓ, B. Modelagem bayesiana da estrutura de covariância de processos espaciais e espaço temporais. In: *17 SINAPE e ABE-Associação Brasileira de Estatística*. Caxambu: Associação Brasileira de Estatística, 2006. **Minicurso**.
- SILVA, E. A. A. (Ed.). **Aplicação de um estimador robusto na análise de variabilidade espacial de parâmetros de agricultura de precisão**. Tese (Mestrado em engenharia agrícola). Universidade Estadual do Oeste do Paraná: Cascavel, 2000.
- SILVA, E. A. A. (Ed.). **Aplicação de métodos geoestatísticos multivariados em problemas de mapeamento de variáveis do sistema solo-planta**. Tese (Doutorado em ciências). Universidade de Federal de Curitiba: Curitiba, 2008.
- SILVA, E. A. A. et al. Um estimador robusto e o semivariograma cruzado na análise de variabilidade espacial de atributos de solo e planta. **Acta Scientiarum**, v. 25, p. 365–371, 2003.
- SMYTH, G.; HU, Y.; DUNN, P. statmod package. September 2009. Acesso em 06/11/09. Disponível em: <<http://cran.r-project.org/web/packages/statmod>>.

SRINIVASAN, A. (Ed.). **Handbook of precision agriculture: principles and applications**. New York: The Haworth of Precision Agriculture, 2006.

STEIN, M. L. **Interpolation of Spatial Data: Some Theory for Kriging**. New York: Springer Verlag, 1999.

STEPHENS, M. A. EDF statistics for goodness of fit and some comparisons. **Journal of American Statistical Association**, n. 69, p. 730–737, 1974.

TJELMELAND, H.; LUND, K. V. Bayesian modelling of spatial compositional data. **Journal of Applied Statistics**, v. 30, p. 87–100, 2003.

TOLOSANA-DELGADO, R.; OTERO, N.; PAWLOWSKY-GLAHN, V. Some basic concepts of compositional geometry. **Mathematical Geology**, v. 37, n. 7, p. 563–580, October 2005.

VENABLES, W. N.; RIPLEY, B. D. *Modern Applied Statistics with S*. Fourth. USA: Springer, 2002.

WACKERNAGEL, H. (Ed.). **Multivariate Geostatistics: An Introduction with Applications**. Second. Germany: Springer, 1998.

ANEXO A – Pacote *geoComp*

Pacote *geoComp*

Arquivos de Dados:

pivo

	X	Y	DenSoil	DenPar	Porotot	Areia	Silte	Argila
1	20	0	1.69	2.63	35.61	31	25	44
2	40	0	1.58	2.87	45.08	24	21	55
3	60	0	1.44	2.55	43.51	23	27	50
4	80	0	1.50	2.57	41.48	22	25	53
5	100	0	1.58	2.56	38.31	22	25	53
.
.
.
81	40	180	1.64	2.56	35.75	25	23	52
82	60	180	1.65	2.70	38.82	30	23	47

tab.coord.pol

	Areia	Silte	Argila	Poligono
1	0.40000	0.00000	0.60000	1
2	0.00000	0.40000	0.60000	1
3	0.00000	0.00000	1.00000	1
4	0.40000	0.00000	0.60000	1
11	0.39998	0.00001	0.60001	2
.	.	.	.	
.	.	.	.	
.	.	.	.	
312	0.85000	0.15000	0.00000	13
412	0.90001	0.00001	0.09998	13

Funções do Pacote:

agl.R

```
agl <-
function(dados){
    acre <- exp(data.frame(dados[1],dados[2],0))
    compos <- acre/rowSums(acre)
    return(compos)
}
```

as.geoComp.R:

```
as.geoComp <-
function(original, # Um data frame contendo Areia Silte Argila X Y
    perc=TRUE # Dados em valores percentuais
){
    if(perc == TRUE){
        malha <- transform(original,Areia=(Areia/100),Silte=(Silte/100),
            Argila=(Argila/100))
    }
    malha <- original
    malha <- transform(malha,y1=log(Areia/Argila),y2=log(Silte/Argila))
    seq11 <- seq(1,length(malha$y1)*2,by=2)
    seq22 <- seq(2,length(malha$y2)*2,by=2)
    y <- c()
    y[seq11] <- malha$y1
    y[seq22] <- malha$y2
    saida <- list()
    saida[[1]] <- malha[c(6,7)]
    saida[[2]] <- data.frame(y)
    saida[[3]] <- data.frame(original[4],original[5])
    names(saida[[1]]) <- c("Y1","Y2")
    names(saida[[2]]) <- c("Y")
    names(saida[[3]]) <- c("Coord.X","Coord.Y")
    class(saida) <- "geoComp"
    return(saida)
}
```

classificaSolo.R:

```

classificaSolo <-
function(dados,class.dados=TRUE,plot.dados=TRUE,
        newClass=TRUE,new.comp,plot.new=TRUE){
  ## dados: data frame com 3 colunas - areia,silte,argila
  ## plot.dados: gera o diagrama de class. incluindo a amostra
  ## newClass: TRUE, se deseja classificar uma nova composição
  ## new.comp: vetor correspondente aos valores da nova composição
  ## plot.new: plota o ponto a classificar no diagrama
  ##*****
  ## Composicao
  ##*****
  dados <- as.matrix(dados)
  res <- c()
  for(i in 1:nrow(dados)) {
    res <- rowSums(dados)
    VF <- res[i] != 100
    if(VF=='TRUE'){
      #print('soma difere de 1')
    }
    dados[i,] <- (dados[i,]/sum(dados[i,]))*100
  }
  data(tab.coord.pol)
  tab.pol <- tab.coord.pol
  poligono <- list()
  poligono[[1]]<- tab.pol[1:4,]
  poligono[[2]]<- tab.pol[5:10,]
  poligono[[3]]<- tab.pol[11:14,]
  poligono[[4]]<- tab.pol[15:18,]
  poligono[[5]]<- tab.pol[19:23,]
  poligono[[6]]<- tab.pol[24:28,]
  poligono[[7]]<- tab.pol[29:34,]
  poligono[[8]]<- tab.pol[42:35,]
  poligono[[9]]<- tab.pol[43:48,]
  poligono[[10]]<- tab.pol[49:55,]
  poligono[[11]]<- tab.pol[56:60,]
  poligono[[12]]<- tab.pol[61:65,]

```



```

poligono[[13]]<- tab.pol[66:69,]
#####
## Composicao 2D
#####
require(sp)
poligono2D <- list()
poligono2D[[1]] <- Polygon(as.matrix(data.frame(x=c(c((poligono[[1]]$Silte-
poligono[[1]]$Areia)*sqrt(1/3),poligono[[1]]$Argila)[1:4]),
y=c((poligono[[1]]$Silte-poligono[[1]]$Areia)*sqrt(1/3),
poligono[[1]]$Argila)[5:8])))
poligono2D[[2]] <- Polygon(as.matrix(data.frame(c(c((poligono[[2]]$Silte-
poligono[[2]]$Areia)*sqrt(1/3),poligono[[2]]$Argila)[1:6]),
c((poligono[[2]]$Silte-poligono[[2]]$Areia)*sqrt(1/3),
poligono[[2]]$Argila)[7:12])))
poligono2D[[3]] <- Polygon(as.matrix(data.frame(c(c((poligono[[3]]$Silte-
poligono[[3]]$Areia)*sqrt(1/3),poligono[[3]]$Argila)[1:4]),
c((poligono[[3]]$Silte-poligono[[3]]$Areia)*sqrt(1/3),
poligono[[3]]$Argila)[5:8])))
poligono2D[[4]] <- Polygon(as.matrix(data.frame(c(c((poligono[[4]]$Silte-
poligono[[4]]$Areia)*sqrt(1/3),poligono[[4]]$Argila)[1:4]),
c((poligono[[4]]$Silte-poligono[[4]]$Areia)*sqrt(1/3),
poligono[[4]]$Argila)[5:8])))
poligono2D[[5]] <- Polygon(as.matrix(data.frame(c(c((poligono[[5]]$Silte-
poligono[[5]]$Areia)*sqrt(1/3),poligono[[5]]$Argila)[1:5]),
c((poligono[[5]]$Silte-poligono[[5]]$Areia)*sqrt(1/3),
poligono[[5]]$Argila)[6:10])))
poligono2D[[6]] <- Polygon(as.matrix(data.frame(c(c((poligono[[6]]$Silte-
poligono[[6]]$Areia)*sqrt(1/3),poligono[[6]]$Argila)[1:5]),
c((poligono[[6]]$Silte-poligono[[6]]$Areia)*sqrt(1/3),
poligono[[6]]$Argila)[6:10])))
poligono2D[[7]] <- Polygon(as.matrix(data.frame(c(c((poligono[[7]]$Silte-
poligono[[7]]$Areia)*sqrt(1/3),poligono[[7]]$Argila)[1:6]),
c((poligono[[7]]$Silte-poligono[[7]]$Areia)*sqrt(1/3),
poligono[[7]]$Argila)[7:12])))
poligono2D[[8]] <- Polygon(as.matrix(data.frame(c(c((poligono[[8]]$Silte-
poligono[[8]]$Areia)*sqrt(1/3),poligono[[8]]$Argila)[1:8]),
c((poligono[[8]]$Silte-poligono[[8]]$Areia)*sqrt(1/3),
poligono[[8]]$Argila)[9:16])))

```

```

poligono2D[[9]] <- Polygon(as.matrix(data.frame(c(c((poligono[[9]]$Silte-
      poligono[[9]]$Areia)*sqrt(1/3),poligono[[9]]$Argila)[1:6]),
      c((poligono[[9]]$Silte-poligono[[9]]$Areia)*sqrt(1/3),
      poligono[[9]]$Argila)[7:12])))

poligono2D[[10]] <- Polygon(as.matrix(data.frame(c(c((poligono[[10]]$Silte-
      poligono[[10]]$Areia)*sqrt(1/3),poligono[[10]]$Argila)[1:7]),
      c((poligono[[10]]$Silte-poligono[[10]]$Areia)*sqrt(1/3),
      poligono[[10]]$Argila)[8:14])))

poligono2D[[11]] <- Polygon(as.matrix(data.frame(c(c((poligono[[11]]$Silte-
      poligono[[11]]$Areia)*sqrt(1/3),poligono[[11]]$Argila)[1:5]),
      c((poligono[[11]]$Silte-poligono[[11]]$Areia)*sqrt(1/3),
      poligono[[11]]$Argila)[6:10])))

poligono2D[[12]] <- Polygon(as.matrix(data.frame(c(c((poligono[[12]]$Silte-
      poligono[[12]]$Areia)*sqrt(1/3),poligono[[12]]$Argila)[1:5]),
      c((poligono[[12]]$Silte-poligono[[12]]$Areia)*sqrt(1/3),
      poligono[[12]]$Argila)[6:10])))

poligono2D[[13]] <- Polygon(as.matrix(data.frame(c(c((poligono[[13]]$Silte-
      poligono[[13]]$Areia)*sqrt(1/3),poligono[[13]]$Argila)[1:4]),
      c((poligono[[13]]$Silte-poligono[[13]]$Areia)*sqrt(1/3),
      poligono[[13]]$Argila)[5:8])))

##*****
## Diagrama de Classificacao
##*****
if(plot.dados==TRUE){
z <- SpatialPolygons(list(Polygons(poligono2D,"Triangulo de Classificacao")))
par(mgp=c(1.7,0.7,0))
plot(z,axes='F')
## legendas dos vertices
sp.text(loc=c(360,95),txt='Argila',add=TRUE)
sp.text(loc=c(100,557),txt='Areia',add=TRUE)
sp.text(loc=c(610,557),txt='Silte',add=TRUE)
## legendas internas aos poligonos
sp.text(loc=c(360,200),txt='Muito argilosa',cex=0.8,add=TRUE)
sp.text(loc=c(180,520),txt='Areia',cex=0.8,add=TRUE)
sp.text(loc=c(130,535),txt='Areia',cex=0.8,add=TRUE)
sp.text(loc=c(200,535),txt='Franca',cex=0.8,add=TRUE)
sp.text(loc=c(250,500),txt='Franco-arenoso',cex=0.8,add=TRUE)
sp.text(loc=c(235,420),txt='Franco',cex=0.8,add=TRUE)

```

```

sp.text(loc=c(233,440),txt='argilo-arenosa',cex=0.8,add=TRUE)
sp.text(loc=c(230,365),txt='Argila',cex=0.8,add=TRUE)
sp.text(loc=c(250,380),txt='arenosa',cex=0.8,add=TRUE)
sp.text(loc=c(270,300),txt='Argila',cex=0.8,add=TRUE)
sp.text(loc=c(350,460),txt='Franco',cex=0.8,add=TRUE)
sp.text(loc=c(380,415),txt='Franco-argiloso',cex=0.8,add=TRUE)
sp.text(loc=c(480,480),txt='Franco-siltoso',cex=0.8,add=TRUE)
sp.text(loc=c(555,520),txt='Silte',cex=0.8,add=TRUE)
sp.text(loc=c(460,390),txt='Franco',cex=0.8,add=TRUE)
sp.text(loc=c(475,410),txt='argilo-siltoso',cex=0.8,add=TRUE)
sp.text(loc=c(460,338),txt='Argila',cex=0.8,add=TRUE)
sp.text(loc=c(460,350),txt='siltosa',cex=0.8,add=TRUE)
}
##
if(class.dados==TRUE){
  class.amostra <- c()
  for(i in 1:nrow(dados)){
    x <- dados[i,]
    class.amostra[i] <- classSolo(x,poligono2D)
    print(i)
  }
  classifica.amostra <- data.frame(dados,class.amostra)
  Tabela.class <- data.frame(table(class.amostra)/100)
}
if(plot.dados==TRUE){
  dados.comp <- dados/100
  dados.comp.2D <- data.frame((dados.comp[,2]-dados.comp[,1])*sqrt(1/3),
                              dados.comp[,3])
  names(dados.comp.2D) <- c('x','y')
  points(dados.comp.2D$x,dados.comp.2D$y,cex=0.5)
}
if(newClass==TRUE){
  x <- new.comp
  resul.class <- classSolo(x,poligono2D)
}
if(plot.new==TRUE){
  x <- new.comp/100
  x2D <- c((x[2]-x[1])*sqrt(1/3),x[3])
}

```

```

    points(x=x2D[1],y=x2D[2],col='red',cex=1,pch=18)
  }
  retorna <- list()
  retorna[[1]] <- classifica.amostra
  retorna[[2]] <- Tabela.class
  if(newClass==TRUE){
    retorna[[3]] <- resul.class
  }
  return(retorna)
}

```

classSolo.R:

```

classSolo <-
function(x,poligono2D){
  ## x: uma composicao
  ## poligono2D
  x <- x/100
  x2D <- c((x[2]-x[1])*sqrt(1/3),x[3])
  pto.class <- data.frame(x=x2D[1],y=x2D[2])
  resp <- c()
  saida <- c()
  for(i in 1:13){
    poli.pto.class <- data.frame(coordinates(poligono2D[[i]]))
    resp <- .geoR_inout(pto.class,poli.pto.class)
    if(resp=='TRUE' & i=='1') saida <- 'Muito argilosa'
    if(resp=='TRUE' & i=='2') saida <- 'Argila'
    if(resp=='TRUE' & i=='3') saida <- 'Argila arenosa'
    if(resp=='TRUE' & i=='4') saida <- 'Argila siltosa'
    if(resp=='TRUE' & i=='5') saida <- 'Franco argiloso'
    if(resp=='TRUE' & i=='6') saida <- 'Franco argilo-siltoso'
    if(resp=='TRUE' & i=='7') saida <- 'Franco argilo-arenosa'
    if(resp=='TRUE' & i=='8') saida <- 'Franco arenoso'
    if(resp=='TRUE' & i=='9') saida <- 'Franco'
    if(resp=='TRUE' & i=='10') saida <- 'Franco-siltoso'
    if(resp=='TRUE' & i=='11') saida <- 'Silte'
    if(resp=='TRUE' & i=='12') saida <- 'Areia Franca'
    if(resp=='TRUE' & i=='13') saida <- 'Areia'
  }
}

```

```

if(is.null(saida)== TRUE) saida <- "Sem class"
respos <- c(saida)
return(respos)
}

```

cokri.bayes.R:

```

cokri.bayes <-
function(esti.par,locations,dados.comp){
  nloc <- nrow(locations)*2
  seq.y1 <- seq(1, nloc, by=2)
  seq.y2 <- seq(2, nloc, by=2)
  res <- rowMeans(sapply(1:nrow(esti.par), function(i) {
    print(i)
    temporario <- cokrigagem(esti.par[i,], locations=locations,
                           dados.comp=dados.comp)
    amos.bivar <- mvrnorm(1, mu=temporario[[1]], Sigma=temporario[[2]])
    acre <- exp(cbind(amos.bivar[seq.y1], amos.bivar[seq.y2], 0))
    return(as.vector(acre/rowSums(acre)))
  })))
  retorna <- list()
  retorna$espe.bayes <- matrix(res, ncol=3)
  retorna$locations <- locations
  return(retorna)
}

```

cokrigagem.R:

```

cokrigagem <-
function(esti.par, ## Estimativas obtidas na otimização
        dim.gride, ## Dimensão da grade para predição
        locations, ## Coordenadas das localizações de predição
        dados.comp ## Uma lista com dados e localizações amostradas
        ){
  if(missing(locations)){
    rang.coord.x <- range(dados.comp[[3]][[1]])
    rang.coord.y <- range(dados.comp[[3]][[1]])

```

```

c.x <- seq(rang.coord.x[1],rang.coord.x[2],length=dim.gride)
c.y <- seq(rang.coord.y[1],rang.coord.y[2],length=dim.gride)
locations <- expand.grid(c.x,c.y)
}
esti.par <- as.matrix(esti.par)
n.local <- length(locations[,1])*2
dad <- list()
dad[[2]] <- data.frame(c(rep(0,length=n.local),dados.comp[[2]][[1]]))
dad[[3]] <- as.data.frame(rbind(as.matrix(locations),
                               as.matrix(dados.comp[[3]])))

## Reparametrizando
theta1ck <- c(esti.par[4]/esti.par[3],esti.par[5]/esti.par[3],
             esti.par[6]/esti.par[3],esti.par[7],esti.par[8])

## Montando a matriz de covariância
sigmack <- monta.V(theta1ck,dad)
n.local.mais <- n.local +1
dim.matriz <- dim(sigmack)[1]
S00 <- sigmack[c(1:n.local),c(1:n.local)]
S0y <- sigmack[c(1:n.local),c(n.local.mais:dim.matriz)]
Sy0 <- sigmack[c(n.local.mais:dim.matriz),c(1:n.local)]
Syy <- sigmack[c(n.local.mais:dim.matriz),c(n.local.mais:dim.matriz)]
muy <- rep(c(esti.par[1],esti.par[2]),length(dados.comp[[1]]$Y1))
## Como a média de Y0 é desconhecida, esta é substituída pela média
## geral estimada
muy0 <- rep(c(esti.par[1],esti.par[2]),n.local/2)
SoyInvSyy <- S0y %*% solve(Syy)
## Calculando a média de cokrigagem estimada:
## muck.est não eficiente:
## muck.est <- muy0 + S0y %*% solve(Syy, (dados[[2]][[1]]-muy))
muck.est <- muy0 + SoyInvSyy %*% (dados[[2]][[1]]-muy)
## Calculando a variância de cokrigagem estimada:
## vck.est não eficiente:
## vck.est <- S00 - S0y %*% solve(Syy,Sy0)
## vck.est eficiente:
vck.est <- S00 - SoyInvSyy %*% Sy0
res.cokri <- list()
res.cokri[[1]] <- muck.est
res.cokri[[2]] <- vck.est

```

```

if(missing(locations)) attr(res.cokri, "locations") <- locations
return(res.cokri)
}

```

InfBayes.R:

```

infbayes <-
function(dados,tune,queima,salto,nsim,verbose=TRUE,mediahiper){
  y1 <- dados[[1]]$Y1
  y2 <- dados[[1]]$Y2
  coords <- dados[[3]]
  mu1 <- mean(y1)
  mu2 <- mean(y2)
  var_y1 <- var(y1)
  s1 <- var_y1/2
  tau1 <- s1
  var_y2 <- var(y2)
  s2 <- var_y2/2
  tau2 <- s2
  dim <- range(dist(coords))
  phi <- dim[1] + 0.2*(dim[2]-dim[1])
  rho <- cor(y1,y2)
  eta <- s2/s1
  nu1 <- tau1/s1
  nu2 <- tau2/s1
  theta1 <- c(eta,nu1,nu2,phi,rho)
  if(verbose==TRUE)print("Otimizando a log-verossimilhanca")
  inicial.optim <- optim(theta1, logarit.vero, dados.comp=dados,
                        method="L-BFGS-B",
                        lower=c(1e-32,1e-32,1e-32,1e-32,-1),
                        upper=c(Inf,Inf,Inf,Inf,1),hessian=TRUE)
  print(inicial.optim$par)
  inv.hessiana <-solve(inicial.optim$hessian)
  sd.5 <- sqrt(c(inv.hessiana[1,1],inv.hessiana[2,2],inv.hessiana[3,3],
                inv.hessiana[4,4],inv.hessiana[5,5]))
  tune <- tune
  diag.tune <- diag(c(tune))
  mc.V <- diag.tune %*% inv.hessiana %*% diag.tune
  if(verbose==TRUE)print("Rodando o MCMC")
}

```

```

theta1.inicial <- inicial.optim$par
if(length(mediahiper)==0){
  mh <- MCMCmetrop1R(posteriori,theta1.inicial,
                    mediahiper=c(theta1.inicial[1:4],sd.5[1:4]),
                    dados.comp=dados,burnin=queima,mcmc=nsim,thin=salto,
                    V=mc.V)
}
if(length(mediahiper)==8){
  mh1 <- MCMCmetrop1R(posteriori,theta1.inicial,
                     mediahiper=c(mediahiper[1:4],mediahiper[5:8]),
                     dados.comp=dados,burnin=queima,mcmc=nsim,thin=salto,
                     V=mc.V)
}
#plot(mh)
saida.mh <- mh
n <- length(dados[[2]]$Y)
p <- length(dados[[1]])
D <- cbind(rep(1:0,length=n),rep(0:1,length=n))
sigma <- c()
tamanho <- dim(mh)[1]
sigma.media <- matrix(ncol=3,nrow=tamanho)
for(i in 1:tamanho){
  V <- monta.V(mh[i,],dados)
  ## beta não eficiente
  ## inv.V <- solve(V)
  ## beta <- solve(t(D)%*%inv.V%*%D)%*%(t(D)%*%inv.V%*%dados[[2]]$Y)
  ## beta eficiente
  Rbayes <- chol(V)
  back.bayes1 <- backsolve(Rbayes,D,upper.tri=TRUE,transpose=TRUE)
  back.bayes2 <- backsolve(Rbayes,dados[[2]]$Y,upper.tri=TRUE,
                          transpose=TRUE)
  inv.tDinvVD <- solve(crossprod(back.bayes1,back.bayes1))
  beta <- inv.tDinvVD%*%crossprod(back.bayes1,back.bayes2)
  ## s2 não eficiente:
  ## s2 <- 1/(n-p)*t(dados[[2]]$Y-D%*%beta)%*%inv.V%*%
  (dados[[2]]$Y-D%*%beta)
  ## s2 eficiente:
  desvio.bayes <- dados[[2]]$Y-D%*%beta

```



```

back.desvio.bayes <- backsolve(Rbayes,desvio.bayes,
                              upper.tri=TRUE,transpose=TRUE)
s2 <- 1/(n-p)*crossprod(back.desvio.bayes,back.desvio.bayes)
sigma[i] <- rinvcchisq(1,n-p,s2)
## var.beta não eficiente:
## var.beta <- as.numeric(sigma[i])*solve(t(D)%*%inv.V%*%D)
## var.beta eficiente:
var.beta <- as.numeric(sigma[i])*inv.tDinvVD
sigma.media[i,] <- c(mvrnorm(1,mu=beta,Sigma=var.beta),sqrt(sigma[i]))
print(i)
}
retorna <- list()
retorna[[1]] <- data.frame(sigma.media)
names(retorna[[1]]) <- c('beta1','beta2','sigma1')
retorna[[2]] <- data.frame(mh)
names(retorna[[2]]) <- c('eta','nu1','nu2','phi','rho')
retorna[[3]] <- data.frame(inicial.optim$par)
return(retorna)
}

```

logarit.vero.R:

```

logarit.vero <-
function(theta1,dados.comp,print.pars=FALSE){
  if(theta1[1] <= 0) return(.Machine$double.xmax^0.5)
  if(theta1[2] <= 0) return(.Machine$double.xmax^0.5)
  if(theta1[3] <= 0) return(.Machine$double.xmax^0.5)
  if(theta1[4] < 0) return(.Machine$double.xmax^0.5)
  if(abs(theta1[5]) > 1) return(.Machine$double.xmax^0.5)
  ## Pegando o Y
  y <- dados.comp[[2]][[1]]
  X <- cbind(rep(1:0,length=length(y)),rep(0:1,length=length(y)))
  V <- monta.V(theta1, dados.comp = dados.comp)
  ldetV <- determinant(V,log=TRUE)$modulus[1]
  ## mu não eficiente:
  ## mu <- drop(solve(crossprod(X,solve(V,X))%*%crossprod(X,solve(V,y)))
  ## mu eficiente:
  R <- chol(V)
  back.x <- backsolve(R,X,upper.tri=TRUE,transpose=TRUE)
}

```

```

back.y <- backsolve(R,y,upper.tri=TRUE,transpose=TRUE)
mu <- as.numeric(solve(crossprod(back.x,back.x))%*%
                  crossprod(back.x,back.y))
## Qe não eficiente:
## Qe <- drop(crossprod(y,solve(V,y))-2*crossprod(y,solve(V,X%*%mu))
              +crossprod(mu,crossprod(X,solve(V,X%*%mu))))
## Qe eficiente:
desvio <- y-X%*%mu
back.desvio <- backsolve(R,desvio,upper.tri=TRUE,transpose=TRUE)
Qe <- as.numeric(crossprod(back.desvio,back.desvio))
if(Qe < 0) return(Qe=.Machine$double.xmax^0.5)
n <- length(y)
s1 <- sqrt(Qe/n)
ll <- drop(-0.5*(n*log(2*pi)+n*log(s1^2)+ ldetV +n ))
if(print.pars) print(c(theta1,ll))
return(-ll)
}

```

mec.R:

```

mec <-
function(dados.comp, metodo="L-BFGS-B",print.pars=FALSE, alpha=0.975){
  if(class(dados.comp) != "geoComp") stop("A classe do objeto deve ser
                                          geoComp")

  y1 <- dados.comp[[1]]$Y1
  y2 <- dados.comp[[1]]$Y2
  coords <- dados.comp[[3]]
  X <- cbind(rep(1:0,length=length(y1)*2),rep(0:1,length=length(y1)*2))
  ## Calculando os valores iniciais
  mu1 <- mean(y1)
  mu2 <- mean(y2)
  var_y1 <- var(y1)
  s1 <- var_y1/2
  tau1 <- s1
  var_y2 <- var(y2)
  s2 <- var_y2/2
  tau2 <- s2
  dim <- range(dist(coords))
  ## Um "chute" para phi

```

```

phi <- dim[1] + 0.2*(dim[2]-dim[1])
rho <- cor(y1,y2)
## Pegando o Y
y <- dados.comp[[2]][[1]]
## Reparametrizando
eta <- s2/s1
nu1 <- tau1/s1
nu2 <- tau2/s1
theta1 <- c(eta,nu1,nu2,phi,rho)
## Calculando a log-vero para o chute inicial
ll <- logarit.vero(theta1,dados.comp)
## Otimizando
if(metodo== "L-BFGS-B"){
  estim <- optim(theta1,logarit.vero,dados.comp = dados.comp,
                hessian=TRUE,method=metodo,print.pars=print.pars,
                lower=c(1e-32,1e-32,1e-32,1e-32,-1),
                upper=c(Inf,Inf,Inf,Inf,1))
  ifelse(estim$convergence[1] == 0, print("O algoritmo convergiu"),
        print("Existe problemas na convergencia"))
}
if(metodo=="Nelder-Mead" | metodo=="CG" | metodo=="BFGS"){
  estim <- optim(theta1,logarit.vero,dados.comp = dados.comp,hessian=TRUE,
                method=metodo,print.pars=print.pars)
  ifelse(estim$convergence[1] == 0, print("O algoritmo convergiu"),
        print("Existe problemas na convergencia"))
}
## Voltando aos valores iniciais
thetaest <- estim$par
Vest <- monta.V(thetaest,dados.comp)
ldetVest <- determinant(Vest,log=TRUE)$modulus[1]
## muest não eficiente:
## muest<- drop(solve(crossprod(X,solve(Vest,X)))*%*%
               crossprod(X,solve(Vest,y)))
## muest eficiente:
R.est <- chol(Vest)
back.x.est <- backsolve(R.est,X,upper.tri=TRUE,transpose=TRUE)
back.y.est <- backsolve(R.est,y,upper.tri=TRUE,transpose=TRUE)
invVXX <- solve(crossprod(back.x.est,back.x.est))

```

```

muest <- as.numeric(invXVX%%crossprod(back.x.est,back.y.est))
## Qeest não eficiente:
## Qeest <- drop(crossprod(y,solve(Vest,y))-
                2*crossprod(y,solve(Vest,X%%muest))+
                crossprod(muest,crossprod(X,solve(Vest,X%%muest))))
## Qeest eficiente:
desvio.est <- y-X%%muest
back.desvio.est <- backsolve(R.est,desvio.est,upper.tri=TRUE,
                             transpose=TRUE)
Qeest <- as.numeric(crossprod(back.desvio.est,back.desvio.est))
n <- length(y)
s1est <- sqrt(Qeest/n)
llest <- drop(-0.5*(n*log(2*pi)+n*log(s1est^2)+ ldetVest +n ))
s2est <- thetaest[1]*s1est
taulest <- thetaest[2]*s1est
tau2est <- thetaest[3]*s1est
## Calculando a variância para mu e s1: 1/Io(mu) - 1/Io(s1)
## V.g.mu não eficiente
## V.g.mu <- (s1est)^2*solve(crossprod(X,solve(Vest,X)))
## V.g.mu eficiente
V.g.mu <- (s1est)^2*invXVX
V.g.s1 <- (s1est)^3/(3*Qeest-n*s1est)
## Calculando o gradiente
g<- matrix(c(s1est,0,0,0,0,0,s1est,0,0,0,0,0,s1est,
             0,0,0,0,0,1,0,0,0,0,0,1),nr=5)
## Calculando a variância para s2,tau1,tau2,phi e rho - usa hessiana
V.g <- crossprod(g,diag(1/diag(estim$hessian))%%g)
## Preparando a saída
para <- c("mu1","mu2","s1","s2","tau1","tau2","phi","rho")
estima <- c(muest[1],muest[2],s1est,s2est,taulest,tau2est,
           thetaest[4],thetaest[5])
errosdpd <- qnorm(alpha)*sqrt(c(V.g.mu[1],V.g.mu[2],V.g.s1,diag(V.g)))
## Intervalo de confiança pelo método Delta
ic.min <- estima - errosdpd
ic.max <- estima + errosdpd
retorna <- list()
retorna[1][[1]] <- data.frame(para,estima,errosdpd,ic.min,ic.max)
names(retorna[1][[1]]) <- c("Parametros","Estimativas","Erro Padrao",

```

```

                                "LI.Delta","LS.Delta")
retorna[2][[1]] <-c(11,11lest)
names(retorna[2][[1]]) <- c("LogLik Start", "LogLik Optim")
return(retorna)
}

```

monta.V.R:

```

monta.V <-
function(theta1,dados.comp){
  y <- dados.comp[[2]][[1]]
  eta <- theta1[1]
  nu1 <- theta1[2]
  nu2 <- theta1[3]
  phi <- theta1[4]
  rho <- theta1[5]
  ## Calculando os elementos da matriz de correlação espacial
  distancia <- unname(as.matrix(dist(dados.comp[[3]],diag=TRUE,upper=TRUE)))
  correla <- exp(-distancia/phi)
  ## Organizando a matriz de correlação espacial com eltos y1 e y2
  ## intercalados. Ou seja, calculando R1.
  coluna1 <- rep(c(1, eta),length(y)/2)
  coluna2 <- rep(c(eta,eta^2),length(y)/2)
  seq1 <- seq(1,length(y),by=2)
  seq2 <- seq(2,length(y),by=2)
  ERRE1 <- matrix(ncol=length(y),nrow=length(y))
  ERRE1[,c(seq1)] <- coluna1
  ERRE1[,c(seq2)] <- coluna2
  ## Organizando a matriz de correlação espacial com eltos y1 e y2
  ## intercalados.
  ERRE2 <- matrix(nrow=length(y),ncol=length(y))
  for(i in 1:length(y)/2){
    ERRE2[seq1[i],] <- rep(correla[i,],each=2)
    ERRE2[seq2[i],] <- rep(correla[i,],each=2)
  }
  ## Calculando a matriz R
  ERRE <- ERRE1*ERRE2
}

```

```

## Calculando a matriz de covariância composicional
Ib1 <- matrix(c(nu1^2,nu1*nu2*rho,nu1*nu2*rho,nu2^2),nc=2)
Ib <- kronecker(diag(length(y)/2),Ib1)
## Matriz de covariância espacial composicional
V <- ERRE + Ib
return(V)
}

```

posteriori.R:

```

## theta1=c(eta,nu1,nu2,phi,rho) obtidos da amostra.
## mediahiper=medias para c(eta,nu1,nu2,phi) e correspondentes
## desvios padrão.
posteriori <-
function(theta1,dados.comp,mediahiper,print.pars=FALSE){
  if(theta1[1] <= 0) return(-Inf)
  if(theta1[2] <= 0) return(-Inf)
  if(theta1[3] <= 0) return(-Inf)
  if(theta1[4] <= 0) return(-Inf)
  if(abs(theta1[5]) > 1) return(-Inf)
  ## Pegando o Y
  y <- dados.comp[[2]][[1]]
  X <- cbind(rep(1:0,length=length(y)),rep(0:1,length=length(y)))
  V <- monta.V(theta1, dados.comp = dados.comp)
  ldetV <- determinant(V,log=TRUE)$modulus[1]
  ## mu não eficiente:
  ## mu <- drop(solve(crossprod(X,solve(V,X)))*%crossprod(X,solve(V,y)))
  ## mu eficiente:
  R <- chol(V)
  back.x <- backsolve(R,X,upper.tri=TRUE,transpose=TRUE)
  back.y <- backsolve(R,y,upper.tri=TRUE,transpose=TRUE)
  mu <- as.numeric(solve(crossprod(back.x,back.x))*%
                    crossprod(back.x,back.y))
  ## Qe não eficiente:
  ## Qe <- drop(crossprod(y,solve(V,y))-2*crossprod(y,solve(V,X%*%mu))+
  ##           crossprod(mu,crossprod(X,solve(V,X%*%mu))))
  ## Qe eficiente:
  desvio <- y-X%*%mu
  back.desvio <- backsolve(R,desvio,upper.tri=TRUE,transpose=TRUE)
}

```

```

Qe <- as.numeric(crossprod(back.desvio,back.desvio))
if(Qe < 0) return(Qe=.Machine$double.xmax^0.5)
n <- length(y)
s1 <- sqrt(Qe/n)
ll <- drop(-0.5*(n*log(2*pi)+n*log(s1^2)+ldetV+n))+
      dlnorm(theta1[1],log(0.24),sd=0.3,log=TRUE)+
      dlnorm(theta1[2],log(0.63),sd=0.3,log=TRUE)+
      dlnorm(theta1[3],log(0.59),sd=0.3,log=TRUE)+
      dgamma(theta1[4],66,1,log=TRUE)
## Usuário entrando com os valores dos hiperparâmetros
if(length(mediahiper)==8){
  ll <- drop(-0.5*(n*log(2*pi)+n*log(s1^2)+ldetV+n))+
        dnorm(theta1[1],mediahiper[1],sd=mediahiper[5],log=TRUE)+
        dnorm(theta1[2],mediahiper[2],sd=mediahiper[6],log=TRUE)+
        dnorm(theta1[3],mediahiper[3],sd=mediahiper[7],log=TRUE)+
        dgamma(theta1[4],mediahiper[4],mediahiper[8],log=TRUE)
}
if(print.pars) print(c(theta1,ll))
return(ll)
}

```

resultado.mh.R:

```

resul.mh <-
function(resultado){
  post.beta1 <- resultado[[1]]$beta1
  post.beta2 <- resultado[[1]]$beta2
  post.sigma1 <- resultado[[1]]$sigma1
  post.sigma2 <- post.sigma1*resultado[[2]]$eta
  post.tau1 <- post.sigma1*resultado[[2]]$nu1
  post.tau2 <- post.sigma1*resultado[[2]]$nu2
  post.phi <- resultado[[2]]$phi
  post.rho <- resultado[[2]]$rho
  saida <- data.frame(post.beta1,post.beta2,post.sigma1,post.sigma2,
                    post.tau1,post.tau2,post.phi,post.rho)
  par(mfrow=c(2,4))
  plot(density(post.beta1),main="Posteriori mu1")
  plot(density(post.beta2),main="Posteriori mu2")
  plot(density(post.sigma1),main="Posteriori s1")

```

```

plot(density(post.sigma2),main="Posteriori s2")
plot(density(post.tau1),main="Posteriori t1")
plot(density(post.tau2),main="Posteriori t2")
plot(density(post.phi),main="Posteriori phi")
plot(density(post.rho),main="Posteriori rho")
m1<-c(quantile(post.beta1,prob=0.025),mean(post.beta1),
      quantile(post.beta1,prob=0.975))
m2<-c(quantile(post.beta2,prob=0.025),mean(post.beta2),
      quantile(post.beta2,prob=0.975))
s1<-c(quantile(post.sigma1,prob=0.025),mean(post.sigma1),
      quantile(post.sigma1,prob=0.975))
s2<-c(quantile(post.sigma2,prob=0.025),mean(post.sigma2),
      quantile(post.sigma2,prob=0.975))
t1<-c(quantile(post.tau1,prob=0.025),mean(post.tau1),
      quantile(post.tau1,prob=0.975))
t2<-c(quantile(post.tau2,prob=0.025),mean(post.tau2),
      quantile(post.tau2,prob=0.975))
phi<-c(quantile(post.phi,prob=0.025),median(post.phi),
      quantile(post.phi,prob=0.975))
rho<-c(quantile(post.rho,prob=0.025),mean(post.rho),
      quantile(post.rho,prob=0.975))
tabela <- t(data.frame(m1,m2,s1,s2,t1,t2,phi,rho))
estimativa.optim <- estima.bayes[[3]]
retorna <- list()
retorna[[1]] <- saida
retorna[[2]] <- tabela
retorna[[3]] <- estimativa.optim
return(retorna)
}

```

volta.cokri.R

```

volta.cokri <-
function(mat.cokri, num.simu, retorna.tudo=FALSE, int.conf=0.95){
  nlinhas <- dim(mat.cokri[[1]])[1]
  compos1 <- data.frame(matrix(nrow=nlinhas/2,ncol=3))
  compos <- data.frame(matrix(nrow=nlinhas/2,ncol=3))
  for(i in 1:num.simu){
    g <- mvrnorm(n=1, mat.cokri[[1]],mat.cokri[[2]])

```



```

seq1 <- seq(1,nlinhas,by=2)
seq2 <- seq(2,nlinhas,by=2)
y1 <- g[seq1]
y2 <- g[seq2]
gerado <- data.frame(y1,y2)
compos <- agl(gerado)
compos1 <- cbind(compos,compos1)
}
compos2 <- as.matrix(compos1)
dim.vetor <- num.simu*3
sy1 <- seq(1,dim.vetor,by=3)
sy2 <- seq(2,dim.vetor,by=3)
sy3 <- seq(3,dim.vetor,by=3)
amostra1 <- as.matrix(compos2[,sy1],ncol=num.simu)
amostra2 <- as.matrix(compos2[,sy2],ncol=num.simu)
amostra3 <- as.matrix(compos2[,sy3],ncol=num.simu)
if(retorna.tudo == TRUE){
  retorna <- list()
  retorna[[1]] <- amostra1
  retorna[[2]] <- amostra2
  retorna[[3]] <- amostra3
  return(retorna)
}
if(retorna.tudo == FALSE){
  med1 <- apply(amostra1,1,mean)
  med2 <- apply(amostra2,1,mean)
  med3 <- apply(amostra3,1,mean)
  q1 <- t(apply(amostra1,1,quantile,prob=c(1-int.conf,int.conf)))
  q2 <- t(apply(amostra2,1,quantile,prob=c(1-int.conf,int.conf)))
  q3 <- t(apply(amostra3,1,quantile,prob=c(1-int.conf,int.conf)))
  quantis <- cbind(q1,q2,q3)
  resultado <- list()
  resultado$preditos <- data.frame(med1,med2,med3)
  names(resultado$preditos) <- c("Areia","Silte","Argila")
  resultado$intervalo <- data.frame(quantis)
  names(resultado$intervalo) <- c("LI Areia", "LS Areia",
    "LI Silte", "LS Silte", "LI Argila", "LS Argila")
  return(resultado)
}

```

```

}
}

```

volta.quad.R:

```

volta.quad <-
function(med.cov,          # Média e covariância de cokrigagem.
        n.pontos=7,      # Ordem da quadratura de Gauss-Hermite.
        Variancia=FALSE # Não calcula a variância.
        ){
mu.ck <- med.cov[[1]]
sigma.ck <- med.cov[[2]]
desvio.pd <- sqrt(diag(sigma.ck))
ic.mu <- data.frame(mu.ck-qnorm(0.975)*desvio.pd,mu.ck+
                    qnorm(0.975)*desvio.pd)
names(ic.mu) <- c("L.Minimo","L.Maximo")
g <- function(Y,mu,R,pos){
  arg.agl <-mu+sqrt(2)*t(R)%*%Y
  g1 <- t(pi^-1*agl(arg.agl))
  g2 <- pi^-1*(t(as.matrix(agl(arg.agl)))-g1)%*%
        t(t(as.matrix(agl(arg.agl)))-g1)
  resultado <- cbind(g1,g2)
  return(resultado[pos])
}
## Calculando a quadratura de Gauss-Hermite
quad.gauss.comp <- function(mu,R,func=g,pos,np=n.pontos){
  AbcPeso <- gauss.quad(np,kind='hermite')
  Y <- AbcPeso$nodes
  peso<- AbcPeso$weights
  soma <- 0
  for(i in 1:np){
    for(j in 1:np){
      soma<-peso[i]*peso[j]*func(Y=c(Y[i],Y[j]),mu=mu,R=R,
                                  pos=pos)+soma
    }
  }
  return(soma)
}
## Calculando a quadratura de Gauss-Hermite para o vetor de médias

```

```

n.linhas <- length(mu.ck)/2
compo.md <- matrix(ncol=3,nrow=n.linhas)
seq1 <- seq(1,n.linhas*2,by=2)
seq2 <- seq(2,n.linhas*2,by=2)
for(i in 1:n.linhas){
  sigma.ck <- med.cov[[2]][seq1[i]:seq2[i],seq1[i]:seq2[i]]
  mu.ck <- med.cov[[1]][seq1[i]:seq2[i]]
  cp <- quad.gauss.comp(mu=mu.ck,R=chol(sigma.ck),func=g,pos=c(1:3),
                        np=n.pontos)
  compo.md[i,] <- cp
}
#rowSums(compo.md)
## Calculando a quadratura de Gauss-Hermite para a matriz de
## covariâncias
if(Variancia == TRUE){
compo.var <- matrix(ncol=3,nrow=n.linhas)
for(i in 1:n.linhas){
  sigma.ck <- med.cov[[2]][seq1[i]:seq2[i],seq1[i]:seq2[i]]
  mu.ck <- med.cov[[1]][seq1[i]:seq2[i]]
  cp <- quad.gauss.comp(mu=mu.ck,R=chol(sigma.ck),func=g,pos=c(4,8,12),
                        np=n.pontos)
  compo.var[i,] <- cp
}}
retorna <- list()
retorna[[1]] <- compo.md
if(Variancia == TRUE){
retorna[[2]] <- compo.var}
return(retorna)
}

```

ANEXO B – Help do pacote *geoComp*

geoComp-package *Análise geoestatística de dados composicionais.*

Description

A função ajusta um modelo geoestatístico bivariado para dados composicionais com três componentes, tanto pelo método clássico como por inferência bayesiana.

Details

Package: geoComp
Type: Package
Version: 1.0
Date: 2010-08-20
License: GPL -2
LazyLoad: yes

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,
Ana Beatriz Tozzo Martins <abtmartins@uem.br>,
Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H. (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n. 3, 456-477.

agl *Transformação ALR inversa.*

Description

A função transforma um vetor bivariado, de dados ALR transformados, em um vetor trivariado, uma composição.

Usage

agl(dados)

Arguments

dados Vetor bivariado de dados ALR transformados.

Value

A função retorna um vetor correspondente a composição.

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,
 Ana Beatriz Tozzo Martins <abtmartins@uem.br>,
 Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H., (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n. 3, 456-477.

Examples

```
x<-c(0.5,0.3)
agl(x)
```

`as.geoComp`*Transforma um data frame em um objeto da classe geoComp.*

Description

A função faz a transformação ALR dos dados e cria um objeto da classe `geoComp`.

Usage

```
as.geoComp(original, perc = TRUE)
```

Arguments

<code>original</code>	Data frame contendo os três componentes.
<code>perc</code>	Lógico, indicando se os dados estão expressos em valores percentuais ('perc = TRUE') ou se os dados estão em valores decimais ('perc = FALSE'). O default é 'TRUE'.

Details

As composições são preparadas para análise geoestatística bivariada.

Value

A função retorna uma lista contendo os seguintes elementos:

<code>comp1</code>	Data frame contendo os dados bivariados alr transformados.
<code>comp2</code>	Vetor contendo os dados bivariados ALR transformados arranjados de forma intercalada por variável.
<code>comp3</code>	Data frame contendo a localização espacial. Geralmente coordenadas Euclidianas num plano.

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,
Ana Beatriz Tozzo Martins <abtmartins@uem.br>,
Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

- Aitchison, J. (1986). *The Statistical analysis of compositional data*, The Blackburn Press, New Jersey.
- Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H. (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n.3, 456-477.
- Pawlowsky-Glahn, V., and Olea, R. A. (2004). *Geostatistical Analysis of Compositional Data*, Oxford University Press, Inc., New York.

Examples

```
### Carregando pacote
require(geoComp)
### Carregando os dados
data(pivo)
### Selecionando as colunas referentes aos Componentes 1, 2 e 3,
### Coordenada X e Coordenada Y
dados <- pivo[,c(6,7,8,1,2)]
### Transformando o objeto "dados" para a classe geoComp
dados <- as.geoComp(dados)
```

```
classificaSolo      Classificação de solo.
```

Description

A função faz a classificação de solo em classes texturais segundo Reichardt e Timm (2004).

Usage

```
classificaSolo(dados, class.dados = TRUE, plot.dados = TRUE, newClass = TRUE,
               new.comp, plot.new = TRUE)
```

Arguments

dados Data.frame contendo 3 colunas correspondentes aos percentuais de areia, silte e argila.

`classSolo` *Classificação de uma composição.*

Description

A função faz a classificação de uma composição de acordo com classificação textural apresentada em Reichardt e Timm (2004).

Usage

```
classSolo(x, poligono2D)
```

Arguments

`x` Vetor correspondente a uma composição.

`poligono2D` Lista contendo as coordenadas dos vértices dos polígonos no diagrama de classificação textural.

Value

A função retorna a classificação de uma composição.

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,
Ana Beatriz Tozzo Martins <abtmartins@uem.br>,
Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

Reichardt, K. and Timm, L. C.(2004). *Solo, Planta e Atmosfera: Conceitos, Processos e Aplicações*. Manole, Barueri, SP.

`cokri.bayes` *Predição espacial bayesiana.*

Description

A função realiza o processo de predição espacial através de métodos bayesianos.

Usage

```
cokri.bayes(esti.par, locations, dados.comp)
```

Arguments

`esti.par` Matriz de estimativas dos parâmetros resultante do processo de estimação bayesiana usando o algoritmo Metropolis-Hastings.

`locations` Data frame com as localizações de predição.

`dados.comp` Objeto da classe `geoComp`.

Value

A função retorna uma lista com os seguintes elementos:

`comp1` matriz dos componentes preditos.

`comp2` data frame com as localizações de predição.

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,
Ana Beatriz Tozzo Martins <abtmartins@uem.br>,
Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H., (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n. 3, 456-477.

Diggle, P. J., and Ribeiro, Jr., P. J. (2007). *Model-based geostatistics*. Springer Series in Statistics, USA.

Examples

```

### Carregando pacotes
require(geoComp)
require(MCMCpack)
require(geoR)
require(MASS)
### Carregando os dados
data(pivo)
### Selecionando as colunas referentes aos componentes 1, 2 e 3,
### Coordenada X e coordenada Y
dados <- pivo[,c(6,7,8,1,2)]
### Transformando o objeto "dados" para a classe geoComp
dados <- as.geoComp(dados)
### Fazendo estimação bayesiana
estima.bayes <- infbayes(dados,tune=c(0.5,0.5,0.5,0.5,0.5),queima=1,salto=5,
                        nsim=20,mediahiper=c())
summary(estima.bayes[[2]])
### Organizando os resultados
res.mh <- resultado.mh(resultado=estima.bayes)
res.mh.ic <- data.frame(res.mh[[2]])
names(res.mh.ic) <- c('2.5%', 'media', '97.5%')
metro.bayes <- data.frame(res.mh[[1]])
### Construindo a borda da área de predição
bor <- cbind(c(0,seq(0,200,l=100),0),c(0,sqrt(200^2-seq(0,200,l=100)^2),0))
### Construindo uma grade de predição
gr <- pred_grid(bor, by=70)
### Fazendo predição espacial
cokri.bayes <- cokri.bayes(estim.par=metro.bayes,locations=gr,
                        dados.comp=dados)

```

cokrigagem

Predição espacial de dados ALR transformados.

Description

A função calcula o vetor de valores esperados preditos e a respectiva matriz de covariância para os dados ALR transformados utilizando os resultados da distribuição gaussiana

multivariada.

Usage

```
cokrigagem(estim.par, dim.gride, locations, dados.comp)
```

Arguments

`estim.par` Vetor das estimativas dos parâmetros calculados através da função `mec`.

`dim.gride` Vetor numérico com o valor da dimensão da grade de predição.

`locations` Data frame contendo as coordenadas das localizações de predição.

`dados.comp` Um objeto da classe `geoComp`.

Details

A predição espacial de Y_0 em localizações não amostradas $x_0 = (x_{10}, x_{20}, \dots, x_{n_20})$ é realizada segundo resultados da distribuição gaussiana multivariada. O vetor de valores esperados nas n_2 localizações é dado por

$$\mu_{Y_0|Y} = \mu_{Y_0} + \Sigma_{Y_0Y} \Sigma_{YY}^{-1} (Y - \mu_Y)$$

e a matriz de covariância é dada por

$$\Sigma_{Y_0|Y} = \Sigma_{Y_0Y_0} - \Sigma_{Y_0Y} \Sigma_{YY}^{-1} \Sigma_{YY_0}.$$

Value

A função retorna uma lista com os seguintes resultados:

`comp1` vetor dos valores esperados preditos.

`comp2` matriz de covariância predita.

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,
 Ana Beatriz Tozzo Martins <abtmartins@uem.br>,
 Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

Diggle, P. J., and Ribeiro, Jr., P. J., (2007). *Model-based geostatistics*. Springer Series in Statistics, USA.

Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H., (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n. 3, 456-477.

Examples

```
### carregando pacotes
require(geoComp)
require(MASS)
require(statmod)
require(geoR)
### Carregando os dados
data(pivo)
### Selecionando as colunas referentes aos componentes 1, 2 e 3,
### Coordenada X e coordenada Y
dados <- pivo[,c(6,7,8,1,2)]
### Transformando o objeto "dados" para a classe geoComp
dados <- as.geoComp(dados)
### Construindo a borda da área de predição
bor <- cbind(c(0,seq(0,200,l=100),0),c(0,sqrt(200^2-seq(0,200,l=100)^2),0))
### Construindo uma grade de predição
gr <- pred_grid(bor, by=70)
### Ajustando o modelo
estima <- mec(dados)
### Obtendo os valores preditos no R^2
md.cov.ck <- cokrigagem(estima[[1]]$Estimativas, loc=gr,dados.comp=dados)
```

infbayes

Ajuste do modelo por inferência bayesiana para dados composicionais.

Description

A função ajusta um modelo para dados composicionais por métodos bayesianos. Utiliza-se o algoritmo Metropolis-Hastings, função `MCMCmetrop1R` do pacote **MCMCpack** para a estimação dos parâmetros η , v_1 , v_2 , ϕ e ρ .

Usage

```
infbayes(dados, tune, queima, salto, nsim, verbose = TRUE, mediahiper)
```

Arguments

<code>dados</code>	Objeto da classe <code>geoComp</code> .
<code>tune</code>	Vetor com 5 elementos iguais aos limites de variâncias para η , ν_1 , ν_2 , ϕ e ρ .
<code>queima</code>	Vetor numérico correspondente ao número de simulações a serem descartadas como período de aquecimento da cadeia.
<code>salto</code>	Vetor numérico que indica de quanto em quanto os dados são armazenados.
<code>nsim</code>	Vetor numérico correspondente ao número de simulações.
<code>verbose</code>	Lógico, indicando se a mensagem de que a log-verossimilhança está sendo otimizada deve aparecer na tela ('verbose = TRUE') ou não ('verbose = FALSE'). O default é 'TRUE'.
<code>mediahiper</code>	Vetor com 8 elementos correspondentes aos valores dos hiperparâmetros das prioris log-normais para η , ν_1 , ν_2 e Gama para ϕ . Os três primeiros correspondem as médias para η , ν_1 , ν_2 , o quarto é o hiperparâmetro α para ϕ , os três seguintes correspondem aos desvios-padrão para η , ν_1 , ν_2 e o último, o hiperparâmetro β para ϕ . O default é (log(24), log(0.63), log(0.59), 66, 0.3, 0.3, 0.3, 1).

Details

A distribuição a posteriori marginal para $\underline{\mu} | \sigma_1^2, \theta^*, \underline{Y}$ é

$$[\underline{\mu} | \sigma_1^2, \theta^*, \underline{Y}] \sim N(\underline{\hat{\mu}}; \sigma_1^2 (D'V^{-1}D)^{-1}),$$

onde $\underline{\hat{\mu}}$ é dado em `mec.R` e para $\sigma_1^2 | \theta^*, \underline{Y}$ é

$$[\sigma_1^2 | \theta^*, \underline{Y}] \sim \chi_{Simv}^2(n - n_{\underline{\mu}}; S^2),$$

onde χ_{Simv}^2 é a distribuição qui-quadrado escalonada inversa e

$$S^2 = \frac{n}{n - n_{\underline{\mu}}} \hat{\sigma}_1^2 = \frac{(\underline{Y} - D\underline{\hat{\mu}})'V^{-1}(\underline{Y} - D\underline{\hat{\mu}})}{n - n_{\underline{\mu}}}.$$

Value

A função retorna uma lista com os seguintes elementos:

`comp1` data frame contendo os valores a posteriori de β_1 , β_2 e σ_1 .

`comp2` data frame contendo os valores a posteriori de η , v_1 , v_2 , ϕ e ρ .

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,

Ana Beatriz Tozzo Martins <abtmartins@uem.br>,

Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

Diggle, P. J., and Ribeiro, Jr., P. J., (2007). *Model-based geostatistics*. Springer Series in Statistics, USA.

Gamerman, D., and Lopes, H. F., (2006). *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. 2.ed. Chapman and Hall/CRC, Londres.

Gelman, A., and Carlin, J. B., and Stern, H. S., and Rubin, D. B., (2003). *Bayesian Data Analysis*. 2.ed. Chapman and Hall, Boca Raton.

Gilks, W. R., and Richardson, S., and Spiegelhalter, D. J., (1996). *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London.

Gill, J., (2000). *Bayesian Methods For The Social And Behavioral Sciences*. Chapman and Hall, London.

Lee, P. M., (2004). *Bayesian Statistics*. 3.ed. Arnold, New York.

Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H., (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n. 3, 456-477.

Ribeiro, Jr, P. J. and Diggle, P. J., (1999). *Bayesian inference in Gaussian model-based geostatistics*. ST-99-08., Dept. of Maths and Stats, Lancaster University, Lancaster, Technical Report.

Rue, H., and Held, L., (2005). *Gaussian Markov Random Fields*. Chapman and Hall, Boca Raton.

Examples

```
### Carregando pacotes
require(geor)
```

```

require(geoComp)
require(MCMCpack)
require(MASS)
### Carregando os dados
data(pivo)
### Selecionando as colunas referentes aos componentes 1, 2 e 3,
### Coordenada X e coordenada Y
dados <- pivo[,c(6,7,8,1,2)]
### Transformando o objeto "dados" para a classe geoComp
dados <- as.geoComp(dados)
### Fazendo o ajuste do modelo por métodos bayesianos
estima.bayes <- infbayes(dados,tune=c(0.5,0.5,0.5,0.5,0.5),queima=1,salto=5,
                          nsim=20,mediahiper=c())

```

logarit.vero *Log-Verossimilhança.*

Description

A função calcula a log-verossimilhança do modelo.

Usage

```
logarit.vero(theta1, dados.comp, print.pars = FALSE)
```

Arguments

theta1 Vetor com valores para os parâmetros η , v_1 , v_2 , ϕ e ρ .

dados.comp Objeto da classe geoComp.

print.pars Lógico. Se 'TRUE', indica que os parâmetros e o valor do negativo da log-verossimilhança (a menos de uma constante) são printadas na tela cada vez que a função a ser minimizada for chamada.

Details

A função de verossimilhança do modelo é

$$L(\theta; Y) = (2\pi)^{-n/2} |\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(Y - \mu_Y)' \Sigma^{-1} (Y - \mu_Y)\right\},$$

com vetor de parâmetros $\theta = (\underline{\mu}, \sigma_1, \sigma_2, \tau_1, \tau_2, \phi, \rho)$. A função de log-verossimilhança com a reparametrização $\eta = \frac{\sigma_2}{\sigma_1}$, $v_1 = \frac{\tau_1}{\sigma_1}$ e $v_2 = \frac{\tau_2}{\sigma_1}$ é

$$l(\theta; \underline{Y}) = -\frac{1}{2}(n \ln(2\pi) + 2n \ln(\sigma_1) + \ln(|V|) + \frac{1}{\sigma_1^2} Qe).$$

em que $Qe = (\underline{Y} - \underline{\mu}_Y)' V^{-1} (\underline{Y} - \underline{\mu}_Y)$. Ao substituir as estimativas de máxima verossimilhança,

$$\hat{\underline{\mu}} = (D'V^{-1}D)^{-1}(D'V^{-1}\underline{Y}) \quad e \quad \hat{\sigma}_1 = \sqrt{\hat{Q}e/n}.$$

na função de verossimilhança tem-se a função de log-verossimilhança concentrada

$$l(\underline{\theta}^*; \underline{Y}) = -\frac{1}{2}[\ln(|V|) + n(\ln(2\pi) + \ln(\hat{Q}e) - \ln(n) + 1)]$$

com $\theta^* = (\eta, v_1, v_2, \phi, \rho)$.

Value

A função retorna um vetor numérico contendo o negativo do valor da log-verossimilhança do modelo.

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,

Ana Beatriz Tozzo Martins <abtmartins@uem.br>,

Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H., (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n. 3, 456-477.

Examples

```
### Carregando pacote
require(geoComp)
### Carregando dados
data(pivo)
### Selecionando dados
dados <- pivo[1:15,c(6,7,8,1,2)]
### Transformando o objeto "dados" para a classe geoComp
```

```

dados.comp <- as.geoComp(dados)
y1 <- dados.comp[[1]]$Y1
y2 <- dados.comp[[1]]$Y2
coords <- dados.comp[[3]]
### Calculando os valores iniciais
mu1 <- mean(y1)
mu2 <- mean(y2)
var_y1 <- var(y1)
s1 <- var_y1/2
tau1 <- s1
var_y2 <- var(y2)
s2 <- var_y2/2
tau2 <- s2
dim <- range(dist(coords))
phi <- dim[1] + 0.2*(dim[2]-dim[1])
rho <- cor(y1,y2)
y <- dados.comp[[2]][[1]]
### Reparametrizando
eta <- s2/s1
nu1 <- tau1/s1
nu2 <- tau2/s1
theta1 <- c(eta,nu1,nu2,phi,rho)
### Calculando a matriz de covariância
V <- monta.V(theta1,dados.comp)
### Calculando o logaritmo da vero. para o theta1 inicial
ll <- logarit.vero(theta1,dados.comp)

```

mec

Ajuste de modelo baseado em verossimilhança para dados composicionais espaciais.

Description

A função ajusta um modelo espacial baseado em verossimilhança para dados composicionais.

Usage

```
mec(dados.comp, metodo = "L-BFGS-B", print.pars = FALSE, alpha = 0.975)
```

Arguments

<code>dados.comp</code>	Objeto da classe <code>geoComp</code> .
<code>metodo</code>	Algoritmo utilizado na função <code>optim</code> . Outras opções: “Nelder-Mead”, “CG” e “BFGS”.
<code>print.pars</code>	Lógico. Se ‘TRUE’, indica que os parâmetros e o valor do negativo da log-verossimilhança (a menos de uma constante) são printadas na tela cada vez que a função a ser minimizada for chamada. O default é ‘FALSE’
<code>alpha</code>	Vetor numérico correspondente ao nível crítico utilizado no cálculo do intervalo de confiança.

Details

Esta função estima os parâmetros do modelo de campo aleatório gaussiano especificado como:

$$Y_1(x) = \mu_1(x) + S_1(x) + Z_1(x)$$

$$Y_2(x) = \mu_2(x) + S_2(x) + Z_2(x)$$

onde

- x define uma localização espacial. Geralmente coordenadas Euclidianas num plano.
- Y_1 e Y_2 são as variáveis observadas na localização x e constituem o vetor Y .
- $\underline{\mu}(x) = D\underline{\mu}$ é o componente média do modelo (tendência).
- $S(x)$ é um processo gaussiano estacionário com variância σ^2 (patamar) e uma função de correlação exponencial com parâmetro de alcance ϕ .
- $Z(x)$ é o efeito composicional.

A função de verossimilhança do modelo é

$$L(\theta; \underline{Y}) = (2\pi)^{-n/2} |\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(\underline{Y} - \underline{\mu}_Y)' \Sigma^{-1} (\underline{Y} - \underline{\mu}_Y)\right\}.$$

Fazendo-se a reparametrização: $\eta = \frac{\sigma_2}{\sigma_1}$; $\nu_1 = \frac{\tau_1}{\sigma_1}$; $\nu_2 = \frac{\tau_2}{\sigma_1}$, pode-se escrever

$$\Sigma = \sigma_1^2 R + \tau_1^2 I_b = \sigma_1^2 V,$$

e a função de log-verossimilhança reparametrizada é dada por

$$l(\theta; \underline{Y}) = -(1/2)(n \ln(2\pi) + 2n \ln(\sigma_1) + \ln(|V|) + (1/\sigma_1^2) Qe).$$

onde $Qe = (\underline{Y} - \underline{\mu}_Y)'V^{-1}(\underline{Y} - \underline{\mu}_Y)$. Os estimadores de máxima verossimilhança são:

$$\hat{\underline{\mu}} = (D'V^{-1}D)^{-1}(D'V^{-1}\underline{Y}) \quad e \quad \hat{\sigma}_1 = \sqrt{\hat{Q}e/n}.$$

As variâncias dos estimadores obtidos pelo método Delta são:

$$Var(\hat{\underline{\mu}}) = \hat{\sigma}_1^2(D'\hat{V}^{-1}D)^{-1} \quad e \quad Var(\hat{\sigma}_1) = (\hat{\sigma}_1^3/(3\hat{Q}e - n\hat{\sigma}_1)).$$

As estimativas para os parâmetros σ_2 , τ_1 , τ_2 , ϕ e ρ são obtidas numericamente.

Value

data.frame contendo A função retorna uma lista com os dois seguintes objetos:

- `comp1` data.frame contendo as estimativas dos parâmetros, erros-padrão e intervalos de 95% de confiança via método Delta.
- `comp2` data.frame contendo o negativo da log-verossimilhança inicial e otimizada.

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,
 Ana Beatriz Tozzo Martins <abtmartins@uem.br>,
 Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

- Azzalini, A., (1996). *Statistical Inference Based on the Likelihood*. Chapman and Hall, London.
- Cox, D. R., and Hinkley, D. V., (1974). *Theoretical Statistics*. Chapman and Hall, London.
- DeGroot, M. H. and Schervish, M. J. (2002). *Probability and Statistics*. 3.ed., Addison Wesley, Boston.
- Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H., (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n. 3, 456-477.
- Pawitan, Y., (2001). *In All Likelihood*. Oxford University Press, Inc, New York.

Examples

```
### Carregando pacote
```

```

require(geoComp)
### Carregando os dados
data(pivo)
### Selecionando as colunas referentes aos componentes 1, 2 e 3,
### Coordenada X e coordenada Y
dados <- pivo[,c(6,7,8,1,2)]
### Transformando o objeto "dados" para a classe geoComp
dados <- as.geoComp(dados)
### Fazendo o ajuste do modelo
estima <- mec(dados)

```

`monta.V` *Matriz de covariância.*

Description

A função constrói a matriz de covariância do modelo bivariado onde as variáveis são dispostas de forma intercalada.

Usage

```
monta.V(theta1, dados.comp)
```

Arguments

`theta1` Vetor com valores para os parâmetros η , v_1 , v_2 , ϕ e ρ .
`dados.comp` Objeto da classe `geoComp`.

Value

A função retorna a matriz de covariância do modelo.

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,
Ana Beatriz Tozzo Martins <abtmartins@uem.br>,
Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H., (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n. 3, 456-477.

Examples

```
### Carregando pacote
require(geoComp)
### Carregando dados
data(pivo)
### Selecionando dados
dados <- pivo[1:15,c(6,7,8,1,2)]
### Transformando o objeto "dados" para a classe geoComp
dados.comp <- as.geoComp(dados)
y1 <- dados.comp[[1]]$Y1
y2 <- dados.comp[[1]]$Y2
coords <- dados.comp[[3]]
### Calculando os valores iniciais
mu1 <- mean(y1)
mu2 <- mean(y2)
var_y1 <- var(y1)
s1 <- var_y1/2
tau1 <- s1
var_y2 <- var(y2)
s2 <- var_y2/2
tau2 <- s2
dim <- range(dist(coords))
phi <- dim[1] + 0.2*(dim[2]-dim[1])
rho <- cor(y1,y2)
y <- dados.comp[[2]][[1]]
### Reparametrizando
eta <- s2/s1
nu1 <- tau1/s1
nu2 <- tau2/s1
theta1 <- c(eta,nu1,nu2,phi,rho)
### Calculando a matriz de covariância
V <-monta.V(theta1,dados.comp)
```

pivo

Propriedades do solo irrigado por um sistema de pivo.

Description

Propriedades do solo irrigado por um sistema de pivo.

Usage

`data(pivo)`

Format

Um data frame contendo 76 observações das seguintes 8 variáveis:

Xum vetor numérico com as coordenadas X das observações.

Yum vetor numérico com as coordenadas Y das observações.

DenSoilnumérico, densidade do solo g/cm^3 .

DenParnumérico, densidade da partícula (%).

Porototnumérico, porosidade total (%).

Areianumérico, fração do conteúdo de areia (%).

Siltenumérico, fração do conteúdo de silte (%).

Argilanumérico, fração do conteúdo de argila (%).

Details

A densidade do solo é definida como:

$$DenSoil = \frac{m_{ss}}{V_T}$$

onde m_{ss} é a massa de solo seco e V_T é o volume total.

$$V_T = V_s + V_p,$$

sendo V_s o volume de sólidos e V_p o volume de poros. A densidade da partícula é definida como:

$$DenPar = \frac{m_{ss}}{V_s}$$

e a porosidade total é como:

$$Porotot = \frac{V_p}{V_T}$$

Source

Gonçalves, A. C. A., (1997). *Variabilidade espacial de propriedades físicas do solo para fins de manejo da irrigação*. 119p., Tese (Doutorado em Agronomia) - Escola Superior de Agricultura "Luiz de Queiroz", Universidade de São Paulo, Piracicaba.

References

Gonçalves, A. C. A., and Folegatti, M. V., and Mata, J. D. V., (2001). Análises exploratória e geoestatística da variabilidade de propriedades físicas de um Argissolo Vermelho. *Acta Scientiarum* **23**, n. 5, 1149-1157.

Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H., (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n. 3, 456-477.

Examples

```
### Carregando pacotes
require(compositions)
require(geoComp)
### Lendo um arquivos de dados
data(pivo)
### Selecionado os dados composicionais
dados <- pivo[,c(6,7,8)]
### Escala representada na classe acomp onde os dados são analisados
### na geometria relativa
comp <- acomp(dados)
### Construindo um diagrama ternário
plot(comp)
### Incluindo o centro da distribuição
plot(mean(comp),add=TRUE,pch=20,col='red')
### Incluindo as regiões 2 e 4-desvios-padrão de confiança
ellipses(mean(comp),var(comp),col='red',r=2)
ellipses(mean(comp),var(comp),col='red',r=4)
```

`posteriori`*Posteriori do modelo bayesiano para dados composicionais espaciais.*

Description

A função calcula a posteriori do modelo geoestatístico bivariado para dados composicionais.

Usage

```
posteriori(theta1, dados.comp, mediahiper, print.pars = FALSE)
```

Arguments

- `theta1` Vetor cujos elementos são os valores para os parâmetros η , v_1 , v_2 , ϕ e ρ .
- `dados.comp` Objeto da classe `geoComp`.
- `mediahiper` Vetor com 8 elementos correspondentes aos valores dos hiperparâmetros das prioris log-normais para η , v_1 , v_2 e Gama para ϕ . Os três primeiros correspondem as médias para η , v_1 , v_2 , o quarto é o hiperparâmetro α para ϕ , os três seguintes correspondem aos desvios-padrão para η , v_1 , v_2 e o último, o hiperparâmetro β para ϕ . O default é $(\log(24), \log(0.63), \log(0.59), 66, 0.3, 0.3, 0.3, 1)$.
- `print.pars` Lógico, indicando se a função retorna o vetor `theta1` e o valor da posteriori (`'print.pars = TRUE'`) ou apenas o valor da posteriori (`'print.pars = FALSE'`).

Details

$\text{posteriori} = \text{vero} \times \text{priori}(\eta) \times \text{priori}(v_1) \times \text{priori}(v_2) \times \text{priori}(\phi) \times 1$

Value

A função retorna o valor da posteriori e o vetor `theta1`.

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,

Ana Beatriz Tozzo Martins <abtmartins@uem.br>,

Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H., (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n. 3, 456-477.

Examples

```
### Carregando pacotes
require(geoComp)
require(MCMCpack)
require(MASS)
### Carregando os dados
data(pivo)
### Selecionando as colunas referentes aos componentes 1, 2 e 3,
### Coordenada X e coordenada Y
dados <- pivo[,c(6,7,8,1,2)]
### Transformando o objeto "dados" para a classe geoComp
dados <- as.geoComp(dados)
### Calculando os valores iniciais
y1 <- dados[[1]]$Y1
y2 <- dados[[1]]$Y2
coords <- dados[[3]]
mu1 <- mean(y1)
mu2 <- mean(y2)
var_y1 <- var(y1)
s1 <- var_y1/2
tau1 <- s1
var_y2 <- var(y2)
s2 <- var_y2/2
tau2 <- s2
dim <- range(dist(coords))
phi <- dim[1] + 0.2*(dim[2]-dim[1])
rho <- cor(y1,y2)
eta <- s2/s1
nu1 <- tau1/s1
nu2 <- tau2/s1
theta1 <- c(eta,nu1,nu2,phi,rho)
### Calculando a posteriori
posteriori(theta1,dados.comp=dados,mediahiper=c(),print.pars=TRUE)
```

`resultado.mh` *Resultados da inferência bayesiana.*

Description

A função organiza os resultados obtidos através do algoritmo Metropolis-Hastings e calcula os intervalos de 95% de credibilidade.

Usage

```
resultado.mh(resultado)
```

Arguments

`resultado` Lista contendo os resultados obtidos através da função `infbayes`.

Value

A função retorna uma lista contendo os seguintes elementos:

- `comp1` data frame contendo as posteriores para β_1 , β_2 , σ_1 , σ_2 , τ_1 , τ_2 , ϕ e ρ .
- `comp2` data frame contendo as esperanças a posteriori e intervalos de 95% de credibilidade para os parâmetros do modelo.
- `comp3` data frame contendo as estimativas obtidas no processo de otimização.

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,
Ana Beatriz Tozzo Martins <abtmartins@uem.br>,
Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H., (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n. 3, 456-477.

Examples

```

### Carregando pacotes
require(geoR)
require(geoComp)
require(MCMCpack)
require(MASS)
### Carregando os dados
data(pivo)
### Selecionando as colunas referentes aos componentes 1, 2 e 3,
### Coordenada X e coordenada Y
dados <- pivo[,c(6,7,8,1,2)]
### Transformando o objeto "dados" para a classe geoComp
dados <- as.geoComp(dados)
### Fazendo o ajuste do modelo por métodos bayesianos
estima.bayes <- infbayes(dados,tune=c(0.5,0.5,0.5,0.5,0.5),queima=1,salto=5,
                        nsim=20,mediahiper=c())
summary(estima.bayes[[2]])
### Organizando os resultados
res.mh <- resultado.mh(resultado=estima.bayes)
res.mh.ic <- data.frame(res.mh[[2]])
names(res.mh.ic) <- c('2.5%', 'media', '97.5%')

```

tab.coord.pol *Coordenadas do diagrama de classificação textural.*

Description

Coordenadas dos vértices no triângulo de classificação textural.

Usage

```
data(tab.coord.pol)
```

Format

Um data frame contendo 69 observações das seguintes 4 variáveis.

Areia numérico, fração do conteúdo de areia (%).

Silte numérico, fração do conteúdo de silte (%).

Argila numérico, fração do conteúdo de argila (%).

Poligonoum vetor numérico.

Examples

```
data(tab.coord.pol)
## maybe str(tab.coord.pol) ; plot(tab.coord.pol) ...
```

`volta.cokri` *Transformação de volta para o simplex por simulação.*

Description

A função transforma o vetor de valores esperados e matriz de covariância preditos no R^2 para o simplex utilizando simulação.

Usage

```
volta.cokri(mat.cokri, num.simu, retorna.tudo = FALSE, int.conf = 0.95)
```

Arguments

`mat.cokri` Uma lista cujo primeiro elemento contém o vetor de valores esperados preditos e segundo elemento contém a matriz de covariância predita.

`num.simu` Vetor numérico correspondente ao número de simulações.

`retorna.tudo` Lógico, indicando se a função deve retornar, separadamente, os valores esperados preditos por simulação para cada componente (`'retorna.tudo = TRUE'`) ou a função deve calcular as médias do número de simulações para cada localização e para cada componente os intervalos de 95% de confiança (`'retorna.tudo = FALSE'`).

`int.conf` Vetor numérico correspondente ao nível de confiança.

Details

Os dados são simulados de uma distribuição gaussiana multivariada com vetor de médias e matriz de covariância dados em `mat.cokri` e em seguida é aplicada a função `agl` transformando-os em composições.

Value

Se `retorna.tudo=TRUE`, a função retorna uma lista com os três primeiros elementos a seguir e, se `retorna.tudo=FALSE`, a função retorna uma lista com os dois últimos:

<code>comp1</code>	valores esperados preditos simulados para cada localização de predição do componente 1.
<code>comp2</code>	valores esperados preditos simulados para cada localização de predição do componente 2.
<code>comp3</code>	valores esperados preditos simulados para cada localização de predição do componente 3.
<code>comp4</code>	um data frame com 3 colunas correspondentes as médias do número de simulações para cada localização e para cada componente.
<code>comp5</code>	um data frame contendo os limites dos intervalos de confiança.

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,
 Ana Beatriz Tozzo Martins <abtmartins@uem.br>,
 Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

- Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H., (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n. 3, 456-477.
- Diggle, P. J., and Ribeiro, Jr., P. J., (2007). *Model-based geostatistics*. Springer Series in Statistics, USA.

Examples

```
### carregando pacotes
require(geoComp)
require(MASS)
```

```

require(geoR)
### Carregando os dados
data(pivo)
### Selecionando as colunas referentes aos componentes 1, 2 e 3,
### Coordenada X e Coordenada Y
dados <- pivo[,c(6,7,8,1,2)]
### Transformando o objeto "dados" para a classe geoComp
dados <- as.geoComp(dados)
### Construindo a borda da área de predição
bor <- cbind(c(0,seq(0,200,l=100),0),c(0,sqrt(200^2-seq(0,200,l=100)^2),0))
### Construindo uma grade de predição
gr <- pred_grid(bor, by=70)
### Ajustando o modelo
estima <- mec(dados)
### Obtendo os valores preditos no R^2
md.cov.ck <- cokrigagem(estima[[1]]$Estimativas, loc=gr,dados.comp=dados)
### Obtendo os valores preditos no simplex por simulação
preditos.simu <- volta.cokri(md.cov.ck,num.simu=10,int.conf=0.95)
preditos.simu <- data.frame(preditos.simu[[1]])
preditos.simu.ic <- data.frame(preditos.simu[[2]])

```

volta.quad	<i>Transformação de volta para o simplex por quadratura de Gauss-Hermite.</i>
------------	---

Description

A função transforma o vetor de valores esperados e matriz de covariância preditos no R^2 para o simplex utilizando quadratura gaussiana.

Usage

```
volta.quad(med.cov, n.pontos = 7, Variancia = FALSE)
```

Arguments

med.cov	Uma lista cujo primeiro elemento contém o vetor de valores esperados preditos e segundo elemento contém a matriz de covariância predita.
---------	--

n.pontos	Vetor numérico correspondente a ordem da quadratura de Gauss-Hermite.
Variância	Lógico, indicando se a função não deve calcular a matriz de covariância no espaço amostral simplex ('Variância = FALSE') ou se deve calcular ('Variância = TRUEf')

Details

O objetivo é calcular para cada localização uma estimativa de

$$\mu_X = \int_{S^B} X f(X) dX \quad e \quad \Sigma_X = \int_{S^B} (X - \mu_X)(X - \mu_X)' f(X) dX.$$

De acordo com Aitchison (1986), Pawlowsky-Glahn e Olea (2004) as integrais são expressas como

$$\mu_X = \int_{R^{B-1}} g_1(Z) f(-Z'Z) dZ \quad e \quad \Sigma_X = \int_{R^{B-1}} g_2(Z) f(-Z'Z) dZ$$

em que Z é a transformação

$$Z = \frac{1}{\sqrt{2}}(R')^{-1}(alr(X) - \mu_Y);$$

R é a decomposição Cholesky de Σ_Y , uma matriz triangular superior; e as funções $g_1(Z)$ e $g_2(Z)$ são da forma

$$g_1(Z) = \pi^{-\frac{B-1}{2}} agl(\mu_Y + \sqrt{2}R'Z),$$

e

$$g_2(Z) = \pi^{-\frac{B-1}{2}} (agl(\mu_Y + \sqrt{2}R'Z) - \mu_X)(agl(\mu_Y + \sqrt{2}R'Z) - \mu_X)'$$

Desta forma, as integrais são aproximadas pela integração de Gauss-Hermite multivariada de ordem k :

$$\int_{R^{B-1}} g(Z) f(-Z'Z) dZ \approx \sum_{i_1=1}^k \sum_{i_2=1}^k \cdots \sum_{i_{B-1}=1}^k \omega_{i_1} \omega_{i_2} \cdots \omega_{i_{B-1}} g(Z_{i_1}, Z_{i_2}, \dots, Z_{i_{B-1}}),$$

em que os pesos $\omega_{i_1} \omega_{i_2} \cdots \omega_{i_{B-1}}$ e as abscissas $Z_{i_1}, Z_{i_2}, \dots, Z_{i_{B-1}}$ podem ser encontrados, por exemplo, em Abramowitz e Stegun (1992) ou calculados usando a função `gauss.quad` do pacote `statmod`.

Value

A função retorna uma lista com os seguintes elementos:

comp1 valores esperados preditos dos componentes 1, 2 e 3 no simplex.

comp2 matriz de covariância predita dos componentes 1, 2 e 3 no simplex.

Author(s)

Wagner Hugo Bonat <wagner@leg.ufpr.br>,
 Ana Beatriz Tozzo Martins <abtmartins@uem.br>,
 Paulo Justiniano Ribeiro Jr. <paulojus@leg.ufpr.br>.

References

- Abramowitz, M., and Stegun, I. A., (1972). *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Milton Abramowitz and Irene A. Stegun, Washington.
- Aitchison, J., (1986). *The Statistical analysis of compositional data*. The Blackburn Press, New Jersey.
- Lark, R. M, and Bishop, T. F. A., (2007). Cokriging particle size fractions of the soil. *European Journal of Soil Science*, june, **58**, 763-774.
- Martins, A. B. T., and Ribeiro Jr, P. J., and Bonat, W. H., (2009). Um modelo geoestatístico para dados composicionais. *Revista Brasileira de Biometria* **27**, n. 3, 456-477.
- Paulino, C. D., and Turkman, M. A. A., and Murteira, B., (2003). *Estatística Bayesiana*. Fundação Calouste Gulbenkian, Lisboa.
- Pawłowsky-Glahn, V., and Olea, R. A., (2004). *Geostatistical Analysis of Compositional Data*. Oxford University Press, Inc, New York.

Examples

```
### carregando pacotes
require(geoComp)
require(MASS)
require(statmod)
require(geoR)
### Carregando os dados
data(pivo)
### Selecionando as colunas referentes aos componentes 1, 2 e 3,
### Coordenada X e coordenada Y
dados <- pivo[,c(6,7,8,1,2)]
### Transformando o objeto "dados" para a classe geoComp
dados <- as.geoComp(dados)
### Construindo a borda da área de predição
```

```
bor <- cbind(c(0,seq(0,200,l=100),0),c(0,sqrt(200^2-seq(0,200,l=100)^2),0))
### Construindo uma grade de prediçãõ
gr <- pred_grid(bor, by=70)
### Ajustando o modelo
estima <- mec(dados)
### Obtendo os valores preditos no R^2
md.cov.ck <- cokrigagem(estima[[1]]$Estimativas, loc=gr,dados.comp=dados)
### Obtendo os valores preditos no simplex por quadratura gaussiana
preditos.gh <- volta.quad(md.cov.ck,n.pontos=6,Variancia=FALSE)
preditos.gh <- data.frame(preditos.gh)
```

ANEXO C – Código fonte R das análises estatísticas

Scripts para o estudo de simulação:

Arquivo **simula.R**

```
## Função para simular um conjunto de dados
simu.dados <- function(theta,n.simul){
  mu1 <- theta[1]
  mu2 <- theta[2]
  s1 <- theta[3]
  s2 <- theta[4]
  tau1 <- theta[5]
  tau2 <- theta[6]
  phi <- theta[7]
  rho <- theta[8]

  ## Gerando erros de uma normal bivariada com mu=0,
  ## variâncias tau1, tau2 e correlação rho
  v <- matrix(c(tau1^2,tau1*tau2*rho,tau1*tau2*rho,tau2^2),2,2)
  Sc <- kronecker(diag(n.simul/2),v)
  set.seed(3333)
  e <- mvrnorm(n.simul,mu=c(0,0),v)
  ## Gerando um campo aleatório gaussiano com média 0, variância 1 e
  ## parâmetros de alcance phi
  set.seed(3333)
  U <- grf(n.simul,grid='irreg',cov.model='exp',cov.pars=c(1,phi),
          mean=0,messages=FALSE)
  S1 <- s1*U$data
  S2 <- s2*U$data
  ## Calculando e organizando os dados
  y1 <- as.geodata(data.frame(U$coords[,1],U$coords[,2],mu1+S1+e[,1]))
  y2 <- as.geodata(data.frame(U$coords[,1],U$coords[,2],mu2+S2+e[,2]))
```

```

seq1 <- seq(1,length(y1$data)*2,by=2)
seq2 <- seq(2,length(y2$data)*2,by=2)
y <- c()
y[seq1] <- y1$data
y[seq2] <- y2$data
coords <- data.frame(U$coords[,1],U$coords[,2])
## Preparando a saída
retorna <- list()
retorna[1][[1]] <- y1$data
retorna[2][[1]] <- y2$data
retorna[3][[1]] <- y
retorna[4][[1]] <- coords
return(retorna)
}

```

Arquivo branco1.R

```

## Script rodado para a 1a configuração de diagrama ternário
## Gera valores preditos por quadratura e simulação pra fazer mapas
require(MASS)
require(geoR)
require(geoComp)
require(compositions)
require(statmod)
source("simula.R")
## Função que gera diagrama ternário
gerater <- function(theta,n.simul=100){
  dados <- simu.dados(theta,n.simul=n.simul)
  y1 <- dados[1][[1]]
  y2 <- dados[2][[1]]
  y12 <- data.frame(y1,y2)
  comp <- agl(y12)
  comp1 <- acomp(comp)
  names(comp1) <- c("Areia", "Silte", "Argila")
  plot(comp1)
  plot(mean(comp1),add=T,pch=20,col='red')
  ellipses(mean(comp1),var(comp1),col='red',r=2)
  ellipses(mean(comp1),var(comp1),col='red',r=4)
  retorna <- data.frame(comp,dados[[4]][,1],dados[[4]][,2])
}

```

```

return(retorna)
}
##### 1a configuração #####
theta <- c(-0.2,-0.5,1,1.5,0.3,0.3,0.6,0.9)
#set.seed(1333)
dados1 <- gerater(theta,n.simul=100)
names(dados1) <- c("Areia","Silte","Argila","Coord.X", "Coord.Y")
write.table(dados1,"dad_conf1_by0.03k7ns1000.txt")
dados1 <- as.geoComp(dados1)
write.table(dados1[[3]],"points1_by0.03k7ns1000.txt")
estima <- mec(dados1)
write.table(estima[[1]],"estima1par_by0.03k7ns1000.txt")
write.table(estima[[2]],"estima1log_by0.03k7ns1000.txt")
borda <- cbind(c(0,0,1,1,0),c(0,1,1,0,0))
gr <- pred_grid(borda, by=0.03)
source("cokrigagem.R")
md.cov.ck <- cokrigagem(estima[[1]]$Estimativas,loc=gr,
                        dados.comp=dados1)
preditos.gh <- volta.quad(md.cov.ck,n.pontos=7,Variancia=FALSE)
write.table(preditos.gh,"dt1predgh_by0.03k7ns1000.txt")
preditos.simu <- volta.cokri(md.cov.ck,num.simu=1000,int.conf=0.95)
write.table(preditos.simu,"dt1predsimu_by0.03k7ns1000.txt")

```

Arquivo **branco2.R**:

```

## Script rodado para a 2a configuração de diagrama ternário
## Gera valores preditos por quadratura e simulação pra fazer mapas
require(MASS)
require(geoR)
require(geoComp)
require(compositions)
require(statmod)
source("simula.R")
## Função que gera diagrama ternário
gerater <- function(theta,n.simul=100){
  dados <- simu.dados(theta,n.simul=n.simul)
  y1 <- dados[1][[1]]
  y2 <- dados[2][[1]]
  y12 <- data.frame(y1,y2)

```

```

comp <- agl(y12)
comp1 <- acomp(comp)
names(comp1) <- c("Areia", "Silte", "Argila")
plot(comp1)
plot(mean(comp1),add=T,pch=20,col='red')
ellipses(mean(comp1),var(comp1),col='red',r=2)
ellipses(mean(comp1),var(comp1),col='red',r=4)
retorna <- data.frame(comp,dados[[4]][,1],dados[[4]][,2])
return(retorna)
}
##### 2a configuração #####
theta <- c(1,1,1.2,1.5,0.9,1,0.6,0.5)
#set.seed(1333)
dados1 <- gerater(theta,n.simul=100)
names(dados1) <- c("Areia","Silte","Argila","Coord.X", "Coord.Y")
write.table(dados1,"dad_conf2_by0.03k7ns1000.txt")
dados1 <- as.geoComp(dados1)
write.table(dados1[[3]],"points2_by0.03k7ns1000.txt")
estima <- mec(dados1)
write.table(estima[[1]],"estima2par_by0.03k7ns1000.txt")
write.table(estima[[2]],"estima2log_by0.03k7ns1000.txt")
borda <- cbind(c(0,0,1,1,0),c(0,1,1,0,0))
gr <- pred_grid(borda, by=0.03)
source("cokrigagem.R")
md.cov.ck <- cokrigagem(estima[[1]]$Estimativas,loc=gr,
                        dados.comp=dados1)
preditos.gh <- volta.quad(md.cov.ck,n.pontos=7,Variancia=FALSE)
write.table(preditos.gh,"dt2predgh_by0.03k7ns1000.txt")
preditos.simu <- volta.cokri(md.cov.ck,num.simu=1000,int.conf=0.95)
write.table(preditos.simu,"dt2predsimu_by0.03k7ns1000.txt")

```

Arquivo branco3.R

```

## Script rodado para a 3a configuração de diagrama ternário
## Gera valores preditos por quadratura e simulação pra fazer mapas
require(MASS)
require(geoR)
require(geoComp)
require(compositions)

```

```

require(statmod)
source("simula.R")
## Função que gera diagrama ternário
gerater <- function(theta,n.simul=100){
  dados <- simu.dados(theta,n.simul=n.simul)
  y1 <- dados[1][[1]]
  y2 <- dados[2][[1]]
  y12 <- data.frame(y1,y2)
  comp <- agl(y12)
  comp1 <- acomp(comp)
  names(comp1) <- c("Areia", "Silte", "Argila")
  plot(comp1)
  plot(mean(comp1),add=T,pch=20,col='red')
  ellipses(mean(comp1),var(comp1),col='red',r=2)
  ellipses(mean(comp1),var(comp1),col='red',r=4)
  retorna <- data.frame(comp,dados[[4]][,1],dados[[4]][,2])
  return(retorna)
}
##### 3a configuração #####
theta <- c(-0.2,-1,0.45,0.13,0.3,0.3,0.6,0.95)
set.seed(1333)
dados1 <- gerater(theta,n.simul=100)
names(dados1) <- c("Areia","Silte","Argila","Coord.X", "Coord.Y")
write.table(dados1,"dad_conf3_by0.03k7ns1000.txt")
dados1 <- as.geoComp(dados1)
write.table(dados1[[3]],"points3_by0.03k7ns1000.txt")
estima <- mec(dados1)
write.table(estima[[1]],"estima3par_by0.03k7ns1000.txt")
write.table(estima[[2]],"estima3log_by0.03k7ns1000.txt")
borda <- cbind(c(0,0,1,1,0),c(0,1,1,0,0))
gr <- pred_grid(borda, by=0.03)
source("cokrigagem.R")
md.cov.ck <- cokrigagem(estima[[1]]$Estimativas,loc=gr,
                      dados.comp=dados1)
preditos.gh <- volta.quad(md.cov.ck,n.pontos=7,Variancia=FALSE)
write.table(preditos.gh,"dt3predgh_by0.03k7ns1000.txt")
preditos.simu <- volta.cokri(md.cov.ck,num.simu=1000,int.conf=0.95)
write.table(preditos.simu,"dt3predsimu_by0.03k7ns1000.txt")

```

Scripts para o estudo dos intervalos de cobertura:

Arquivo **simula.R**

```
## Função para simular um conjunto de dados
simu.dados <- function(theta,n.simul){
  mu1 <- theta[1]
  mu2 <- theta[2]
  s1 <- theta[3]
  s2 <- theta[4]
  tau1 <- theta[5]
  tau2 <- theta[6]
  phi <- theta[7]
  rho <- theta[8]

  ## Gerando erros de uma normal bivariada com mu=0,
  ## variâncias tau1, tau2 e correlação rho
  v <- matrix(c(tau1^2,tau1*tau2*rho,tau1*tau2*rho,tau2^2),2,2)
  Sc <- kronecker(diag(n.simul/2),v)
  e <- mvrnorm(n.simul,mu=c(0,0),v)

  ## Gerando um campo aleatório gaussiano com média 0, variância 1
  ## e parâmetros de alcance phi
  U <- grf(n.simul,grid='irreg',cov.model='exp',cov.pars=c(1,phi),
          mean=0,messages=FALSE)

  S1 <- s1*U$data
  S2 <- s2*U$data

  ## Calculando e organizando os dados
  y1 <- as.geodata(data.frame(U$coords[,1],U$coords[,2],mu1+S1+e[,1]))
  y2 <- as.geodata(data.frame(U$coords[,1],U$coords[,2],mu2+S2+e[,2]))
  seq1 <- seq(1,length(y1$data)*2,by=2)
  seq2 <- seq(2,length(y2$data)*2,by=2)
  y <- c()
  y[seq1] <- y1$data
  y[seq2] <- y2$data
  coords <- data.frame(U$coords[,1],U$coords[,2])

  ## Preparando a saída
  retorna <- list()
  retorna[1][[1]] <- y1$data
  retorna[2][[1]] <- y2$data
  retorna[3][[1]] <- y
}
```



```

  retorna[4][[1]] <- coords
  return(retorna)
}

```

Arquivo milSimulacoes.R

```

simulando <- function(theta,n.simu=1000,n.amostra=300,
                      metodo="L-BFGS-B"){
  MATRIZ <- matrix(nrow=n.simu,ncol=27)
  dados <- list()
  assign("dados", dados, envir = .GlobalEnv)
  X <- cbind(rep(1:0,length=n.amostra*2),rep(0:1,length=n.amostra*2))
  assign("X", X, envir = .GlobalEnv)
  y <- c()
  assign("y", y, envir = .GlobalEnv)
  for( i in 1:n.simu){
    sim <- simu.dados(theta,n.simul=n.amostra)# n.simul=n. de
                                                # observações simuladas

    y1 <- sim[1][[1]]
    y2 <- sim[2][[1]]
    y <- sim[3][[1]]
    assign("y",y, envir= .GlobalEnv)
    coords <- sim[4][[1]]
    dados[1][[1]] <- sim[3][[1]]
    dados[2][[1]] <- sim[4][[1]]
    assign("dados", dados, envir = .GlobalEnv)
    estimado <- try(mec(y1,y2,X,coords=coords,metodo=metodo,
                      otimizador="optim"))
    recurso <- list()
    recurso[[1]] <- data.frame(matrix(ncol=5,nrow=8))
    recurso[[2]] <- c(1,1,1)
    if(class(estimado) == "try-error"){
      est <- recurso
    }
    if(class(estimado) != "try-error"){
      est <- estimado
    }
    print(est)
    t1 <- as.matrix(est[[1]][c(2,4,5)])[1,]
  }
}

```

```

t2 <- as.matrix(est[[1]][c(2,4,5)])[2,]
t3 <- as.matrix(est[[1]][c(2,4,5)])[3,]
t4 <- as.matrix(est[[1]][c(2,4,5)])[4,]
t5 <- as.matrix(est[[1]][c(2,4,5)])[5,]
t6 <- as.matrix(est[[1]][c(2,4,5)])[6,]
t7 <- as.matrix(est[[1]][c(2,4,5)])[7,]
t8 <- as.matrix(est[[1]][c(2,4,5)])[8,]
linha <- c(matrix(c(t1,t2,t3,t4,t5,t6,t7,t8,est[[2]][1],est[[2]][2],
                    est[[2]][3]),nrow=1))
MATRIZ[i,] <- linha
}
MATRIZ <- data.frame(MATRIZ)
names(MATRIZ) <- c("mu1","Lmin.mu1","Lmax.mu1","mu2","Lmin.mu2",
                  "Lmax.mu2","s1","Lmin.s1","Lmax.s1","s2" ,
                  "Lmin.s2","Lmax.s2","t1","Lmin.t1","Lmax.t1",
                  "t2","Lmin.t2","Lmax.t2" ,"phi","Lmin.phi",
                  "Lmax.phi","rho","Lmin.rho","Lmax.rho","logvero",
                  "logvero.est","conv")
return(MATRIZ)
}

```

Arquivo **LBFGBS.R**

```

## Carregando pacotes
require(MASS)
require(geoR)
require(geoComp)
## Carregando funções
source("simula.R")
source("milSimulacoes.R")
## Usando a função simulando
theta <- c(50,70,2.5,3.5,2.5,3.5,0.6,0.5)
LBFGBS <- simulando(theta,n.simu=1000,n.amostra=300,metodo="L-BFGS-B")
write.table(LBFGBS,"EstudoSimns1000na300.txt")

```

Gráficos referentes aos intervalos de cobertura

```

## Gráficos de densidades à posteriori
dad.sim.na300ns1000 <- read.table('EstudoSimns1000na300.txt')

```

```

par(mfrow=c(2,5),mar=c(3.5,3.5,1.5,1.5),mgp=c(1.6,0.6,0))
hist(dad.sim.na300ns1000$mu1,prob=T,main=' ',xlim=c(44,56),
      ylim=c(0,0.25),xlab=('mu1 estimado'),ylab=('Densidade'))
lines(density(dad.sim.na300ns1000$mu1),col=2)
hist(dad.sim.na300ns1000$mu2,prob=T,main=' ',xlim=c(60,80),
      ylim=c(0,0.20),xlab=('mu2 estimado'),ylab=('Densidade'))
lines(density(dad.sim.na300ns1000$mu2),col=2)
hist(dad.sim.na300ns1000$s1,prob=T,main=' ',xlim=c(0,5),
      ylim=c(0,0.9),xlab=('s1 estimado'),ylab=('Densidade'))
lines(density(dad.sim.na300ns1000$s1),col=2)
hist(dad.sim.na300ns1000$s2,prob=T,main=' ',xlim=c(0,8),
      ylim=c(0,0.6),xlab=('s2 estimado'),ylab=('Densidade'))
lines(density(dad.sim.na300ns1000$s2),col=2)
hist(dad.sim.na300ns1000$t1,prob=T,main=' ',xlab=('t1 estimado'),
      ylab=('Densidade'))
lines(density(dad.sim.na300ns1000$t1),col=2)
hist(dad.sim.na300ns1000$t2,prob=T,main=' ',xlim=c(2,5),
      ylim=c(0,2.5),xlab=('t2 estimado'),ylab=('Densidade'))
lines(density(dad.sim.na300ns1000$t2),col=2)
hist(dad.sim.na300ns1000$phi,prob=T,main=' ',ylim=c(0,3),
      xlab=('phi estimado'),ylab=('Densidade'))
lines(density(dad.sim.na300ns1000$phi),col=2)
hist(dad.sim.na300ns1000$rho,prob=T,main=' ',xlim=c(0.2,0.7),
      ylim=c(0,10),xlab=('rho estimado'),ylab=('Densidade'))
lines(density(dad.sim.na300ns1000$rho),col=2)
hist(dad.sim.na300ns1000$logvero,prob=T,main=' ',xlim=c(1450,1700),
      ylim=c(0,0.02),xlab=('logvero'),ylab=('Densidade'))
lines(density(dad.sim.na300ns1000$logvero),col=2)
hist(dad.sim.na300ns1000$logvero.est,prob=T,main=' ',xlim=c(-1560,-1420),
      ylim=c(0,0.025),xlab=('logvero estimada'),ylab=('Densidade'))
lines(density(dad.sim.na300ns1000$logvero.est),col=2)
## Boxplots das posteriores dos parâmetros
par(mfrow=c(2,4),mar=c(3.5, 3.5, 1.5, 0.5), mgp=c(1.8,.8, 0))
boxplot(dad.sim.na300ns1000$mu1,main='mu1',ylab="Estimativas")
boxplot(dad.sim.na300ns1000$mu2,main='mu2',ylab="Estimativas")
boxplot(dad.sim.na300ns1000$s1,main='s1',ylab="Estimativas")
boxplot(dad.sim.na300ns1000$s2,main='s2',ylab="Estimativas")
boxplot(dad.sim.na300ns1000$t1,main='tau1',ylab="Estimativas")

```

```

boxplot(dad.sim.na300ns1000$t2,main='tau2',ylab="Estimativas")
boxplot(dad.sim.na300ns1000$phi,main='phi',ylab="Estimativas")
boxplot(dad.sim.na300ns1000$rho,main='rho',ylab="Estimativas")

```

Script para a obtenção dos valores preditos utilizando os dados do pivo:

```

require(geoComp)
require(MASS)
require(statmod)
require(geoR)
data(pivo)
dados <- pivo[,c(6,7,8,1,2)]
dados <- as.geoComp(dados)
bor <- cbind(c(0,seq(0,200,l=100),0),
             c(0,sqrt(200^2-seq(0,200,l=100)^2),0))
estima <- mec(dados)
gr <- pred_grid(bor, by=4)
md.cov.ck <- cokrigagem(estima[[1]]$Estimativas, loc=gr,
                      dados.comp=dados)
preditos.gh <- volta.quad(md.cov.ck,n.pontos=7,Variancia=FALSE)
preditos.gh <- data.frame(preditos.gh)
write.table(preditos.gh,"pred_by4k7ns1000MBM.txt")
preditos.simu <- volta.cokri(md.cov.ck,num.simu=1000,int.conf=0.95)
preditos.simu <- data.frame(preditos.simu[[1]])
preditos.simu.ic <- data.frame(preditos.simu[[2]])
write.table(preditos.simu,"predsimu_by4k7ns1000MBM.txt")
write.table(preditos.simu.ic,"predsimuic_by4k7ns1000MBM.txt")
## Fazendo os mapas: Quadratura de Gauss-Hermite e Simulação
require(geoR)
require(sp) # palhete de cores boa tanto coloridas como
            # preta/branca do pacote sp:
            # bpy.colors()(bpy = blue pink yellow).
            # Anterior: col=terrain.colors(21)
bor <- data.frame(cbind(c(0,seq(0,200,l=100),0),
                      c(0,sqrt(200^2-seq(0,200,l=100)^2),0)))
gr <- pred_grid(bor, by=4)
predi.gh.MBM <- read.table("predgh_by4k7ns1000MBM.txt", header=T)
predi.simu.MBM <- read.table("predsimu_by4k7ns1000MBM.txt",header=T)
par(mfrow=c(2,3), mar=c(3,3,0.5,0.5), mgp=c(1.7,0.7, 0))

```

```

xleg <- c(85, 200); yleg <- c(185, 200)
image(structure(list(predict=predi.gh.MBM$Areia), class="kriging"),
      loc=gr,bor, col=bpy.colors(15),xlab='',ylab='',
      x.leg=xleg, y.leg=yleg,ylim=c(0,220))
text(x=165,y=170,"(A)Areia/QG")
image(structure(list(predict=predi.gh.MBM$Silte), class="kriging"),
      loc=gr,bor, col=bpy.colors(15),main="Silte",xlab='',ylab='',
      x.leg=xleg, y.leg=yleg,ylim=c(0,220))
text(x=165,y=170,"(B)Silte/QG")
image(structure(list(predict=predi.gh.MBM$Argila), class="kriging"),
      loc=gr,bor, col=bpy.colors(15),main="Argila",xlab='',ylab='',
      x.leg=xleg, y.leg=yleg,ylim=c(0,220))
text(x=165,y=170,"(C)Argila/QG")
image(structure(list(predict=predi.simu.MBM$Areia), class="kriging"),
      loc=gr,bor, col=bpy.colors(15),xlab='',ylab='',
      x.leg=xleg, y.leg=yleg,ylim=c(0,220))
text(x=165,y=170,"(D)Areia/Sim")
image(structure(list(predict=predi.simu.MBM$Silte), class="kriging"),
      loc=gr,bor, col=bpy.colors(15),main="Silte",xlab='',ylab='',
      x.leg=xleg, y.leg=yleg,ylim=c(0,220))
text(x=165,y=170,"(E)Silte/Sim")
image(structure(list(predict=predi.simu.MBM$Argila), class="kriging"),
      loc=gr,bor, col=bpy.colors(15),main="Argila",xlab='',ylab='',
      x.leg=xleg, y.leg=yleg,ylim=c(0,220))
text(x=165,y=170,"(F)Argila/Sim")

```

Scripts para o estudo da verossimilhança perfilhada:

Perfilhada para ϕ

Arquivo **monta.V.phi.R**

```

## theta1=(eta,nu1,nu2,rho)
monta.V.phi <-function(theta1, phi, dados.comp){
  y <- dados.comp[[2]][[1]]
  eta <- theta1[1]
  nu1 <- theta1[2]
  nu2 <- theta1[3]
  rho <- theta1[4]
  phi <- phi
  ## Calculando os elementos da matriz de correlação espacial

```

```

distancia <- unname(as.matrix(dist(dados.comp[[3]],diag=TRUE,
                                upper=TRUE)))
correla <- exp(-distancia/phi)
## Organizando a matriz de correlação espacial com eltos y1 e y2
## intercalados. Ou seja, calculando R1
coluna1 <- rep(c(1, eta),length(y)/2)
coluna2 <- rep(c(eta,eta^2),length(y)/2)
seq1 <- seq(1,length(y),by=2)
seq2 <- seq(2,length(y),by=2)
ERRE1 <- matrix(ncol=length(y),nrow=length(y))
ERRE1[,c(seq1)] <- coluna1
ERRE1[,c(seq2)] <- coluna2
## Organizando a matriz de correlação espacial com eltos y1 e y2
## intercalados
ERRE2 <- matrix(nrow=length(y),ncol=length(y))
for(i in 1:length(y)/2){
  ERRE2[seq1[i],] <- rep(correla[i,],each=2)
  ERRE2[seq2[i],] <- rep(correla[i,],each=2)
}
## Calculando a matriz R
ERRE <- ERRE1*ERRE2
## Calculando a matriz de covariância composicional
Ib1 <- matrix(c(nu1^2,nu1*nu2*rho,nu1*nu2*rho,nu2^2),nc=2)
Ib <- kronecker(diag(length(y)/2),Ib1)
## Matriz de covariância espacial composicional
V <- ERRE + Ib
return(V)
}

```

Arquivo **log.vero.phi.R**

```

## theta1=(eta,nu1,nu2,rho)
log.vero.phi <-function(theta1,phi,dados.comp,print.pars=F){
  if(theta1[1] <= 0) return(.Machine$double.xmax^0.5)
  if(theta1[2] <= 0) return(.Machine$double.xmax^0.5)
  if(theta1[3] <= 0) return(.Machine$double.xmax^0.5)
  if(abs(theta1[4]) > 1) return(.Machine$double.xmax^0.5)
  y <- dados.comp[[2]][[1]]
  X <- cbind(rep(1:0,length=length(y)),rep(0:1,length=length(y)))
}

```

```

V <- monta.V.phi(theta1=theta1,phi=phi ,dados.comp = dados.comp)
ldetV <- determinant(V,log=TRUE)$modulus[1]
mu <- drop(solve(crossprod(X,solve(V,X))) %*%crossprod(X,solve(V,y)))
Qe <- drop(crossprod(y,solve(V,y))-2*crossprod(y,solve(V,X%*%mu))+
           crossprod(mu,crossprod(X,solve(V,X%*%mu))))
if(Qe < 0) return(Qe=.Machine$double.xmax^0.5)
n <- length(y)
s1 <- sqrt(Qe/n)
ll <- drop(-0.5*(n*log(2*pi)+n*log(s1^2)+ ldetV + (1/(s1^2))*Qe))
if(print.pars) print(c(theta1,ll))
return(-ll)
}

```

Arquivo **perf.phi.R**

```

## Obtendo valores iniciais para theta1 = (eta, nu1, nu2, rho)
perf.phi <- function(dados.comp,tamanho, # Tamanho vetor phi pra perfilhar
                    min.delta, # Qto % menor que o mínimo dado pelo delta
                    max.delta, # Qto % maior que o máximo
                    alpha){
  var_y1 <- var(dados.comp[[1]][1])
  s1 <- var_y1/2
  tau1 <- s1
  var_y2 <- var(dados.comp[[1]][2])
  s2 <- var_y2/2
  tau2 <- s2
  eta <- s2/s1
  nu1 <- tau1/s1
  nu2 <- tau2/s1
  rho <- cor(dados[[1]][1],dados[[1]][2])
  theta1 <- c(eta,nu1,nu2,rho)
  ## Fazendo a maximização
  source("monta.V.phi.R")
  source("log.vero.phi.R")
  estima <- mec(dados.comp=dados.comp)
  ## ic_i.delta.phi = limite inferior para phi pelo método delta
  ## ic_s.delta.phi = limite superior para phi pelo método delta
  ic_i.delta.phi <- estima[[1]]$LI.Delta[7]
  ic_s.delta.phi <- estima[[1]]$LS.Delta[7]

```

```

phi <- seq(ic_i.delta.phi*min.delta,ic_s.delta.phi*max.delta,l=tamanho)
lista.res <- list()
for(i in 1:length(phi)){
  teste <- optim(theta1, log.vero.phi, phi=phi[i],
                dados.comp=dados.comp, method="L-BFGS-B",
                lower=c(1e-32,1e-32,1e-32,-1),
                upper=c(Inf,Inf,Inf,1), hessian=TRUE)
  ## Todos os resultados para cada valor de phi
  lista.res[i][[1]] <- teste
}
perf.phi <- c()
for(i in 1:length(phi)){
  perf.phi[i] <- lista.res[[i]][2]$value
}
## Gráfico da perfilhada: phi versus log-verossimilhança
#postscript('perf.phi.eps')
#plot(phi,perf.phi,ylab='Log-verossimilhanca Perfilhada')
#dev.off()
maximo <- -estima[[2]][2]
deviance.phi <- 2*(perf.phi-maximo)
# plot(phi,deviance.phi,ylab='Deviance',type="l")
# abline(h=qchisq(0.835,df=1))
val.phi.perf.dev <- data.frame(phi,perf.phi,deviance.phi)
intervalo.phi <- val.phi.perf.dev[which(val.phi.perf.dev$deviance.phi<
                                     qchisq(alpha,df=1)),]$phi
retorna <- list()
retorna[[1]] <- val.phi.perf.dev
retorna[[2]] <- intervalo.phi
return(retorna)
}

```

Arquivo **script.perfilhada.phi.R**

```

rm(list=ls(all=T))
## Carregando o arquivo de dados do geoComp
source('pivo.R')
##data(pivo)
pivo
## Selecionando os componentes e as coordenadas

```



```

dados <- pivo[,c(6,7,8,1,2)]
#head(dados)
## Convertendo o arquivo para o formato geoComp
## Transformacao ALR ( denominador=argila )
source('as.geoComp.R')
dados <- as.geoComp(dados)
source('monta.V.R')
source('logarit.vero.R')
source('mec.R')
dados.comp <- dados
estima <- mec(dados)
## Fazendo a perfilhada para phi com alpha=0.566
source('perf.phi.R')
perfilhada.phi <-perf.phi(dados,tamanho=150,min.delta=0.8,max.delta=5,
                        alpha=0.566)
perfilhada.phi
## Arquivo contendo valores de phi, perf.phi e deviance.phi
perfilhada.phi.1 <- perfilhada.phi[[1]]
write.table(perfilhada.phi.1,'dataframePerfphi0566p150.txt')
## Arquivo contendo todos os valores para os quais
## deviance<qchisq(alpha,df=1))
perfilhada.phi.2 <- perfilhada.phi[[2]]
write.table(perfilhada.phi.2 , 'IC.Perf.phi.0566p150.txt')
## Fazendo a perfilhada para phi com alpha=0.835
perfilhada.phi.3 <-perf.phi(dados,tamanho=150,min.delta=0.8,max.delta=5,
                        alpha=0.835)
## Arquivo contendo valores de phi, perf.phi e deviance.phi
perfilhada.phi.4 <- perfilhada.phi.3[[1]]
write.table(perfilhada.phi.4,'dataframePerfphi0835p150.txt')
## Arquivo contendo todos os valores para os quais
## deviance<qchisq(alpha,df=1))
perfilhada.phi.5 <- perfilhada.phi.3[[2]]
write.table(perfilhada.phi.5 , 'IC.Perf.phi.0835p150.txt')
## Lendo os resultados
dad.per.phi.0566 <- read.table('dataframePerfphi0566p150.txt')
dad.ic.phi.0566 <- read.table('IC.Perf.phi.0566p150.txt')
dad.ic.phi.0835 <- read.table('IC.Perf.phi.0835p150.txt')
## Organizando os intervalos

```

```

dad.ic.phi.0566 <- data.frame(dad.ic.phi.0566[1,],dad.ic.phi.0566[41,])
names(dad.ic.phi.0566) <- c('Li.phi.0566','Ls.phi.0566')
dad.ic.phi.0835 <- data.frame(dad.ic.phi.0835[1,],dad.ic.phi.0835[141,])
names(dad.ic.phi.0835) <- c('Li.phi.0835','Ls.phi.0835')
## Gráficos das perfilhadas: phi versus log-verossimilhanca
postscript('perfphi0566.eps')
par(mfrow=c(1,1),mar=c(3,3,1.5,1.5),mgp=c(1.8,0.8,0))
plot(dad.per.phi.0566$phi,-dad.per.phi.0566$perf.phi,xlab=expression(phi),
      ylab="Log-verossimilhanca Perfilhada",type='l')
dev.off()
#postscript('perfphi0835.eps')
#plot(dad.per.phi.0835$phi,-dad.per.phi.0835$perf.phi,xlab=expression(phi),
      ylab="Log-verossimilhanca Perfilhada",type='l')
#dev.off()
# Gráficos das Deviances: phi versus Deviance
postscript('deviancephi060e0835.eps')
par(mfrow=c(1,1),mar=c(3,3,1.5,1.5),mgp=c(1.8,0.8,0))
plot(dad.per.phi.0566$phi,dad.per.phi.0566$deviance.phi,xlab=expression(phi),
      ylab='Deviance',type="l")
segments(x0=dad.ic.phi.0566$Li.phi.0566,x1=dad.ic.phi.0566$Ls.phi.0566,
          y0=qchisq(0.566,df=1),y1=qchisq(0.566,df=1))
arrows(x0=dad.ic.phi.0566$Li.phi.0566,y0=qchisq(0.566,df=1),
        x1=dad.ic.phi.0566$Li.phi.0566,y1=0)
arrows(x0=dad.ic.phi.0566$Ls.phi.0566,y0=qchisq(0.566,df=1),
        x1=dad.ic.phi.0566$Ls.phi.0566,y1=0)
text(x=70,y=-0.1,label=as.character(round(dad.ic.phi.0566$Li.phi.0566,2)),
      cex=0.8)
text(x=dad.ic.phi.0566$Ls.phi.0566,y=-0.1,
      label=as.character(round(dad.ic.phi.0566$Ls.phi.0566,2)),cex=0.8)
segments(x0=dad.ic.phi.0835$Li.phi.0835,x1=dad.ic.phi.0835$Ls.phi.0835,
          y0=qchisq(0.835,df=1),y1=qchisq(0.835,df=1))
arrows(x0=dad.ic.phi.0835$Li.phi.0835,y0=qchisq(0.835,df=1),
        x1=dad.ic.phi.0835$Li.phi.0835,y1=0)
arrows(x0=dad.ic.phi.0835$Ls.phi.0835,y0=qchisq(0.835,df=1),
        x1=dad.ic.phi.0835$Ls.phi.0835,y1=0)
text(x=23,y=-0.1,label=as.character(round(dad.ic.phi.0835$Li.phi.0835,1)),
      cex=0.8)
text(x=dad.ic.phi.0835$Ls.phi.0835,y=-0.1,

```

```
label=as.character(round(dad.ic.phi.0835$ls.phi.0835,2)),cex=0.8)
dev.off()
```

Perfilhada para ρ

Arquivo **monta.V.rho.R**

```
## theta1=(eta,nu1,nu2,phi)
monta.V.rho <-function(theta1, rho, dados.comp){
  y <- dados.comp[[2]][[1]]
  eta <- theta1[1]
  nu1 <- theta1[2]
  nu2 <- theta1[3]
  phi <- theta1[4]
  rho <- rho
  ## Calculando os elementos da matriz de correlação espacial
  distancia <- unname(as.matrix(dist(dados.comp[[3]],diag=TRUE,
                                   upper=TRUE)))
  correla <- exp(-distancia/phi)
  ## Organizando a matriz de correlação espacial com eltos y1 e y2
  ## intercalados. Ou seja, calculando R1
  coluna1 <- rep(c(1, eta),length(y)/2)
  coluna2 <- rep(c(eta,eta^2),length(y)/2)
  seq1 <- seq(1,length(y),by=2)
  seq2 <- seq(2,length(y),by=2)
  ERRE1 <- matrix(ncol=length(y),nrow=length(y))
  ERRE1[,c(seq1)] <- coluna1
  ERRE1[,c(seq2)] <- coluna2
  ## Organizando a matriz de correlação espacial com eltos y1 e y2
  ## intercalados
  ERRE2 <- matrix(nrow=length(y),ncol=length(y))
  for(i in 1:length(y)/2){
    ERRE2[seq1[i],] <- rep(correla[i,],each=2)
    ERRE2[seq2[i],] <- rep(correla[i,],each=2)
  }
  ## Calculando a matriz R
  ERRE <- ERRE1*ERRE2
  ## Calculando a matriz de covariância composicional
  Ib1 <- matrix(c(nu1^2,nu1*nu2*rho,nu1*nu2*rho,nu2^2),nc=2)
  Ib <- kronecker(diag(length(y)/2),Ib1)
```

```

## Matriz de covariância espacial composicional
V <- ERRE + Ib
return(V)
}

```

Arquivo **log.vero.rho.R**

```

## theta1=(eta,nu1,nu2,phi)
log.vero.rho <-function(theta1,rho,dados.comp,print.pars=F){
  if(theta1[1] <= 0)      return(.Machine$double.xmax^0.5)
  if(theta1[2] <= 0)      return(.Machine$double.xmax^0.5)
  if(theta1[3] <= 0)      return(.Machine$double.xmax^0.5)
  if(theta1[4] < 0)       return(.Machine$double.xmax^0.5)
  y <- dados.comp[[2]][[1]]
  X <- cbind(rep(1:0,length=length(y)),rep(0:1,length=length(y)))
  V <- monta.V.rho(theta1=theta1,rho=rho ,dados.comp = dados.comp)
  ldetV <- determinant(V,log=TRUE)$modulus[1]
  mu <- drop(solve(crossprod(X,solve(V,X))) %*%crossprod(X,solve(V,y)))
  Qe <- drop(crossprod(y,solve(V,y))-2*crossprod(y,solve(V,X%*%mu))+
             crossprod(mu,crossprod(X,solve(V,X%*%mu))))
  if(Qe < 0) return(Qe=.Machine$double.xmax^0.5)
  n <- length(y)
  s1 <- sqrt(Qe/n)
  ll <- drop(-0.5*(n*log(2*pi)+n*log(s1^2)+ ldetV + (1/(s1^2))*Qe))
  if(print.pars) print(c(theta1,ll))
  return(-ll)
}

```

Arquivo **perf.rho.R**

```

## Obtendo valores iniciais para theta1 = (eta, nu1, nu2, phi)
perf.rho <- function(dados.comp,
                    tamanho,# Tamanho do vetor rho p/ perfilhar
                    min.delta, # Qto % menor que o mínimo dado pelo delta
                    max.delta, # Qto % maior que o máximo
                    alpha){
  var_y1 <- var(dados.comp[[1]][1])
  s1 <- var_y1/2
  tau1 <- s1

```

```

var_y2 <- var(dados.comp[[1]][2])
s2 <- var_y2/2
tau2 <- s2
dim <- range(dist(dados[[3]]))
phi <- dim[1]+0.2*(dim[2]-dim[1])
eta <- s2/s1
nu1 <- tau1/s1
nu2 <- tau2/s1
theta1 <- c(eta,nu1,nu2,phi)
## Fazendo a maximização
source("monta.V.rho.R")
source("log.vero.rho.R")
estima <- mec(dados.comp=dados.comp)
## ic.inf.delta = limite inferior para rho pelo metodo delta
## ic.sup.delta = limite superior para rho pelo metodo delta
ic_i.delta.rho <- estima[[1]]$LI.Delta[8]
ic_s.delta.rho <- estima[[1]]$LS.Delta[8]
rho <- seq(ic_i.delta.rho*min.delta,ic_i.delta.rho*max.delta,l=tamanho)
rho[which(rho > 1*0.9995)] <- 1*0.9995
lista.res <- list()
for(i in 1:length(rho)){
  teste <- optim(theta1, log.vero.rho, rho=rho[i],
                dados.comp=dados.comp, method="L-BFGS-B",
                lower=c(1e-32,1e-32,1e-32,1e-32),
                upper=c(20,20,20,200), hessian=TRUE)
  ## Todos os resultados para cada valor de rho
  lista.res[i][[1]] <- teste
}
perf.rho <- c()
for(i in 1:length(rho)){
  perf.rho[i] <- lista.res[[i]][2]$value
}
perf.rho
## Gráfico da perfilhada: rho versus log-verossimilhança
#postscript('perf.rho.eps')
#plot(rho,perf.rho,ylab="Log-verossimilhanca Perfilhada")
#dev.off()
maximo <- -estima[[2]][2]

```

```

deviance.rho <- 2*(perf.rho-maximo)
#postscript('deviance.rho.eps')
#plot(rho,deviance.rho,ylab='Deviance',type="l")
#dev.off()
#abline(h=qchisq(0.95,df=1))
val.rho.perf.dev <- data.frame(rho,perf.rho,deviance.rho)
#intervalo.rho <- range(val.rho.perf.dev[which(val.rho.perf.dev$deviance<
      qchisq(0.95,df=1)),]$rho)
intervalo.rho <- val.rho.perf.dev[which(val.rho.perf.dev$deviance.rho<
      qchisq(alpha,df=1)),]$rho

retorna <- list()
retorna[[1]] <- val.rho.perf.dev
retorna[[2]] <- intervalo.rho
return(retorna)
}

```

Arquivo **script.perfilhada.rho.R**

```

rm(list=ls(all=T))
## Carregando o arquivo de dados do geoComp
source('pivo.R')
##data(pivo)
pivo
## Selecionando os componentes e as coordenadas
dados <- pivo[,c(6,7,8,1,2)]
## Convertendo o arquivo para o formato geoComp
## Transformacao ALR ( denominador=argila )
## dados[[1]]= Y1 Y2 ; dados[[2]]= Y ; dados[[3]]= Coord.X Coord.Y
source('as.geoComp.R')
dados <- as.geoComp(dados)
source('monta.V.R')
source('logarit.vero.R')
source('mec.R')
dados.comp <- dados
estima <- mec(dados)
## Fazendo a perfilhada para rho com alpha=095
source('perf.rho.R')
perfilhada.rho <- perf.rho(dados,tamanho=150,min.delta=0.85,max.delta=1.10,
      alpha=0.95)

```

```

perfilhada.rho.1 <- perfilhada.rho[[1]]
write.table(perfilhada.rho.1,'dataframePerfrho095p150.txt')
perfilhada.rho.2 <- perfilhada.rho[[2]]
write.table(perfilhada.rho.2 , 'IC.Perf.rho.095p150.txt')
## Lendo os resultados
dad.per.rho.095 <- read.table('dataframePerfrho095p150.txt')
dad.ic.rho.095 <- read.table('IC.Perf.rho.095p150.txt')
## Organizando os intervalos
dad.ic.rho.095 <- data.frame(dad.ic.rho.095[1,],dad.ic.rho.095[90,])
names(dad.ic.rho.095) <- c('Li.rho.095','Ls.rho.095')
## Gráficos das perfilhadas: rho versus log-verossimilhanca
postscript('perfrho095.eps')
par(mfrow=c(1,1),mar=c(3,3,1.5,1.5),mgp=c(1.8,0.8,0))
plot(dad.per.rho.095$rho,-dad.per.rho.095$perf.rho,xlab=expression(rho),
      ylab="Log-verossimilhanca Perfilhada",type='l')
dev.off()
## Gráficos das deviances: rho versus Deviance
postscript('deviancerho095.eps')
par(mfrow=c(1,1),mar=c(3,3,1.5,1.5),mgp=c(1.8,0.8,0))
plot(dad.per.rho.095$rho,dad.per.rho.095$deviance.rho,xlab=expression(rho),
      ylab='Deviance',type="l")
segments(x0=dad.ic.rho.095$Li.rho.095,x1=dad.ic.rho.095$Ls.rho.095,
          y0=qchisq(0.95,df=1),y1=qchisq(0.95,df=1))
arrows(x0=dad.ic.rho.095$Li.rho.095,y0=qchisq(0.95,df=1),
        x1=dad.ic.rho.095$Ls.rho.095,y1=0)
text(x=dad.ic.rho.095$Li.rho.095,y=-0.2,
      label=as.character(round(dad.ic.rho.095$Li.rho.095,2)),cex=0.8)
dev.off()

```

Script para fazer inferência bayesiana:

```

## Carregando pacotes
require(geOR) # necessário para usar rinvchisq
require(geoComp)
require(MCMCpack)
require(MASS)
## Lendo dados
dados <- pivo[,c(6,7,8,1,2)]
## Transformando os dados para a classe geoComp

```

```

dados <- as.geoComp(dados)
estima.bayes <- infBayes(dados,tune=c(0.5,0.5,0.5,0.5,0.5),
                        queima=1000,salto=10,nsim=12000,mediahiper=c())
write.table(estima.bayes[[1]],'estima.bayes1MBMCCAD.txt')
write.table(estima.bayes[[2]],'estima.bayes2MBMCCAD.txt')
## Opção para o usuário entrar com os parâmetros das prioris:
#estima.bayes <- infBayes(dados,tune=c(0.5,0.5,0.5,0.5,0.5),
                        queima=1000,salto=10,nsim=12000,
                        mediahiper=c(1,1,1,1,1,1,1,1,1,1))
summary(estima.bayes[[2]])
res.mh <- resultado.mh(resultado=estima.bayes)
res.mh.ic <- data.frame(res.mh[[2]])
names(res.mh.ic) <- c('2.5%', 'media', '97.5%')
write.table(res.mh.ic, 'res.mh.icMBMCCAD.txt')
metro.bayes <- data.frame(res.mh[[1]])
write.table(metro.bayes, 'metro.bayesMBMCCAD.txt')
bor <- cbind(c(0,seq(0,200,l=100),0),
            c(0,sqrt(200^2-seq(0,200,l=100)^2),0))
gr <- pred_grid(bor, by=4)
cokri.bayes <- cokri.bayes(estim.par=metro.bayes,locations=gr,
                        dados.comp=dados)
write.table(cokri.bayes[[1]], 'cokri1MBMCCAD.txt')
write.table(cokri.bayes[[2]], 'cokri2MBMCCAD.txt')
## Construindo os mapas
require(geoR)
tes1 <- read.table('cokri1MBMCCAD.txt')#composicao
tes2 <- read.table('cokri2MBMCCAD.txt')#coordenadas
tes1 <- as.matrix(tes1)
tes2 <- as.matrix(tes2)
bor <- cbind(c(0,seq(0,200,l=100),0),c(0,sqrt(200^2-seq(0,200,l=100)^2),0))
require(sp)
par(mfrow=c(1,3), mar=c(3.5, 3.5, 0.5, 0.5), mgp=c(1.8,.8, 0))
xleg <- c(85, 198); yleg <- c(185, 200)
image(structure(list(predict=tes1[,1]), class="kriging"),
      loc=tes2,bor, col=bpy.colors(15),xlab='',ylab='',
      x.leg=xleg, y.leg=yleg,ylim=c(0,220))
text(x=165,y=170,"(A)Areia/IB")
image(structure(list(predict=tes1[,2]), class="kriging"),

```



```

loc=tes2,bor, col=bpy.colors(15),xlab='',ylab='',
x.leg=xleg, y.leg=yleg,ylim=c(0,220))
text(x=165,y=170,"(B)Silte/IB")
image(structure(list(predict=tes1[,3]), class="kriging"),
loc=tes2,bor, col=bpy.colors(15),xlab='',ylab='',
x.leg=xleg, y.leg=yleg,ylim=c(0,220))
text(x=165,y=170,"(C)Argila/IB")
dev.off()

```

Arquivo **densidadeBayesNaoBayes.R**

```

## Script que compara as densidades por inferência bayesiana e não-bayesiana.
a <- read.table('cokri1MBMCCAD.txt',head=TRUE)
b <- read.table('cokri2MBMCCAD.txt',head=TRUE)
c <- read.table('predsimu.are_by4k7ns1000MBM.txt',head=TRUE)
d <- read.table('predsimu.sil_by4k7ns1000MBM.txt',head=TRUE)
e <- read.table('predsimu.arg_by4k7ns1000MBM.txt',head=TRUE)
sim.c.t <- t(c)
dim(sim.c.t)
saida.c.t <- c()
for(i in 1:dim(sim.c.t)[2]){
  saida.c.t[i] <- mean(sim.c.t[,i])
}
saida.c.t
sim.d.t <- t(d)
dim(sim.d.t)
saida.d.t <- c()
for(i in 1:dim(sim.d.t)[2]){
  saida.d.t[i] <- mean(sim.d.t[,i])
}
saida.d.t
sim.e.t <- t(e)
dim(sim.e.t)
saida.e.t <- c()
for(i in 1:dim(sim.e.t)[2]){
  saida.e.t[i] <- mean(sim.e.t[,i])
}
saida.e.t
simu <- data.frame(saida.c.t,saida.d.t,saida.e.t)

```

```

names(simu) <- c('AreSim','SilSim','ArgSim')
simu
## Gráficos das densidades
postscript('densidadeBayesNaoBayes.eps',width=8,height=2.66,horizontal=FALSE)
#pdf('densidadeBayesNaoBayes.pdf',width=6,height=2)
par(mfrow=c(1,3),mar=c(3,3,1.5,0.5),mgp=c(1.8,0.8,0))
plot(density(a$V1),lty=1,main='Areia',ylim=c(0,12),xlab='Percentual',
      ylab='Densidade')
lines(density(simu$AreSim),lty=2)
plot(density(a$V2),lty=1,main='Silte',xlab='Percentual',ylab='Densidade')
lines(density(simu$SilSim),lty=2)
plot(density(a$V3),lty=1,main='Argila',xlab='Percentual',ylab='Densidade')
lines(density(simu$ArgSim),lty=2)
legend(x=0.49,y=14, legend=c("Bayes", "N.Bayes"), fill=NULL, border="white",
text.col = "black", bg = 'white',lty=c(1,2),bty='n')
dev.off()

```

Script para fazer diagnóstico de convergência:

```

## Fazendo diagnóstico
require(coda)
metro.emv <- read.table('metro.bayesMBMCCAD.txt')
metro.emv <- as.mcmc(metro.emv)
## Calculando as autocorrelações das posteriori dos parâmetros e fazendo os
## respectivos gráficos.
ac <- autocorr.diag(metro.emv)
par(mar=c(3.5,3.5,1.5,0.5),mgp=c(1.8,0.8,0))
autocorr.plot <- function(x, lag.max, auto.layout = TRUE, ask, ...){
  if (missing(ask)) { ask <- if (is.R()) { dev.interactive() }
                    else { interactive() }
  }
  oldpar <- NULL
  on.exit(par(oldpar))
  if (auto.layout)
    oldpar <- par(mfrow = c(3,3))
  if (!is.mcmc.list(x))
    x <- mcmc.list(as.mcmc(x))
  for (i in 1:nchain(x)) {
    xacf <- if (missing(lag.max))

```

```

        acf(as.ts.mcmc(x[[i]]), plot = FALSE)
    else acf(as.ts.mcmc(x[[i]]), lag.max = lag.max, plot = FALSE)
    for (j in 1:nvar(x)) { plot(xacf$lag[, j, j], xacf$acf[, j, j],
                               type = "h", ylab = "Autocorrelacao",
                               xlab = "Defasagem", ylim = c(-1,1), ...)
        title(paste(varnames(x)[j], ifelse(is.null(chanames(x)), "", ":"),
                    chanames(x)[i], sep = ""))
        if (i == 1 && j == 1)
            oldpar <- c(oldpar, par(ask = ask))
    }
}
% invisible(x)
}
autocorr.plot(metro.emv)
## Calculando as correlações dos parâmetros a posteriori
crosscor <- crosscorr(metro.emv)
## Construindo os gráficos das trajetórias das cadeias
traceplot <- function (x, smooth = TRUE, col = 1:6, type = "l", ylab = "", ...)
{
  x <- mcmc.list(x)
  args <- list(...)
  for (j in 1:nvar(x)) {
    xp <- as.vector(time(x))
    yp <- if (nvar(x) > 1)
        x[, j, drop = TRUE]
    else x
    yp <- do.call("cbind", yp)
    matplot(xp, yp, xlab = "Iteracoes", ylab = ylab, type = type,
            col = col, ...)
    if (!is.null(varnames(x)) && is.null(list(...)$main))
        title(paste("Trace of", varnames(x)[j]))
    if (smooth) {
        scol <- rep(col, length = nchain(x))
        for (k in 1:nchain(x)) lines(lowess(xp, yp[, k]),
            col = scol[k])
    }
  }
}
}

```

```

par(mfrow=c(4,2),mar=c(3,3,1.5,0.5),mgp=c(1.8,0.8,0))
traceplot(metro.emv[,1], main=expression(mu[1]))
traceplot(metro.emv[,2], main=expression(mu[2]))
traceplot(metro.emv[,3], main=expression(tau[1]))
traceplot(metro.emv[,4], main=expression(tau[2]))
traceplot(metro.emv[,5], main=expression(sigma[1]))
traceplot(metro.emv[,6], main=expression(sigma[2]))
traceplot(metro.emv[,7], main=expression(phi))
traceplot(metro.emv[,8], main=expression(rho))
## Calculando o intervalo da mais alta densidade a posteriori
hpd <-HPDinterval(metro.emv)
## Construindo os gráficos das densidades estimadas para as posteriores}
## dos parâmetros com respectivos intervalos de alta densidade a
## posteriori a 95%
par(mfrow=c(4,2),mar=c(3.5,3.5,1.5,0.5),mgp=c(1.8,0.8,0))
plot(density(metro.emv[,1]), main=expression(mu[1]),ylab='Densidade')
rug(metro.emv[,1])
#lty=linha,lwd=largura da linha
lines(seq(hpd[1,1][1],hpd[1,2][1],l=10),rep(0,10),type="l",lty=1,lwd=5)
plot(density(metro.emv[,2]), main=expression(mu[2]),ylab='Densidade')
rug(metro.emv[,2])
lines(seq(hpd[2,1][1],hpd[2,2][1],l=10),rep(0,10),type="l",lty=1,lwd=5)
plot(density(metro.emv[,3]), main=expression(tau[1]),ylab='Densidade')
rug(metro.emv[,3])
lines(seq(hpd[3,1][1],hpd[3,2][1],l=10),rep(0,10),type="l",lty=1,lwd=5)
plot(density(metro.emv[,4]), main=expression(tau[2]),ylab='Densidade')
rug(metro.emv[,4])
lines(seq(hpd[4,1][1],hpd[4,2][1],l=10),rep(0,10),type="l",lty=1,lwd=5)
plot(density(metro.emv[,5]), main=expression(sigma[1]),ylab='Densidade')
rug(metro.emv[,5])
lines(seq(hpd[5,1][1],hpd[5,2][1],l=10),rep(0,10),type="l",lty=1,lwd=5)
plot(density(metro.emv[,6]), main=expression(sigma[2]),ylab='Densidade')
rug(metro.emv[,6])
lines(seq(hpd[6,1][1],hpd[6,2][1],l=10),rep(0,10),type="l",lty=1,lwd=5)
plot(density(metro.emv[,7]), main=expression(phi),ylab='Densidade')
rug(metro.emv[,7])
lines(seq(hpd[7,1][1],hpd[7,2][1],l=10),rep(0,10),type="l",lty=1,lwd=5)
plot(density(metro.emv[,8]), main=expression(rho),ylab='Densidade')

```

```

rug(metro.emv[,8])
lines(seq(hpd[8,1][1],hpd[8,2][1],l=10),rep(0,10),type="l",lty=1,lwd=5)
teste <- geweke.diag(metro.emv,frac1=0.90,frac2=0.5)
## Construindo os gráficos do diagnóstico de convergência para
## o Teste de Geweke.
par(mar=c(3.5,3.5,1.5,0.5),mgp=c(1.8,0.8,0))
grafico <- geweke.plot(metro.emv,frac1=0.90,frac2=0.50,nbins=20,pvalue=0.05)
## Obtendo uma segunda cadeia:
## Rodar o "Script para fazer inferência bayesiana" utilizando
## theta1.inicial <-c(1,1,1,40,0.8) na função infbayes.R
## para fazer o teste Gelman eRubin
metro.emv.cadeia4 <- read.table('metro.bayesCadeia4.txt')
metro.emv.cadeia4 <- as.mcmc(metro.emv.cadeia4)
## Teste de Gelman e Rubin
dadoPgelman <- list()
dadoPgelman[[1]] <- metro.emv
dadoPgelman[[2]] <- metro.emv.cadeia4
res.gelman <- gelman.diag(dadoPgelman,confidence=0.95,transform=FALSE,
                          autoburnin=TRUE)
## Construindo os gráficos do diagnóstico de Gelman e Rubin
par(mar=c(3.5,3.5,1.5,0.5),mgp=c(1.8,0.8,0))
gelmanplot <- gelman.plot(dadoPgelman,bin.width=10,max.bins=50,confidence=0.95,
                          transform=FALSE,auto.layout=TRUE)

```

Scripts referentes aos funcionais:

Arquivo **Funcionais.R**:

```

## Script para obter 1000 simulações de areia, silte e argila em cada
## localização
rm(list=ls(all=TRUE))
require(MASS)
require(statmod)
require(geoR)
require(geoComp)
data(pivo)
dados <- pivo[,c(6,7,8,1,2)]
dados <- as.geoComp(dados)
bor <- cbind(c(0,seq(0, 200, l=100), 0),
            c(0,sqrt(200^2-seq(0, 200, l=100)^2), 0))

```

```

estima <- mec(dados)
gr <- pred_grid(bor, by=4)
md.cov.ck <- cokrigagem(estima[[1]]$Estimativas, loc=gr, dados.comp=dados)
#preditos.gh <- volta.quad(md.cov.ck,n.pontos=7,Variancia=FALSE)
#preditos.gh <- data.frame(preditos.gh)
#names(preditos.gh) <- c("Areia","Silte","Argila")
#write.table(preditos.gh,"predgh_by4k7ns1000MBM.txt")
preditos.simu <- volta.cokri(md.cov.ck,num.simu=1000,retorna.tudo=TRUE,
                           int.conf=0.95)
preditos.simu.areia <- data.frame(preditos.simu[[1]])
preditos.simu.silte <- data.frame(preditos.simu[[2]])
preditos.simu.argila <- data.frame(preditos.simu[[3]])
write.table(preditos.simu.areia,"predsimu.are__by4k7ns1000MBM.txt")
write.table(preditos.simu.silte,"predsimu.sil__by4k7ns1000MBM.txt")
write.table(preditos.simu.argila,"predsimu.arg_by4k7ns1000MBM.txt")
### Se retorna.tudo=FALSE use as 4 linhas seguintes
##preditos.simu <- data.frame(preditos.simu[[1]])
##preditos.simu.ic <- data.frame(preditos.simu[[2]])
##write.table(preditos.simu,"predsimu_by4k7ns1000MBM.txt")
##write.table(preditos.simu.ic,"predsimuic_by4k7ns1000MBM.txt")

```

Arquivo dadosMBM.Montando.R:

```

areMBM <- read.table('predsimu.are_by4k7ns1000MBM.txt',header=TRUE)
silMBM <- read.table('predsimu.sil_by4k7ns1000MBM.txt',header=TRUE)
argMBM <- read.table('predsimu.arg_by4k7ns1000MBM.txt',header=TRUE)
are100 <- areMBM[1:2601,1:100]
sil100 <- silMBM[1:2601,1:100]
arg100 <- argMBM[1:2601,1:100]
dadosMBM1 <- data.frame(areia= are100, silte=sil100, argila=arg100)
write.table(dadosMBM1,'dadosMBM1.txt')
are200 <- areMBM[1:2601,101:200]
sil200 <- silMBM[1:2601,101:200]
arg200 <- argMBM[1:2601,101:200]
dadosMBM2 <- data.frame(areia= are200, silte=sil200, argila=arg200)
write.table(dadosMBM2,'dadosMBM2.txt')
are300 <- areMBM[1:2601,201:300]
sil300 <- silMBM[1:2601,201:300]
arg300 <- argMBM[1:2601,201:300]

```

```
dadosMBM3 <- data.frame(areia= are300, silte=sil300, argila=arg300)
write.table(dadosMBM3, 'dadosMBM3.txt')
are400 <- areMBM[1:2601, 301:400]
sil400 <- silMBM[1:2601, 301:400]
arg400 <- argMBM[1:2601, 301:400]
dadosMBM4 <- data.frame(areia= are400, silte=sil400, argila=arg400)
write.table(dadosMBM4, 'dadosMBM4.txt')
are500 <- areMBM[1:2601, 401:500]
sil500 <- silMBM[1:2601, 401:500]
arg500 <- argMBM[1:2601, 401:500]
dadosMBM5 <- data.frame(areia= are500, silte=sil500, argila=arg500)
write.table(dadosMBM5, 'dadosMBM5.txt')
are600 <- areMBM[1:2601, 501:600]
sil600 <- silMBM[1:2601, 501:600]
arg600 <- argMBM[1:2601, 501:600]
dadosMBM6 <- data.frame(areia= are600, silte=sil600, argila=arg600)
write.table(dadosMBM6, 'dadosMBM6.txt')
are700 <- areMBM[1:2601, 601:700]
sil700 <- silMBM[1:2601, 601:700]
arg700 <- argMBM[1:2601, 601:700]
dadosMBM7 <- data.frame(areia= are700, silte=sil700, argila=arg700)
write.table(dadosMBM7, 'dadosMBM7.txt')
are800 <- areMBM[1:2601, 701:800]
sil800 <- silMBM[1:2601, 701:800]
arg800 <- argMBM[1:2601, 701:800]
dadosMBM8 <- data.frame(areia= are800, silte=sil800, argila=arg800)
write.table(dadosMBM8, 'dadosMBM8.txt')
are900 <- areMBM[1:2601, 801:900]
sil900 <- silMBM[1:2601, 801:900]
arg900 <- argMBM[1:2601, 801:900]
dadosMBM9 <- data.frame(areia= are900, silte=sil900, argila=arg900)
write.table(dadosMBM9, 'dadosMBM9.txt')
are1000 <- areMBM[1:2601, 901:1000]
sil1000 <- silMBM[1:2601, 901:1000]
arg1000 <- argMBM[1:2601, 901:1000]
dadosMBM10 <- data.frame(areia= are1000, silte=sil1000, argila=arg1000)
write.table(dadosMBM10, 'dadosMBM10.txt')
```

Arquivo **script1.R**:

```
## Carregando as funções ###
require(geoR)
require(sp)
source('classificaSolo.R')
source("classSolo.R")
data(pivo)
data(tab.coord.pol) # ou load("tab.coord.pol.rda")
tab.pol <- tab.coord.pol
## Fazendo a classificação do solo em cada localização das 100 simulações
n.for = 100
resultado <- data.frame(matrix(nrow=2601,ncol=n.for))
for(i in 1:n.for){
temp = read.table("dadosMBM1.txt",header=TRUE)
temp2 = data.frame(temp[,i],temp[,i+100],temp[,i+200])
res.temp = classificaSolo(dados=temp2,class.dados=TRUE,plot.dados=FALSE,
                          newClass=FALSE,
                          new.comp,plot.new=FALSE)
saida.temp = data.frame(classe = res.temp[[1]]$class.amostra)
resultado[,i] <- saida.temp
write.table(resultado,"resultadoMBM1.txt")
# repetir este procedimento até "dadosMBM10.txt", obtendo
# "resultadoMBM10.txt"
}
```

Arquivo **funcionalEst.R**:

```
## Função para calcular funcionais dos valores preditos simulados
funcionalEst <- function(dados,funcional,prob.quartil){
  if(funcional=="Maximo"){
    max <- apply(dados,1,max)
    return(max)
  }
  if(funcional=="Minimo"){
    min <- apply(dados,1,min)
    return(min)
  }
  ## Outros possíveis funcionais
```



```

if(funcional=="Media"){
  media <- apply(dados,1,mean)
  return(media)
}
if(funcional=="Variancia"){
  require(stats)
  var <- apply(dados,1,var)
  desv.pad <- sqrt(var)
  resultado <- data.frame(var,desv.pad)
  names(resultado) <- c("Variancia","Desvio padrao")
  return(resultado)
}
if(funcional=="Quartil"){
  quartil <- data.frame(t(apply(dados,1,quantile,prob=c(prob.quartil))))
  return(quartil)
}
}

```

Arquivo **histFunMaxMinMBM.R**:

```

## Lendo os arquivos de dados
predi.simu.are <- read.table('predsimu.are_by4k7ns1000MBM.txt',header=TRUE)
predi.simu.sil <- read.table('predsimu.sil_by4k7ns1000MBM.txt',header=TRUE)
predi.simu.arg <- read.table('predsimu.arg_by4k7ns1000MBM.txt',header=TRUE)
## Funcionais máximos e mínimos por coluna
funcionalEst <- function(dados,funcional,prob.quartil){
  if(funcional=="Maximo"){
    max <- apply(dados,2,max)
    return(max)
  }
  if(funcional=="Minimo"){
    min <- apply(dados,2,min)
    return(min)
  }
}
max.are <- matrix(funcionalEst(dados=predi.simu.are,funcional='Maximo'),nc=1)
min.are <- matrix(funcionalEst(dados=predi.simu.are,funcional='Minimo'),nc=1)
max.sil <- matrix(funcionalEst(dados=predi.simu.sil,funcional='Maximo'),nc=1)
min.sil <- matrix(funcionalEst(dados=predi.simu.sil,funcional='Minimo'),nc=1)

```

```

max.arg <- matrix(funcionalEst(dados=predi.simu.arg,funcional='Maximo'),nc=1)
min.arg <- matrix(funcionalEst(dados=predi.simu.arg,funcional='Minimo'),nc=1)
## Construindo os gráficos das densidades
postscript('histFunMaxMinMBM.eps',width=7.5,height=5)
par(mfrow=c(2,3), mar=c(3.1,3.1,0.5,.5), mgp=c(1.8,0.8, 0))
hist(max.ave,,prob=T,main="", xlab="Areia",ylab="Densidade",ylim=c(0,8))
lines(density(max.ave),col=2)
hist(max.sil,prob=T,main="", xlab="Silte",ylab="Densidade",ylim=c(0,15),
      xlim=c(0.4,0.65))
lines(density(max.sil),col=2)
hist(max.arg,prob=T, main="", xlab="Argila",ylab="Densidade",ylim=c(0,25))
lines(density(max.arg),col=2)
hist(min.ave,prob=T, main="", xlab="Areia",ylab="Densidade",ylim=c(0,50))
lines(density(min.ave),col=2)
hist(min.sil,prob=T, main="", xlab="Silte",ylab="Densidade",ylim=c(0,50))
lines(density(min.sil),col=2)
hist(min.arg,prob=T, main="", xlab="Argila",ylab="Densidade",ylim=c(0,20))
lines(density(min.arg),col=2)
dev.off()

```

Arquivo **FunHistSim.R**:

```

## Lendo os resultados das 1000 simulações
res1MBM <- read.table('resultadoMBM1.txt',head=T)
res2MBM <- read.table('resultadoMBM2.txt',head=T)
res3MBM <- read.table('resultadoMBM3.txt',head=T)
res4MBM <- read.table('resultadoMBM4.txt',head=T)
res5MBM <- read.table('resultadoMBM5.txt',head=T)
res6MBM <- read.table('resultadoMBM6.txt',head=T)
res7MBM <- read.table('resultadoMBM7.txt',head=T)
res8MBM <- read.table('resultadoMBM8.txt',head=T)
res9MBM <- read.table('resultadoMBM9.txt',head=T)
res10MBM <- read.table('resultadoMBM10.txt',head=T)
resultado1000MBM <- data.frame(res1MBM,res2MBM,res3MBM,res4MBM,
                              res5MBM,res6MBM,res7MBM,res8MBM,
                              res9MBM,res10MBM)

dim(resultado1000MBM)
write.table(resultado1000MBM,'resultado1000MBM.txt')
resultado <- read.table('resultado1000MBM.txt')

```

```

## Função que calcula para cada simulação os percentuais de cada categoria
## de classificação
classifica.pos <- function(categoria='Muito argilosa',nlinhas=2601,
                           ncolunas=100,resultado1){
  temporario <- c()
  for(i in 1:ncolunas){
    temporario[i] = length(resultado[which(
      resultado[,i] == categoria),][,i])/nlinhas
  }
  return(temporario)
}

#postscript('histClassificaDensidade1000MBM.eps',width=5,height=4.5)
pdf('histClassificaDensidade1000MBM.pdf',width=5,height=4.5)
par(mfrow=c(4,3), mar=c(2.5, 3.0, 0.5, 1.5), mgp=c(1.6,0.6,0),cex=0.6)
Muito.Argilosa <- classifica.pos('Muito argilosa',2601,1000,resultado)
hist(Muito.Argilosa,prob=T,xlim=c(0.15,0.35),ylim=c(0,15),main='',
     xlab='Muito argilosa',ylab='Densidade')
lines(density(Muito.Argilosa),col=2)
Argila <- classifica.pos('Argila',2601,1000,resultado)
hist(Argila,prob=T,ylim=c(0,30),main='',xlab='Argila',ylab='Densidade')
lines(density(Argila),col=2)
Argila.arenosa <- classifica.pos('Argila arenosa',2601,1000,resultado)
hist(Argila.arenosa,prob=T,ylim=c(0,200),main='',xlab='Argila arenosa',
     ylab='Densidade')
lines(density(Argila.arenosa),col=2)
#####
Argila.siltosa <- classifica.pos('Argila siltosa',2601,1000,resultado)
hist(Argila.siltosa,ylim=c(0,1500),prob=T,main='',xlab='Argila siltosa',
     ylab='Densidade')
lines(density(Argila.siltosa),col=2)
Franco.argilo.arenosa <- classifica.pos('Franco argilo-arenosa',
                                         2601,1000,resultado)
hist(Franco.argilo.arenosa,prob=T,xlim=c(.02,0.14),ylim=c(0,25),main='',
     xlab='Franco argilo-arenosa',ylab='Densidade')
lines(density(Franco.argilo.arenosa),col=2)
Franco.argiloso <- classifica.pos('Franco argiloso',2601,1000,resultado)
hist(Franco.argiloso,prob=T,xlim=c(.14,0.22),ylim=c(0,40),main='',
     xlab='Franco argiloso',ylab='Densidade')

```

```

lines(density(Franco.argiloso),col=2)
#####
Franco.argilo.siltoso <- classifica.pos('Franco argilo-siltoso',2601,
                                     1000,resultado)
hist(Franco.argilo.siltoso,prob=T,xlim=c(0,0.0025),ylim=c(0,2000),main='',
     xlab='Franco argilo-siltoso',ylab='Densidade')
lines(density(Franco.argilo.siltoso),col=2)
Franco.arenoso <- classifica.pos('Franco arenoso',2601,1000,resultado)
hist(Franco.arenoso,prob=T,main='',xlab='Franco arenoso',ylab='Densidade',
     xlim=c(0,0.08))
lines(density(Franco.arenoso),col=2)
Franco <- classifica.pos('Franco',2601,1000,resultado)
hist(Franco,prob=T,xlim=c(0.030,0.065),ylim=c(0,100),main='',xlab='Franco',
     ylab='Densidade')
lines(density(Franco),col=2)
#####
Franco.siltoso <- classifica.pos('Franco-siltoso',2601,1000,resultado)
hist(Franco.siltoso,prob=T,ylim=c(0,8000),main='',xlab='Franco siltoso',
     ylab='Densidade')
lines(density(Franco.siltoso),col=2)
Areia.franca <- classifica.pos('Areia Franca',2601,1000,resultado)
hist(Areia.franca,prob=T,main='',ylim=c(0,500),xlab='Areia Franca',
     ylab='Densidade')
lines(density(Areia.franca,bw=0.0009),col=2)#bw=0.0005
Areia <- classifica.pos('Areia',2601,1000,resultado)
hist(Areia,prob=T,ylim=c(0,2000),main='',xlab='Areia',ylab='Densidade')
lines(density(Areia,bw=0.0002),col=2)
dev.off()

```

ANEXO D – Código fonte R dos diagramas ternários

Script para a construção da Figura 2.1:

```
require(MASS)
data(Skye)
X <- Skye
ternaryv <- function(X,pch=par('pch'),lcex=1,add=FALSE,ord=1:3){
  X <- as.matrix(X)
  if(any(X<0)) stop('X must be non-negative')
  s <- drop(X %*% rep(1,ncol(X)))
  if(any(s<=0)) stop('each row of X must have a positive sum')
  if(max(abs(s-1))>1e-6){
    warning('rows(s) of will be rescaled')
    X <- X/s
  }
  X <- X[,ord]
  s3 <- sqrt(1/3)
  if(!add){
    oldpty <- par('pty')
    on.exit(par(pty=oldpty))
    par(pty='s')
    par(mar=c(3.5,3.5,1.5,1.5),mgp=c(1.7,0.7,0))
    plot(c(-s3,s3),c(0.5-s3,0.5+s3),type='n',axes=TRUE,xlab='',ylab='')
    segments(x0=0,y0=0,x1=0,y1=1,lty=2);text(x=0,y=-0.04,'H') #altura
    text(x=0.03,y=0.3, 'h')
    ## paralelas FM
    text(x=0.39,y=0.57, 'L'); text(x=0.48,y=0.22, 'E')
    ## paralelas AM
    segments(x0=-0.231,y0=0.6,x1=0.115,y1=0,lty=3);text(x=0.12,y=-0.04,'I')
    ##p-13
  }
}
```

```

segments(x0=-0.115,y0=0.4,x1=-0.115,y1=0,lty=1)
##p-24
segments(x0=-0.115,y0=0.4,x1=0.231,y1=0.60,lty=1)
##p-5
segments(x0=-0.115,y0=0.4,x1=-0.289,y1=0.50,lty=1)
##15-20
segments(x0=0.115,y0=0,x1=0.462,y1=0.20,lty=1);text(x=0.25,y=0.11,'X2')
polygon(c(0,-s3,s3),c(1,0,0),density=0)
}
points(-0.115,0.40,pch=20);text(x=-0.11,y=0.45,'p')#ponto p
text(x=-0.07,y=0.2,'X1');text(x=0.05,y=0.53,'X2');text(x=-0.21,y=0.51,'X3')
text(x=-0.57,y=-0.06,'F');text(x=0.58,y=-0.06,'M');text(x=0.0,y=1.03,'A')
text(x=0.26,y=0.7,'Simplex');text(x=0.595,y=1.092,'2');
text(x=0.58,y=1.05,'R')
}
ternaryv(X/100, ord=c(1,3,2))

```

Script para a construção da Figura 2.2:

```

require(MASS)
data(Skye)
X <- Skye
ternaryp <- function(X,pch=par('pch'),lcex=1,add=FALSE,ord=1:3,...){
  limite <- read.table('limitePorcPaw.txt')
  x <- limite$V1
  y <- limite$V2
  X <- as.matrix(X)
  if(any(X<0)) stop('X must be non-negative')
  s <- drop(X %*% rep(1,ncol(X)))
  if(any(s<=0)) stop('each row of X must have a positive sum')
  if(max(abs(s-1))>1e-6){
    warning('rows(s) of will be rescaled')
    X <- X/s
  }
  X <- X[,ord]
  s3 <- sqrt(1/3)
  if(!add){
    oldpty <- par('pty')
    on.exit(par(pty=oldpty))
  }
}

```

```

par(ply='s')
par(mar=c(3.5,3.5,1.5,1.5),mgp=c(1.7,0.7,0))
plot(c(0.18,1.21),c(0.16,1.1),type='n',axes=TRUE,xlab='',ylab='')
## paralelas FM
segments(x0=0.65,y0=0.98,x1=0.75,y1=0.98,lty=3);text(x=0.81,y=0.99,'10%')
segments(x0=0.60,y0=0.89,x1=0.8,y1=0.89,lty=3);text(x=0.86,y=0.9,'20%')
segments(x0=0.55,y0=0.81,x1=0.85,y1=0.81,lty=3);text(x=0.91,y=0.82,'30%')
segments(x0=0.50,y0=0.72,x1=0.9,y1=0.72,lty=3);text(x=0.96,y=0.73,'40%')
segments(x0=0.45,y0=0.63,x1=0.95,y1=0.63,lty=3);text(x=1.01,y=0.63,'50%')
segments(x0=0.40,y0=0.55,x1=1,y1=0.55,lty=3);text(x=1.06,y=0.55,'60%')
segments(x0=0.35,y0=0.46,x1=1.05,y1=0.46,lty=3);text(x=1.11,y=0.46,'70%')
segments(x0=0.30,y0=0.37,x1=1.1,y1=0.37,lty=3);text(x=1.16,y=0.37,'80%')
segments(x0=0.25,y0=0.29,x1=1.15,y1=0.29,lty=3);text(x=1.21,y=0.29,'90%')
## paralelas AF
segments(x0=0.30,y0=0.20,x1=0.75,y1=0.98,lty=3);text(x=0.59,y=0.99,'90%')
segments(x0=0.40,y0=0.20,x1=0.8,y1=0.89,lty=3);text(x=0.54,y=0.9,'80%')
segments(x0=0.50,y0=0.20,x1=0.85,y1=0.81,lty=3);text(x=0.49,y=0.82,'70%')
segments(x0=0.60,y0=0.20,x1=0.9,y1=0.72,lty=3);text(x=0.44,y=0.73,'60%')
segments(x0=0.70,y0=0.20,x1=0.95,y1=0.63,lty=3);text(x=0.39,y=0.63,'50%')
segments(x0=0.80,y0=0.20,x1=1,y1=0.55,lty=3);text(x=0.34,y=0.55,'40%')
segments(x0=0.90,y0=0.20,x1=1.05,y1=0.46,lty=3);text(x=0.29,y=0.46,'30%')
segments(x0=1,y0=0.20,x1=1.1,y1=0.37,lty=3);text(x=0.23,y=0.37,'20%')
segments(x0=1.10,y0=0.20,x1=1.15,y1=0.29,lty=3);text(x=0.19,y=0.29,'10%')
## paralelas AM
segments(x0=0.65,y0=0.99,x1=1.10,y1=0.20,lty=3);text(x=0.29,y=0.17,'90%')
segments(x0=0.60,y0=0.89,x1=1.00,y1=0.20,lty=3);text(x=0.39,y=0.17,'80%')
segments(x0=0.55,y0=0.81,x1=0.90,y1=0.20,lty=3);text(x=0.50,y=0.17,'70%')
segments(x0=0.50,y0=0.72,x1=0.80,y1=0.20,lty=3);text(x=0.60,y=0.17,'60%')
segments(x0=0.45,y0=0.63,x1=0.70,y1=0.20,lty=3);text(x=0.70,y=0.17,'50%')
segments(x0=0.40,y0=0.55,x1=0.60,y1=0.20,lty=3);text(x=0.80,y=0.17,'40%')
segments(x0=0.35,y0=0.46,x1=0.50,y1=0.20,lty=3);text(x=0.90,y=0.17,'30%')
segments(x0=0.30,y0=0.37,x1=0.40,y1=0.20,lty=3);text(x=1.00,y=0.17,'20%')
segments(x0=0.25,y0=0.29,x1=0.30,y1=0.20,lty=3);text(x=1.10,y=0.17,'10%')
##p-13
segments(x0=0.6,y0=0.55,x1=0.60,y1=0.20,lty=1)
##p-24
segments(x0=0.6,y0=0.55,x1=0.9,y1=0.72,lty=1)
##p-5

```

```

    segments(x0=0.6,y0=0.55,x1=0.45,y1=0.63,lty=1)
    ##15-20
    segments(x0=0.80,y0=0.20,x1=1.1,y1=0.37,lty=1)
    polygon(c(0.7,0.2,1.2),c(0.2+sqrt(3)/2,0.2,0.2),density=0)
  }
  points(limite[,1], limite[,2],pch=3)
  points(0.6,0.55,pch=20)
  ##points((X[2]-X[3])*s3,X[1],pch=20)
  text(x=0.61,y=0.58,'p')
  text(x=0.63,y=0.39,'X1');text(x=0.75,y=0.67,'X2');text(x=0.52,y=0.62,'X3')
  text(x=0.95,y=0.32,'X2')
  text(x=0.20,y=0.15,'F');text(x=1.19,y=0.15,'M');text(x=0.70,y=1.10,'A')
}
ternaryp(X/100, ord=c(1,3,2))

```

Script para a construção da Figura 2.3:

```

require(MASS)
data(Skye)
X <- Skye
ternaryv <- function(X,pch=par('pch'),lcex=1,add=FALSE,ord=1:3){
  limite <- read.table('limitePorcVenables.txt')
  x <- limite$V1
  y <- limite$V2
  limite <- matrix(c(x,y),nc=2)
  X <- as.matrix(X)
  if(any(X<0)) stop('X must be non-negative')
  s <- drop(X %*% rep(1,ncol(X)))
  if(any(s<=0)) stop('each row of X must have a positive sum')
  if(max(abs(s-1))>1e-6){
    warning('rows(s) of will be rescaled')
    X <- X/s
  }
  X <- X[,ord]
  s3 <- sqrt(1/3)
  if(!add){
    oldpty <- par('pty')
    on.exit(par(pty=oldpty))
    par(pty='s')
  }
}

```



```

plot(c(-s3,s3),c(0.5-s3,0.5+s3),type='n',axes=TRUE,xlab='',ylab='')
### linhas partindo do vertice A
## linha da altura
segments(x0=0,y0=0,x1=0,y1=1,lty=1);text(x=0,y=-0.08,'H') #altura
points(x=0,y=0.80,pch=19);text(x=0.03,y=0.8, 'a')
points(x=0,y=0.60,pch=19);text(x=0.03,y=0.6, 'b')
points(x=0,y=0.40,pch=19);text(x=0.03,y=0.3, 'c')
points(x=0,y=0.20,pch=19);text(x=0.03,y=0.2, 'd')
## linha com ponto em 80%
segments(x0=0,y0=1,x1=-0.346,y1=0,lty=1)
points(x=-0.17,y=0.50,pch=19);text(x=-0.2,y=0.5, 'i')
points(x=-0.346,y=0,pch=19);text(x=-0.32,y=0.03, 'j')
## mais uma linha
segments(x0=0,y0=1,x1=-0.115,y1=0,lty=1)
points(x=-0.06,y=0.50,pch=19);text(x=-0.09,y=0.50,'e')
points(x=-0.09,y=0.20,pch=19);text(x=-0.12,y=0.20,'f')
points(x=-0.115,y=0,pch=19) ;text(x=-0.09,y=0.03,'g')
### linhas partindo do vertice F
segments(x0=-0.58,y0=0,x1=0.29,y1=0.50,lty=1)
points(x=-0.06,y=0.30,pch=19);text(x=-0.03,y=0.28,'n')
points(x=0.115,y=0.40,pch=19);text(x=0.15,y=0.39,'o')
points(x=-0.23,y=0.20,pch=19);text(x=-0.23,y=0.23,'m')
## linha com interceptando em 75%
segments(x0=-0.58,y0=0,x1=0.43,y1=0.25,lty=1)
points(x=-0.375,y=0.05,pch=19);text(x=-0.39,y=0.08,'q')
points(x=0.129,y=0.175,pch=19);text(x=0.16,y=0.16,'s')
# paralelas FM
text(x=0.36,y=0.6, 'L')
segments(x0=-0.057,y0=0.9,x1=0.057,y1=0.9,lty=3);text(x=0.11,y=0.9, '10%')
segments(x0=-0.115,y0=0.8,x1=0.115,y1=0.8,lty=3);text(x=0.17,y=0.8, '20%')
segments(x0=-0.173,y0=0.7,x1=0.173,y1=0.7,lty=3);text(x=0.23,y=0.7, '30%')
segments(x0=-0.231,y0=0.6,x1=0.231,y1=0.6,lty=3);text(x=0.29,y=0.6, '40%')
segments(x0=-0.289,y0=0.5,x1=0.289,y1=0.5,lty=3);text(x=0.34,y=0.5, '50%')
segments(x0=-0.346,y0=0.4,x1=0.346,y1=0.4,lty=3);text(x=0.40,y=0.4, '60%')
segments(x0=-0.404,y0=0.3,x1=0.404,y1=0.3,lty=3);text(x=0.46,y=0.3, '70%')
segments(x0=-0.462,y0=0.2,x1=0.462,y1=0.2,lty=3);text(x=0.52,y=0.2, '80%')
segments(x0=-0.519,y0=0.1,x1=0.519,y1=0.1,lty=3);text(x=0.57,y=0.1, '90%')
## paralelas AF

```

```

segments(x0=-0.46,y0=0,x1=0.057,y1=0.9,lty=3);text(x=-0.11,y=0.9,'90%')
segments(x0=-0.346,y0=0,x1=0.115,y1=0.8,lty=3);text(x=-0.17,y=0.8,'80%')
segments(x0=-0.231,y0=0,x1=0.173,y1=0.7,lty=3);text(x=-0.23,y=0.7,'70%')
segments(x0=-0.115,y0=0,x1=0.231,y1=0.6,lty=3);text(x=-0.29,y=0.6,'60%')
segments(x0=0,y0=0,x1=0.289,y1=0.5,lty=3);text(x=-0.34,y=0.5,'50%')
segments(x0=0.115,y0=0,x1=0.346,y1=0.4,lty=3);text(x=-0.40,y=0.4,'40%')
segments(x0=0.231,y0=0,x1=0.404,y1=0.3,lty=3);text(x=-0.46,y=0.3,'30%')
segments(x0=0.346,y0=0,x1=0.462,y1=0.2,lty=3);text(x=-0.52,y=0.2,'20%')
segments(x0=0.46,y0=0,x1=0.519,y1=0.1,lty=3);text(x=-0.57,y=0.1,'10%')
## paralelas AM
segments(x0=-0.057,y0=0.9,x1=0.46,y1=0,lty=3);text(x=-0.46,y=-0.04,'90%')
segments(x0=-0.115,y0=0.8,x1=0.346,y1=0,lty=3);text(x=-0.34,y=-0.04,'80%')
segments(x0=-0.173,y0=0.7,x1=0.231,y1=0,lty=3);text(x=-0.22,y=-0.04,'70%')
segments(x0=-0.231,y0=0.6,x1=0.115,y1=0,lty=3);text(x=-0.11,y=-0.04,'60%')
segments(x0=-0.289,y0=0.5,x1=0,y1=0,lty=3);text(x=0,y=-0.04,'50%')
segments(x0=-0.346,y0=0.4,x1=-0.115,y1=0,lty=3);text(x=0.12,y=-0.04,'40%')
segments(x0=-0.404,y0=0.3,x1=-0.231,y1=0,lty=3);text(x=0.22,y=-0.04,'30%')
segments(x0=-0.462,y0=0.2,x1=-0.346,y1=0,lty=3);text(x=0.34,y=-0.04,'20%')
segments(x0=-0.519,y0=0.1,x1=-0.46,y1=0,lty=3);text(x=0.46,y=-0.04,'10%')
polygon(c(0,-s3,s3),c(1,0,0),density=0)
}
points(limite[,1],limite[,2],pch=3)# ponto p
text(x=-0.57,y=-0.06,'F');text(x=0.58,y=-0.06,'M');text(x=0.0,y=1.03,'A')
}
ternaryv(X/100,ord=c(1,3,2))

```

Script para a construção da Figura 2.4:

```

require(MASS); require(compositions)
dados <- read.table('ArticoLakepont.txt')
comp <- acomp(dados)
names(comp) <- c("Areia", "Silte", "Argila")
plot(comp,pch=16)
plot(mean(comp),add=T,pch=19,col="red",cex=1.8)
# exibindo a região 2-sigma
ellipses(mean(comp),var(comp),col="red",r=2)

```