GVPESQUISA

DATA MINING POR MEIO DE ANÁLISE DE REDES, NO CONTEXTO DE FILTRO COLABORATIVO

Relatório 16/2005

FRANCISCO JOSÉ ESPÓSITO ARANHA FILHO

Agradecimentos

Gostaria de registrar meus agradecimentos:

- à Profa. Lúcia Barroso pela orientação, apoio e confiança;
- aos auxiliares de pesquisa Alfredo Roberto Jr, Priscila Rosa e Sandro Venezuela, cujo suporte foi decisivo na realização desta pesquisa;
- ao NPP – Núcleo de Pesquisas e Publicações da FGV-EAESP, pelo apoio financeiro para a realização do projeto.

Dissertação de Mestrado

A primeira parte deste trabalho foi submetida à banca examinadora do Instituto de Matemática e Estatística da Universidade de São Paulo, em fevereiro de 2003, como dissertação para obtenção do título de Mestre em Estatística, sob o título de "*Data Mining* em Grandes Redes: Superfícies de Coesão sobre Base Multidimensionalmente Escalonada" tendo sido aprovada com distinção.

Observação

O presente projeto de pesquisa foi realizado em duas fases. Um relatório parcial, referente à Primeira Etapa do projeto de pesquisa original, foi entregue em março de 2003 ao NPP e aprovado pelos seus revisores.

Este relatório final engloba o relatório parcial e o complementa. O conteúdo já submetido corresponde, grosso modo, ás seções 1 a 14. A seção 15 apresenta o conteúdo novo, desenvolvido na Segunda Etapa do projeto, que diz respeito à estabilidade das representações de redes produzidas por meio da aplicação do algoritmo RGR proposto. Finalmente, as seções 16 a 20, do fechamento, consolidam resultados e comentários relativos às duas etapas.

A retomada do relatório parcial teve por objetivo a produção de um documento final que representasse o projeto como um todo.

Resumo

Tendo como motivação o desenvolvimento de uma representação gráfica de redes com grande número de vértices, útil para aplicações de Filtro Colaborativo, este trabalho propõe a utilização de superfícies de coesão sobre uma base temática multidimensionalmente escalonada. Para isso, utiliza uma combinação de Escalonamento Multidimensional Clássico e Análise de Procrustes, em algoritmo iterativo que encaminha soluções parciais, depois combinadas numa solução global. Aplicado a um exemplo de transações de empréstimo de livros pela Biblioteca Karl A. Boedecker, o algoritmo proposto produz saídas interpretáveis e coerentes tematicamente, e apresenta um *stress* menor que a solução por Escalonamento Clássico. O estudo da estabilidade da representação de redes frente à variação amostral dos dados, realizado com base em simulações envolvendo 500 réplicas em 6 níveis de probabilidade de inclusão das arestas nas réplicas, fornece evidência em favor da validade dos resultados obtidos.

Palavras-Chave

Análise de Procrustes; Análise de Redes; Escalonamento Multidimensional; Data Mining; Link Analysis; Cooperação Indireta; Distância Temática; Filtro Colaborativo; Teoria dos Grafos.

Abstract

In this dissertation, a graphical representation of large networks based on the use of cohesion surfaces over a multidimensionally scaled thematic base is proposed as a tool for Collaborative Filtering. For its development Classic Multidimensional Scaling and Procrustes Analysis are combined in an iterative algorithm, which consolidates partial solutions into an overall continuous representation. Tested on a set of book lending transactions at the Karl A. Boedecker Library, the algorithm produces an output that is thematically interpretable and consistent, with a stress measure smaller than Classic MDS solutions. The study of representation stability in face of sampling uncertainty, based on a sampling simulation at 6 different levels of sampling probability and 500 replications for each level, provides evidence in support of algorithm results validity.

Key-Words

Collaborative Filtering; Data Mining; Graph Theory; Indirect Cooperation; Link Analysis; Multidimensional Scaling; Procrustes Analysis.

Sumário

AGRADECIMENTOS **1**

DISSERTAÇÃO DE MESTRADO **2**

OBSERVAÇÃO **3**

RESUMO **4**

PALAVRAS-CHAVE **4**

ABSTRACT **5**

KEY-WORDS **5**

1. ORGANIZAÇÃO DO TRABALHO 13
2. INTRODUÇÃO 14
2.1. Sobrecarga de Informação *14*
2.2. Respostas Tecnológicas à Sobrecarga de Dados *15*
2.2.1 Filtro Informativo (IF) 16
2.2.2 Filtro Colaborativo (CF) 17
3. PESQUISAS ANTERIORES 19
3.1. Projeto 1: Colaboração Indireta na Biblioteca Karl A. Boedecker (ARANHA, 2001a) *19*
3.2. Projeto 2: Análise de Redes no Sistema de Recomendações (ARANHA, 2001b) *20*
4. OBJETIVO DESTE TRABALHO 22
4.1. Superfície de Coesão sobre Base Multidimensionalmente Escalonada *22*
4.2. Criação de um Algoritmo Escalável para a Implementação da Nova Estratégia *24*

SEGUNDA PARTE: DEFINIÇÕES E TÉCNICAS ESTATÍSTICAS **25**

5. REDES 25
5.1. Definições e Notação *25*
*5.2. Análise de Redes Sociais (*Social Network Analysis*)* *26*
5.3. Problemas Clássicos e Contemporâneos com Estrutura de Rede *27*
5.3.1 Difusão de Inovações 27
5.3.2 Busca de Empregos 28
5.3.3 Internet 29
5.3.4 Cooperação Indireta: Clientes de Uma Livraria Virtual 30
6. ESCALONAMENTO MULTIDIMENSIONAL (MULTIDIMENSIONAL SCALING – MDS) 32
6.1. Tipos de Dados *32*
6.1.1 Escala de Medida 32
6.1.2 Número de Modos 32
6.1.3 Número de Direções 32
6.2. Modelos de Escalonamento Multidimensional *33*

6.2.1 Escalonamento Clássico 33
6.2.2 Escalonamento Métrico por Mínimos Quadrados 33
6.2.3 Escalonamento Não-Métrico 34
6.2.4 Análise Procrustes 34
6.2.5 Outros Modelos 34
7. ALGUNS DETALHES SOBRE MDS CLÁSSICO E PROCRUSTES 35
7.1. MDS Clássico *35*
7.1.1 Implementação da Função cmdscale no R 36
7.2. Procrustes *37*
7.2.1 Algoritmo de Procrustes 38
TERCEIRA PARTE: FORMULAÇÃO DO PROBLEMA E ENCAMINHAMENTOS 40
8. TRÊS ARQUIVOS COM DADOS PARA EXEMPLO 40
8.1. Exemplo 1 – EX1.dat *41*
8.2. Resumo Comparativo dos Três Arquivos de Exemplos *45*
9. FORMULAÇÃO GERAL DO PROBLEMA 47
9.1. Pré-Processamento da Lista de Transações *47*
9.2. Redes Bimodais e Redes Monomodais *49*
9.3. Cálculo e Armazenamento das Distâncias Temáticas *51*
9.3.1 Cálculo 53
9.3.2 Armazenamento 54
9.4. Cálculo da Coesão *54*
9.5. Representação da Rede Em Espaço Temático *56*
9.5.1 Arestas como Distâncias Temáticas 58
9.6. Substituição de Arestas por Superfície de Coesão *62*
10. BASE ESCALONADA 66
10.1. Quebrando o Problema em Partes Hierarquizadas (Agrupando de Baixo para Cima) *67*
10.2. Coleta das Listas de Arestas *70*
10.2.1 dp0p0 71
10.2.2 dp1p0 71
10.2.3 dp1p1 71
10.2.4 Níveis Mais Agregados 72
10.3. Escalonando (de Cima para Baixo) *72*
10.3.1 Configuração do Último Nível (Nível Mais Elevado) 72
10.3.2 Triangulando e Encontrando Soluções Parciais 74
10.3.3 Juntando as Soluções Parciais 80
11. SUPERFÍCIE DE COESÃO 83
11.1. Interpretação do Relevo *85*
11.2. Estratégia de Implementação da Representação Gráfica *88*
12. PARTICULARIDADES DO PROBLEMA DE COOPERAÇÃO INDIRETA 90
12.1. Matriz de Adjacência Escassa *90*
12.2. Matriz de Distâncias Temáticas Escassa *91*
12.3. Distribuição das Distâncias Temáticas *91*

12.3.1 Matriz D de Distâncias Temáticas 91
12.4. Relevância Local *97*
13. CÓDIGO COMENTADO 99
13.1. R *99*
13.1.1 Vantagens 99
13.1.2 Desvantagens 99
13.2. Fluxograma *101*
13.2.1 Pré-Processamento 101
13.2.2 Lista Vizinhos e Distância Temática 101
13.2.3 "Sobe" 102
13.2.4 "Desce" 102
13.3. Pseudo Código *103*
13.3.1 Pré-Processamento 103
13.3.2 Lista Vizinhos e Distância Temática 103
13.3.3 "Sobe" 103
13.3.4 "Desce" 104
14. COMPARAÇÃO DE DUAS SOLUÇÕES: MDS CLÁSSICO E ALGORITMO PROPOSTO 105
14.1. Para a Aplicação, o MDS Clássico Não É Boa Referência *105*
14.1.1 MDS Clássico Fornece Solução Praticamente Arbitrária 106
14.1.2 Algoritmo Proposto Produz Stress Menor Onde Interessa 107
QUARTA PARTE: ESTABILIDADE DA SOLUÇÃO 110
15. ESTABILIDADE DAS REPRESENTAÇÕES DE REDES PRODUZIDAS PELA APLICAÇÃO DO ALGORITMO RGR 110
15.1. Estratégia Global da Averigução da Estabilidade da RGR *110*
15.2. "População" *111*
15.3. Processo de amostragem *116*
15.4. Número de Réplicas *118*
15.5. "Passeio" e Coesão de Usuários Selecionados *120*
15.5.1 "Regressão" das Coordenadas do Usuário para o Centróide da Configuração 134
15.5.2 "Regressão" da Coesão Estimada para 0,5 134
15.5.3 Consistência 134
15.6. Relevo de Coesão das Amostras *134*
15.6.1 Forma cônica ou cilíndrica para superfícies com base em amostras pequenas 147
15.7. Síntese do Seção *148*
QUINTA PARTE: FECHAMENTO 149
16. CONCLUSÃO 149
16.1. Data Mining: Problemas e Soluções na Representação de Redes *149*
17. CONTRIBUIÇÕES DO TRABALHO 151
18. LIMITAÇÕES 152
18.1. Triangulação *152*
18.2. Informática, Interface *152*

18.3. Técnica Descritiva *153*
18.4. Isolamento da Base de Dados *153*
18.5. Desconexão de Interface Gráfica *153*
18.6. Função Geradora de Probabilidade Sensível ao Grau do Nó *153*
19. PRÓXIMOS PASSOS 154
19.1. Aperfeiçoamentos Planejados *154*
19.1.1 Triangulação – Remoção de *Bug* Conhecido e Aproveitamento de Redundância 154
19.1.2 Iteração no Cálculo da Distância Temática 154
19.1.3 Aperfeiçoamento do Ajuste por Procrustes 154
19.1.4 Ajuste Fino: Uma Janela Voadora 154
19.1.5 Stress Deve ser Indiferente a Distorções de $\delta_{ij} = 1$ para $d_{ij} > 1$ 155
19.2. Idéia Novas para Interpretação dos Resultados *155*
19.2.1 Delimitação Automática do Relevo 155
19.2.2 Medidas de Interesse 155
19.2.3 Identificação de "Conectores" 155
20. ANEXO: LISTAGEM DA IMPLEMENTAÇÃO EM R 156
20.1. Funções do Algoritmo RGR (Representação de Grandes Redes) *156*
20.1.1 Deduplique 156
20.1.2 Remova Cílios 157
20.1.3 Vértices 158
20.1.4 Lista Vizinhos 159
20.1.5 Cria Transação 161
20.1.6 Lista Vizinhos Simples 162
20.1.7 Distância Temática 163
20.1.8 Agrupa 164
20.1.9 Distância Pai – Filho 166
20.1.10 Monta Matriz 168
20.1.11 Distância entre Centróides 169
20.1.12 Desce 171
20.1.13 Triângulo 173
20.1.14 Reúne 179
20.1.15 Monta Matriz Simples 181
20.1.16 Arruma Posição 182
20.1.17 Traço 183
20.1.18 Procrustes 184
20.1.19 Ajustar 185
20.1.20 Suaviza Dados 186
20.2. Corpo do Programa para RGR do Exemplo 2 ..... *188*
20.3. Funções para Estudo da Estabilidade do Algoritmo de RGR *192*
20.3.1 Algoritmo RGR (função f.alg.chico) 192
20.3.2 Algoritmo para Amostragem (f.alg.amostra) 195
20.3.3 Agoritmo para Inferência (f.inferência) 197
20.3.4 Lista Vizinhos dos Livros em um Passo 198
20.3.5 Calcula Distância entre Usuários e Livros 199

20.3.6 Amostra da TO 200
20.3.7 Constrói Gráficos 201
20.3.8 Mostra Distribuição de Distância 202
20.3.9 Mostra Distribuição de Probabilidades 202
20.4. Corpo do Programa para Estudo da Estabilidade do Algoritmo RGR 203
21. BIBLIOGRAFIA 206

Figuras

Figura 1 – Espaço Temático. 23
Figura 2 – Relevo de Subgrupos Especializados. 24
Figura 3 – Interface Principal do Sistema de Recomendações da Amazon.com. 30
Figura 4 – Representação Gráfica da Rede do Exemplo 1. 43
Figura 5 – Exemplo de Remoção de Cílio..... 48
Figura 6 – Fluxograma do Pré-Processamento 49
Figura 7 – Matrizes de Adjacência A e A^2 do Exemplo 1, ... 50
Figura 8 – Representação Gráfica da Rede de Usuários do Exemplo 1. 51
Figura 9 – Diversas Representações de um Grafo de Ordem 10 e Tamanho 15..... 57
Figura 10 – Rede do Exemplo 1 em Espaço Temático (sem Legenda de Pontos) 59
Figura 11 – Rede do Exemplo 1 em Espaço Temático (com Legenda de Pontos)..... 60
Figura 12 – Rede do Exemplo 2 em Espaço Temático (sem Legenda de Pontos) 61
Figura 13 – Rede do Exemplo 2 em Espaço Temático (com Legenda de Pontos)..... 62
Figura 14 – Superfície de Coesão do Exemplo 1. 64
Figura 15 – Superfície de Coesão do Exemplo 2. 65
Figura 16 – Alocação de Vértices a Grupos e Criação de Representantes. 68
Figura 17 – Redução do Número de Vértices em Iterações Sucessivas. 68
Figura 18 – Arquivo de Exemplo 2: Vértices da Rede 70
Figura 19 – Centróides de Nível 1 do Exemplo 2, com Rótulos de Vértices. 73
Figura 20 – Triangulação dos Vértices do Nível 1 do Exemplo 2. 76
Figura 21 – Configuração no Triângulo 1 do Exemplo 2 80
Figura 22 – Configuração Antes do Ajuste 82
Figura 23 – Configuração Depois do Ajuste. 82
Figura 24 – Relevo Temático do Exemplo 2..... 84
Figura 25 – Legenda dos Vértices da Rede do Exemplo 2. 85
Figura 26 – Retícula sobre os Vértices do Exemplo 2. 89
Figura 27 – Histograma das Distâncias Temáticas entre Usuários do Exemplo 1..... 93
Figura 28 – Histograma das Distâncias Temáticas do Exemplo 2..... 95
Figura 29 – Histograma das Distâncias Temáticas do Exemplo 3..... 97
Figura 30 – MDS Clássico para k = 2..... 105
Figura 31 – Algoritmo Proposto..... 105
Figura 32 – Scree-plot dos Componentes do Exemplo 2 107
Figura 33 – Distribuição do *Stress*..... 108
Figura 34 – Posição "Verdadeira" dos Vértices de Usuários e de Livros Integrantes da "População" Formada pelos Nós do Exemplo 2. 112
Figura 35 – Posicionamento dos Livros em Relação aos Usuários. 114
Figura 36 – Histograma das Distâncias Entre Usuários Selecionados e Todos os Livros da População. 116
Figura 37 – Histograma das Probabilidadesde Livros Serem Sorteados pra o Usuário 1347, Em Vários Níveis de α 118
Figura 38 – Histograma do R^2: 500 Réplicas por Nível de α 120

Figura 39 – "Passeio" de Usuários Selecionados. 133
Figura 40 – Representação da Superfície de Coesão para 9 Réplicas em Cada Nível de α. 146

Tabelas

Tabela 1 – Principais Características das Técnicas de IR, IF e CF 16
Tabela 2 – Resumo Comparativo dos Arquivos de Exemplos 45
Tabela 3 – Distâncias do Primeiro "Triângulo" 77
Tabela 4 – Configuração Processada no Triângulo 1 do Exemplo 2. 79
Tabela 5 – Livros Mais Retirados no "Morro Sudoeste" 86
Tabela 6 - Livros Mais Retirados no "Morro Sudeste" 87
Tabela 7 - Livros Mais Retirados no "Morro Norte" 88

Saídas

Saída 1 – Conteúdo do Arquivo EX1.dat 42
Saída 2 – Arquivo EX1.dat Pré-Processado 43
Saída 3 – Matriz de Adjacência do Exemplo 1. 45
Saída 4 – Lista de Vizinhos dos Nós de Usuários no Exemplo 1. 52
Saída 5 – Distâncias Temáticas entre Usuários do Exemplo 1 54
Saída 6 – Coesão dos Vértices do Exemplo 1 56
Saída 7 – Matriz **D** de Distâncias (Temáticas) entre Usuários do Exemplo 1. 92
Saída 8 – Distâncias Temáticas entre Nós de Usuários do Exemplo 2. 94
Saída 9 – Distâncias Temáticas entre Nós de Usuários do Exemplo 3 96

Data Mining por Meio de Análise de Redes, no Contexto de Filtro Colaborativo

Francisco José Espósito Aranha Filho

Primeira Parte: Informações Preliminares

1. ORGANIZAÇÃO DO TRABALHO

O presente trabalho está organizado em cinco partes. Na **primeira**, apresentamos o problema do ponto de vista da aplicação e do contexto em que ele surgiu; na **segunda**, fixamos notação e algumas definições e revisamos sucintamente duas técnicas estatísticas (MDS Clássico e Procrustes Analysis) utilizadas na solução.

Em seguida, na **terceira** parte encaminhamos a solução computacional; e na **quarta**, avaliamos a estabilidade da representação da rede por meio da representação contínua. Finalmente, na **quinta parte** apresentamos nossas conclusões, limitações do trabalho e possíveis desdobramentos.

2. INTRODUÇÃO

Com o avanço da informática e das telecomunicações, o volume de informações disponíveis sobre todos os aspectos da atividade humana, e em todos os ramos da ciência, passou a crescer a taxas vertiginosas:

> "Nos últimos trinta anos, produziu-se um volume de informações novas maior do que [o volume produzido] nos cinco mil anos precedentes. Cerca de mil livros são publicados no mundo por dia, e o total do conhecimento impresso duplica a cada oito anos." (LARGE, 1984)

2.1. Sobrecarga de Informação

O aumento da capacidade de se analisarem estes dados não é proporcional ao crescimento da oferta. A possibilidade de acesso dos usuários aos dados é limitada, quando não por outros motivos, ao menos pela finitude do tempo disponível para examiná-los; assim, a maior parte do conteúdo disponível permanece sem utilização, e mesmo quando as informações chegam a ser escrutinadas, encontram uma capacidade humana de absorção e entendimento também restrita. Uma estimativa eloqüente indica, por exemplo, que o leitor contemporâneo de um jornal de qualidade recebe mais informação em um único domingo do que um cidadão vivendo na Inglaterra do século XVII recebia ao longo de toda a sua vida (WURMAN, 1991). No mesmo período a expectativa de vida de um inglês passou de cerca de 45 para cerca de 80 anos.

Como resposta ao desafio apresentado pela sobrecarga de dados, o ser humano aplica suas habilidades cognitivas, lingüísticas e sociais no desenvolvimento de estratégias para fazer avaliações rápidas de relevância e escopo: cotidianamente, por exemplo, julgam-se livros pela capa, artigos pelo *abstract* e filmes pelo *trailer*. Os jornais de domingo são lidos apenas muito "na diagonal". Em Marketing, sabe-se que, como parte do processo decisório de compra de produtos, o consumidor envolve-se numa coleta de informações limitada e seletiva, apoiando sua decisão final em uma fração pequena do total de informações disponíveis (NEWMAN e STAELIN, 1972; MIDGLEY, 1983; BEATTY e SMITH, 1987). Estas são todas estratégias racionais, em que se troca eficácia por eficiência, a taxas amplamente

vantajosas. Ademais, diversas formas de inteligência artificial têm sido desenvolvidas e utilizadas no tratamento destes dados.

2.2. Respostas Tecnológicas à Sobrecarga de Dados

Há três tipos de tecnologia usualmente empregados para minorar os efeitos da sobrecarga de dados (GOOD, SCHAFER e OUTROS, 1999), cada um focalizado num conjunto distinto de aspectos do problema. A **Recuperação de Informação** (IR, *Information Retrieval)* atende interesses efêmeros, respondendo a consultas *ad hoc*, do tipo "quantos clientes compraram sabão em pó em nossas lojas da zona sul nos últimos três meses?" O **Filtro Informativo** (IF, *Information Filtering*) trata de tarefas ligadas à classificação de novos fluxos de conteúdo em categorias pré-definidas ("Sobre qual assunto versa este novo livro do acervo?") e ligadas à comparação do ajuste destes conteúdos ao perfil de interesse dos usuários ou consumidores ("Quem se interessa por livros deste assunto?"). Finalmente, o **Filtro Colaborativo** (CF, *Collaborative Filtering*) dedica-se a produzir indicadores da relevância localizada de itens de informação, com base na análise da manipulação destes itens por grupos de usuários com interesses especializados ("Que usuários tomam emprestados muitos livros em comum? Apresentemos a cada um no grupo os livros não consultados por ele, mas presente no acervo do grupo". A Tabela 1, a seguir, resume as principais características destas três técnicas; nas subseções 2.2.1 e 2.2.2 as técnicas são discutidas (com exceção das características da IR, que não são de interesse neste trabalho).

Tabela 1 – Principais Características das Técnicas de IR, IF e CF

Técnica	Participação do Usuário	Principais Procedimentos	Traços Distintivos
Information Retrieval (IR)	Ativa, na formulação das consultas	Indexação de documentos segundo várias chaves	Não captura preferências do usuário
Information Filtering (IF)	Geralmente ativa, na criação de perfil de interesses ou necessidades	Análise e classificação de conteúdos; comparação do perfil do conteúdo com o perfil do usuário	O número de usuários é irrelevante; eficaz na circulação de novos itens de informação; ineficaz na diversificação do conteúdo sugerido
Collaborative Filtering (CF)	Passiva, os usuários do sistema não precisam fornecer *inputs*; o sistema, encarrega-se de obter os *inputs* a partir da observação do comportamento dos usuários	Análise e classificação do comportamento dos usuários; consolidação de listas de itens com relevância local	Exige um número grande de usuários; não analisa o conteúdo dos itens; lento na circulação de novos itens de informação; eficaz na diversificação de conteúdos

2.2.1 Filtro Informativo (IF)

O IF, também chamado de Filtro de Conteúdo, é uma das técnicas já tradicionais na seleção de informações relevantes dentro de um conjunto volumoso de dados disponíveis (SMYTH e COTTER, 2000). Seu sucesso apoia-se na habilidade de

- **representar acuradamente cada item** de informação no acervo, com base em um subconjunto de suas características, principalmente seu enquadramento em uma tipologia temática; e de
- **representar o interesse do usuário** através de um perfil baseado no mesmo subconjunto de características extraídas dos itens.

Neste contexto, a relevância de um item para um usuário é proporcional **à similaridade do perfil do item ao perfil do usuário**; os itens selecionados para serem submetidos à atenção do usuário são os mais parecidos com o seu perfil de interesse.

O problema da utilização de Filtros Informativos é justamente a necessidade de se caracterizarem os itens e o interesse dos usuários: esta atividade é complexa, onerosa, e requer conhecimento especializado sobre cada campo do conhecimento humano. Numa biblioteca, por exemplo, a caracterização dos itens corresponde ao processo de classificação, indexação e catalogação dos novos livros ou periódicos. Para usufruir do sistema de IF, o usuário também necessita informar ativamente, e com precisão, o perfil do seu interesse.

Mesmo concluída com sucesso a etapa de caracterização dos itens e do perfil do usuário, as recomendações por meio de IF terão sempre um escopo limitado, uma vez que esta técnica somente seleciona itens em que as duas caracterizações se aproximam. Como o perfil do usuário é explicitamente construído num ponto fixo do tempo (e raramente atualizado), ou inferido com base no registro de escolhas anteriores, **um subespaço temático tende a ser reforçado**, principalmente no caso de usuários novos para o sistema. Os itens recomendados podem ser relevantes, mas nem de longe correspondem à amplitude total dos interesses do usuário. Por estes motivos, o sistema tampouco permite **descobertas serendipitosas** (*serendipitous*), isto é, descobertas que se fazem sem que se procure pelo item encontrado.

2.2.2 Filtro Colaborativo (CF)

Em contraposição ao Filtro Informativo, as técnicas de Filtro Colaborativo movem-se para além da experiência de cada usuário isolado: procuram valer-se das experiências de **grupos** de usuários. Em vez de identificar a semelhança entre usuários e itens, **busca localizar usuários parecidos entre si**. Sendo parecidos os membros de um grupo, o que se descobre pelo monitoramento de seu comportamento no sistema, infere-se que os itens de interesse para um membro do grupo também serão de interesse para os demais membros.

A grande vantagem desta estratégia é **não ser necessário entrar no mérito do conteúdo dos itens**. O grande problema é que itens novos, conhecidos por poucos usuários, demoram a aparecer como sugestões, isto é, novos itens têm um longo período de latência. Assim, por exemplo, um grande fã de Woody Allen teria que esperar até um novo filme deste diretor ter sido assistido por muitos membros de seu grupo, para só então o filme surgir como recomendação. Um sistema de Filtro Informativo detectaria a proximidade de perfis (filme de Woody Allen/gosto por filmes de Woody Allen) imediatamente, fazendo a recomendação independentemente do filme já ter sido muito assistido.

Também fora do domínio da criação de sistemas de recomendação, várias iniciativas de implementação de Filtro Colaborativo foram documentadas recentemente. PAYTON (1998) procurou facilitar o contato entre pessoas com interesses comuns, explicitando seu padrão de navegação na Internet. KAUTZ, SELMAN e SHAH (1997a, 1997b) caracterizaram redes de pesquisadores que mantinham vínculos sociais, analisando a co-ocorrência de nomes em documentos públicos na Internet. SCHWARTZ e WOOD (1993) identificaram colaboradores potenciais pela análise do tráfego de emails em pontos selecionados da rede. SWANSON e SMALHEISER (2000) desenvolveram o software *Arrowsmith* para identificar relações pouco evidentes entre achados científicos na área de biomédicas.

Caminhando em direção oposta, isto é, perseguindo o objetivo de **dificultar** a cooperação, foram publicados vários trabalhos na área de detecção de fraudes contra companhias seguradoras e de saúde (CABENA e OUTROS, 1997) combate à lavagem de dinheiro e combate ao crime organizado em geral (JENSEN, 1999; HANN 1998).

3. PESQUISAS ANTERIORES

O problema de pesquisa a ser abordado neste trabalho materializou-se no decorrer de dois projetos de Filtro Colaborativo anteriormente realizados. Para esclarecer-lhe o contexto, convém reportar brevemente os projetos precedentes.

3.1. Projeto 1: Colaboração Indireta na Biblioteca Karl A. Boedecker (ARANHA, 2001a)

No esforço de reagir à perda da posição de fornecedor privilegiado de informações, decorrente do avanço de fontes alternativas como a Internet, as bibliotecas precisam reformular seu papel e seus objetivos (CARSON e OUTROS, 1997). Dentre as mudanças necessárias, destacam-se (Stephen Abram, segundo MILLER, 1998):

- o reposicionamento da biblioteca como **mapeadora e auditora de conteúdos** internos e externos à organização, e não como mera **gerenciadora do acervo**; e
- a transferência do foco das operações **para o usuário e seu comportamento**; esta mudança se manifesta através do cultivo de relacionamentos pró-ativos e personalizados com os usuários, segundo os princípios do marketing um-a-um (PEPPERS e ROGERS, 1999).

Interessada em investigar as possibilidades concretas de iniciativas imbuídas deste espírito de reformulação, a Biblioteca Karl A. Boedecker (KAB) apoiou, ao longo de 1999, a realização de um projeto piloto de Sistema de Recomendações de títulos aos usuários. A idéia básica era de que o sistema deveria pró-ativamente chamar a atenção do usuário para itens que inferisse serem do seu interesse.

Para o teste do protótipo do sistema foram analisadas 22.500 transações de empréstimo ocorridas entre 01/03 e 19/07/1999 e selecionados 410 usuários entre professores e alunos de pós-graduação para a geração de recomendações. SANTOS (2000) desenhou e implementou o necessário *data mart* das transações.

O protótipo resultante do Projeto 1 foi organizado em torno de duas funcionalidades críticas (ARANHA, 2000):

- a identificação de Assuntos Significativos (AS), que são grandes categorias temáticas dos itens do acervo; e
- a formação de Subgrupos Especializados (SGE), de usuários com interesses semelhantes.

Estes dois núcleos funcionais foram implementados, respectivamente, por meio de técnicas de Análise de Cestas (BERRY e LINOFF, 1997 e 2000) e de Análise de Agrupamentos (BUSSAB, MIAZAKI e ANDRADE, 1990; HAIR, TATHAM e OUTROS, 1991; JOHNSON e WICHERN, 2002).

3.2. Projeto 2: Análise de Redes no Sistema de Recomendações (ARANHA, 2001b)

No período de abril a dezembro de 2000, foi realizado um segundo projeto de pesquisa ligado ao Sistema de Recomendações da Biblioteca KAB, com objetivo de substituir as técnicas de Análise de Agrupamentos empregadas no "motor" da funcionalidade de criação dos Subgrupos Especializados por técnicas de Análise de Redes, conforme sugerido na bibliografia de Filtro Colaborativo (por exemplo, em SCHWARTZ e WOOD, 1993).

Na abordagem por Análise de Redes, a movimentação de consultas ao acervo da biblioteca é representada graficamente por meio de um diagrama de rede, em que livros e usuários são representados por nós (ou vértices) e as transações de empréstimo, por arestas. Alternativamente, a rede pode ser representada por uma matriz de adjacência em que linhas e colunas são nós (livros e usuários), e as arestas (transações de empréstimo) são representadas por números nas caselas da matriz. Por hora, basta saber que vértices e arestas são os objetos por meio dos quais estruturas de dados de interesse na aplicação de Filtro Colaborativo podem ser representados. O leitor não familiarizado com os conceitos de grafo ou rede talvez deseje folhear a seção 5, onde o tema de redes é tratado em maior detalhe. Ali, a Figura 4, na página 43, traz um exemplo de rede e a Figura 7, na página 50, traz um exemplo de matriz de adjacência.

No segundo projeto da biblioteca KAB, uma medida de distância temática (ARANHA, 2001b), inicialmente desenvolvida para variáveis dicotômicas, foi generalizada e aplicada a matrizes de co-ocorrências (ARANHA, 2001b), permitindo o aproveitamento de toda a informação disponível sobre o

comportamento dos usuários com relação aos itens consultados. O cálculo das distâncias temáticas foi realizado por meio de rotinas escritas no software S-Plus, suficientes para o processamento de lotes de até (estimados) 10.000 usuários.

Como resultado das alterações no Sistema de Recomendações, os usuários foram reclassificados em novos Subgrupos Especializados, para os quais Listas-Base e Listas Personalizadas (ARANHA, 2001b) foram geradas segundo o procedimento definido no projeto anterior.

Por um lado, o "motor" da funcionalidade de geração de SGE foi considerado um avanço com relação ao mecanismo anterior, e suficiente para as necessidades da KAB, uma vez que a biblioteca tem cerca de 6.000 usuários ativos. Por outro lado identificou-se na estratégia adotada uma fragilidade: o tamanho máximo do grupo de usuários passível de análise (10.000 usuários) era relativamente pequeno, principalmente quando se consideram os volumes típicos de usuários de aplicações para a Internet. O limite decorria do fato de que a necessidade de arquivo e memória RAM para processamento das rotinas do "motor" de SGE aumentava exponencialmente conforme crescia o número de usuários em análise.

No final de 2000, com o aparecimento da oportunidade de aplicação do Sistema de Recomendações desenvolvido nos Projetos 1 e 2 aos clientes de uma grande livraria virtual brasileira, o limite de processamento foi rapidamente atingido e mostrou-se uma restrição inaceitável em situação real de mercado. O sistema comprovou-se não escalável, tornando inviável sua utilização com os recursos computacionais disponíveis, quando o número de clientes processados se aproximou dos previstos 10.000 usuários.

4. OBJETIVO DESTE TRABALHO

O objetivo deste trabalho é **propor e testar uma nova estratégia de utilização de Análise de Redes** (AR) como técnica de Filtro Colaborativo, oferecendo alternativas para superar **os problemas de explosão combinatória** enfrentados na sua aplicação a grafos com elevado número de vértices.

Em particular, pretendemos:

- utilizar medidas caracterizadoras dos nós de uma rede (em termos de conectividade em sua vizinhança e de proximidade temática com outros nós) para criar uma representação contínua da rede (subseção 4.1, a seguir); e

- desenvolver um algoritmo que permita realizar o escalonamento multidimensional de um grande número de vértices, procedimento necessário à implementação da subseção anterior (veja discussão em 4.2, a seguir).

4.1. Superfície de Coesão sobre Base Multidimensionalmente Escalonada

O estudo da Teoria dos Grafos (veja seção 5) e a experiência adquirida nas pesquisas descritas na seção 3, sugerem a conveniência da utilização conjunta de duas medidas de Análise de Redes na implementação do motor do módulo de Subgrupos Especializados (SGE). Estas medidas, definidas com maior detalhe na seção 5, são:

- d_{ij}, isto é, a distância (temática) entre dois usuários; e
- γ_i, isto é, a coesão na vizinhança do vértice *i*.

A distância temática d_{ij} será utilizada para posicionar, através de Escalonamento Multidimensional (MDS, veja seção 6), os usuários num plano chamado de "espaço de semelhanças temáticas". Ou seja, a partir do conhecimento da distância entre os vértices, procurar-se-á recompor o "mapa" que revela a sua posição relativa. A Figura 1 exemplifica um espaço temático hipotético para 5 usuários de uma biblioteca, considerando-se como nós da rede os usuários e os livros do acervo; e,

como arestas, a ocorrência de uma transação de empréstimo, pela qual o usuário num extremo da aresta retirou o livro no outro extremo.

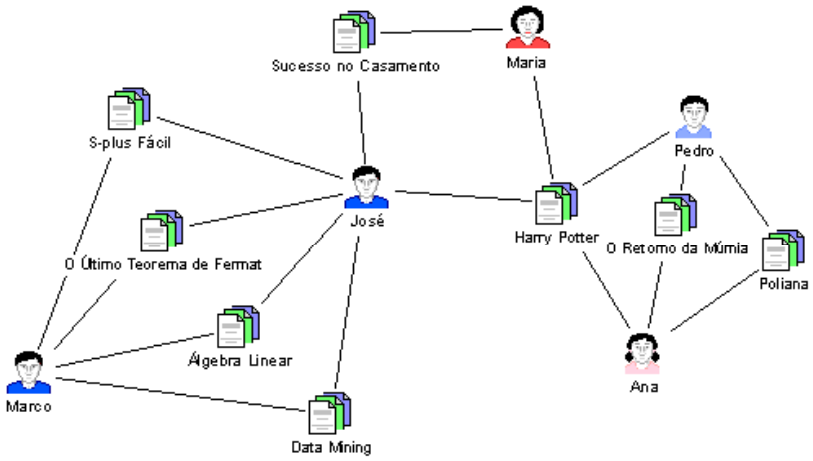


Figura 1 – Espaço Temático.

A outra medida, γ_i será utilizada para formar uma superfície de coesão sobre a base temática. O exemplo hipotético na Figura 2, a seguir, apresenta uma rede em que somente os usuários de uma biblioteca são representados: a cada nó observado da rede (isto é, a cada usuário em análise, cujas coordenadas no "espaço temático" nesta etapa do algoritmo já serão conhecidas) será associada uma "altura" correspondente ao seu γ_i. Os pontos do espaço temático para os quais não foram observados vértices terão suas alturas estimadas por interpolação.

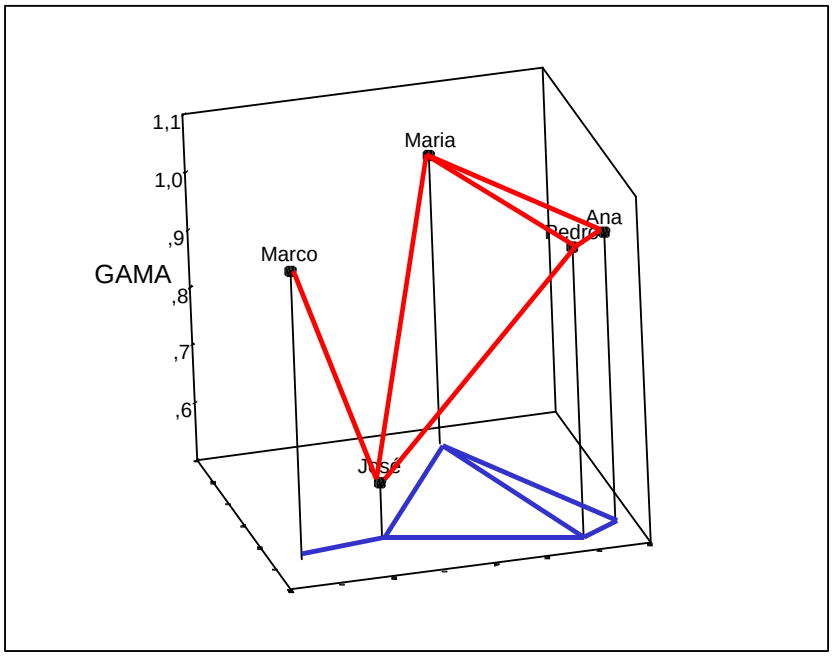


Figura 2 – Relevo de Subgrupos Especializados.

4.2. Criação de um Algoritmo Escalável para a Implementação da Nova Estratégia

O cálculo de d_{ij} e da aplicação de MDS caracterizam-se por serem pouco escaláveis, isto é, um aumento no número de nós a serem analisados leva a uma explosão no número de cálculos necessários, tornando-os inviáveis em termos de capacidade computacional e tempo de processamento. Assim, **será necessário propor um algoritmo** cujo consumo de recursos cresça de forma viável com relação ao aumento do número de usuários a serem analisados.

Para isso, utilizaremos a estratégia de programação de "dividir para conquistar" (veja, por exemplo, AHO, HOPCROFT e ULLMAN, 1974), e particularidades do tipo de rede de interesse para o problema de Filtro Colaborativo (principalmente o fato da matiz de adjacência ser escassa e da distribuição de distâncias ter muita massa à direita – veja subseção 8.2).

Segunda Parte: Definições e Técnicas Estatísticas

5. REDES

Muitas estruturas do mundo real podem ser representadas por **pontos** (também chamados nós ou vértices) e **linhas** (elos ou arestas). Uma abstração matemática destas estruturas leva ao conceito de **grafos** ou **redes**. TRUDEAU (1993) e WILSON (1996), entre outros, produziram textos de introdução ao assunto. BOLLOBÁS (1998) apresenta uma excelente síntese da teoria dos grafos, numa abordagem matematicamente mais formal; e em outro trabalho (BOLLOBÁS, 1985), sistematiza a teoria dos grafos aleatórios, inicialmente desenvolvida por Solomonoff e Rapoport em 1951 (BARABASI 2002) e por Erdos e Renyi em oito famosos artigos publicados entre 1959 e 1968 (GRAHAM e NESETRIL, 1997; CLUNG e GRAHAM 1998). BALAKRISHNAN (1997) produziu um resumo da teoria, do qual adaptamos a notação e as definições que se seguem (em BALAKRISHNAN, freqüentemente as definições são mais gerais do que é útil para o problema em análise; nestes casos, foram adotadas definições mais restritivas – e mais simples).

5.1. Definições e Notação

Um **grafo** G consiste em um conjunto V de **vértices** e uma coleção E de pares (não ordenados e com elementos distintos) de vértices chamados de **arestas**. Neste trabalho, V e E são considerados finitos. A **ordem,** #(V), de um grafo é o seu número de vértices e seu **tamanho,** #(E), é o seu número de arestas.

Se os vértices u e v são unidos pela aresta e, dizemos que a aresta e **incide** sobre u e v; e que os vértices u e v **incidem** sobre a aresta e.

Dois grafos G e G' são considerados **idênticos** se $V = V'$ e $E = E'$. E são considerados **isomórficos** se existe uma correspondência biunívoca f de V em V', chamada **isomorfismo**, tal que exista uma aresta entre $f(u)$ e $f(v)$ em G' se, e somente se, existir uma aresta entre u e v em G. Para todos os efeitos práticos, dois grafos isomórficos podem ser considerados como um mesmo grafo.

O grafo $H = (W, F)$ é um **subgrafo** de $G = (V, E)$ se W é subconjunto de V e F é subconjunto de E.

Lembrando que a definição adotada de grafo exclui a possibilidade de existência de uma alça fechada (*loop*), isto é uma aresta cujos dois extremos incidam no mesmo vértice, define-se **grau** de um vértice como o número de arestas nele incidentes.

Se $G = (V, E)$ é um grafo em que $V = \{1, 2, ... , n\}$, a **matriz de adjacência A** do grafo é a matriz n x n $\mathbf{A} = [a_{ij}]$, onde a_{ij}, assume o valor 1 se os vértices i,j são ligados por uma aresta (diz-se: são vizinhos, ou são adjacentes); e a_{ij} assume o valor 0, caso contrário; como não existem *loops*, a diagonal de **A** é formada por zeros.

Define-se Γ_v, **vizinhança** do vértice v pertencente a G, como o subgrafo de G que contém todos os vértices adjacentes a v e todas as arestas existentes entre estes; note que o próprio v não faz parte de sua vizinhança (WATTS, 1999, pág 31). A **coesão** da vizinhança de v (chamada de coeficiente de agrupamento, *clustering coefficient*, por Watts), representada por γ_v, é dada pelo número de arestas existentes na vizinhança dividido pelo número de arestas possíveis na vizinhança, ou seja, a proporção das arestas possíveis que é, de fato, observada (WATTS, 1999, pág.33).

Finalmente, entende-se por d_{ij}, distância (temática) entre dois vértices de G, a quantidade $1- [\#(\Gamma_i \cap \Gamma_j)/\#(\Gamma_i \cup \Gamma_j)]$, conceito adaptado de SCHWARTZ e WOOD (1993).

*5.2. Análise de Redes Sociais (***Social Network Analysis***)*

As técnicas de Análise de Redes (veja, por exemplo, KNOKE e KUKLINSKY, 1982; WASSERMAN e FAUST, 1994; WASSERMAN e GALASKEWICZ, 1994) **caracterizam um objeto de interesse pelas suas relações com os demais objetos** em estudo, em vez de os caracterizar por seus atributos individuais. Na área de Ciências Sociais, redes de relacionamentos podem ser especificadas por amizade, dominância, comunicação, etc. (KNOKE e KUKLINSKI, 1982).

A literatura de Análise de Redes tem grande superposição com a Matemática e a Estatística quando tratam da Teoria dos Grafos. O segmento não coincidente diz respeito às aplicações da Teoria dos Grafos e à sua interpretação, no contexto das Ciências Sociais.

5.3. Problemas Clássicos e Contemporâneos com Estrutura de Rede

Inúmeros problemas práticos podem ser representados por meio de redes e endereçados com ferramentas de Teoria dos Grafos ou de *Link Analysis*. A título de exemplo, destacamos dois problemas clássicos, com papel importante no desenvolvimento teórico dos respectivos campos de conhecimento (difusão de inovações e procura de empregos), e dois problemas contemporâneos, de grande impacto social e econômico (acessibilidade de documentos na Internet e geração de conhecimento por cooperação indireta).

5.3.1 Difusão de Inovações

A difusão de inovações (ROGERS, 1995) é um processo de comunicação especial pelo qual idéias, conhecimento e tecnologias percebidas como novas são transmitidos ao longo do tempo entre os membros de um sistema social. Justamente em decorrência da novidade do conteúdo comunicado, o processo de difusão vem associado a um nível de incerteza que é reduzido pelo compartilhamento de informação.

Os canais de comunicação através dos quais mensagens passam de um indivíduo para outro formam uma **rede social** cuja estrutura tem forte impacto na velocidade e na taxa de adoção da inovação; a intensidade de comunicações homofílicas, isto é, entre nós semelhantes (GRANOVETTER, 1973; ROGERS, 1995) ou coesos, *versus* a intensidade de comunicações heterofílicas, isto é, entre nós dessemelhantes ou pouco coesos, também impacta fortemente o ritmo e as conseqüências do processo de adoção.

Exemplos de processos de difusão cujos fatores proeminentes de sucesso ou fracasso envolvem características da rede social subjacente são abundantes no clássico trabalho de ROGERS (1995); destacamos os de adoção de: uso de água fervida por moradores de uma vila no Peru; computadores pela alta administração de empresas em Pittsburgh; mecanização agrícola em uma comunidade holandesa; novos métodos de ensino de matemática em escolas do condado de Allegheny, Pennsylvania; novo antibiótico por uma comunidade de médicos. Todos eles, e os demais no livro de Rogers, são muito estimulantes de idéias para novas pesquisas.

5.3.1.1 Desdobramento Recente: *The Tipping Point*

Os modelos de difusão de inovação têm grande semelhança com modelos de disseminação de epidemias. Tanto assim que um dos exemplos apresentados por ROGERS (1995) no capítulo sobre redes de difusão (de inovação) é o caso do "Paciente Zero e Redes Sexuais no Início da Disseminação de AIDS".

GLADWELL (2000), em seu *best seller* internacional ***The Tipping Point,*** faz o caminho inverso e utiliza o modelo epidêmico para analisar o processo pelo qual idéias, modismos e comportamentos sociais disseminam-se por populações, principalmente nos casos em que o "contágio" atinge grandes proporções.

O objetivo de Gladwell é identificar os fatores que, convergindo, fazem um sistema social desequilibrar-se para além do limiar cujo trespasse garante à epidemia massa crítica suficiente para atingir um crescimento exponencial; e também, inversamente, é enumerar os motivos pelos quais a grande maioria das novas idéias, condutas ou produtos "não pegam", isto é, falham no processo de adoção, não atingem o limiar necessário. Neste esforço, destaca a importância de certos nós especiais da rede social; e do contexto fornecido pela rede aos nós individuais. Para consubstanciar sua análise, apresenta, entre outros, os casos da epidemia de sífilis em Baltimore nos anos 90, do programa infantil Vila Sésamo, de uma empresa de alta tecnologia de Delaware e do fim da onda de crimes no metrô de Nova York. Trata-se de um trabalho extremamente provocante e relevante para profissionais envolvidos com comunicação, marketing, ação social, e todas as pessoas que se interessam por propagar produtos ou idéias.

5.3.2 Busca de Empregos

Um bom exemplo do papel de um tipo particular de nó na rede social, chamado por GLADWELL (2000) de "Conector" e caracterizado por apresentar um grau muito elevado, é relatado no tradicional e amplamente citado (ROGERS 1995; WATTS, 1999; BARABASI, 2002) estudo de GRANOVETTER (1973) sobre a forma como profissionais liberais e técnicos, moradores de Newton, subúrbio de Boston, encontram emprego.

Granovetter realizou algumas centenas de entrevistas, detalhando o histórico profissional dos entrevistados. Constatou que 56% deles encontraram emprego através de um contato pessoal; destes, apenas 17% encontravam freqüentemente o contato; na maioria dos casos, o contato era apenas superficial e esporádico, caracterizando a "força das ligações fracas" (GRANOVETTER, 1973).

A justificativa para este fenômeno está em que há pouca informação nova a ser compartilhada por pessoas cujos vínculos são estreitos: por circularem no mesmo grupo social, o que uma sabe a outra também sabe. Já pessoas mais distantes, que freqüentam grupos basicamente distintos, têm mais informação nova para trocar.

5.3.3 Internet

A Internet já foi considerada a materialização melhor acabada da liberdade de expressão. Qualquer pessoa com acesso a um provedor e recursos econômicos e técnicos modestos pode publicar um documento. Para o bem ou para o mal, uma vez "no ar", uma página de Internet torna-se instantaneamente disponível para o mundo e é difícil de censurar.

Isso não significa, no entanto que ela será lida.

Para ser lida, antes de mais nada, ela deve, evidentemente, poder ser encontrada entre os hoje mais de **um bilhão** de documentos disponíveis na *web* (Lawrence e Giles conforme BARABASI, 2002). E para ser encontrada, ela precisa ser "visível".

Uma boa medida de visibilidade de um documento na Internet é o número de *links* que **apontam** para ele. Se todas as páginas da Internet apontassem para a sua página, em pouco tempo todos os internautas a teriam visitado. No entanto, as páginas têm em média seis *links*, cada um apontando para apenas um documento dentre os um bilhão de documentos existentes. Isso faz com que a probabilidade de um particular documento apontar para a sua página seja próxima de zero (BARABASI, 2002).

Diante dessa problemática, perguntas de pesquisa quanto ao número de documentos e de *links* disponíveis na Internet; diâmetro da rede; distribuição do grau dos nós;

distribuição da coesão e distância temática dos documentos; e discrição da topologia da rede são extremamente relevantes para todos os pesquisadores que se preocupam com a acessibilidade de informação na *web*.

5.3.4 Cooperação Indireta: Clientes de Uma Livraria Virtual

O varejo tem liderado a implementação de Sistemas de Recomendações (com potencial de aplicação em variadas outras áreas de atividade) que ajudam consumidores a encontrar rapidamente itens relevantes dentre o conjunto de produtos e serviços oferecidos. Segundo Jeff Bezos, criador da Amazon.com, o principal motivo pelo qual clientes voltam a uma particular livraria virtual é justamente o grau de **ajuda ativa** que o *site* oferece na localização de títulos de interesse para o cliente (RAMO, 1999). A Figura 3 mostra a interface principal do Sistema de Recomendações da Amazon.com.

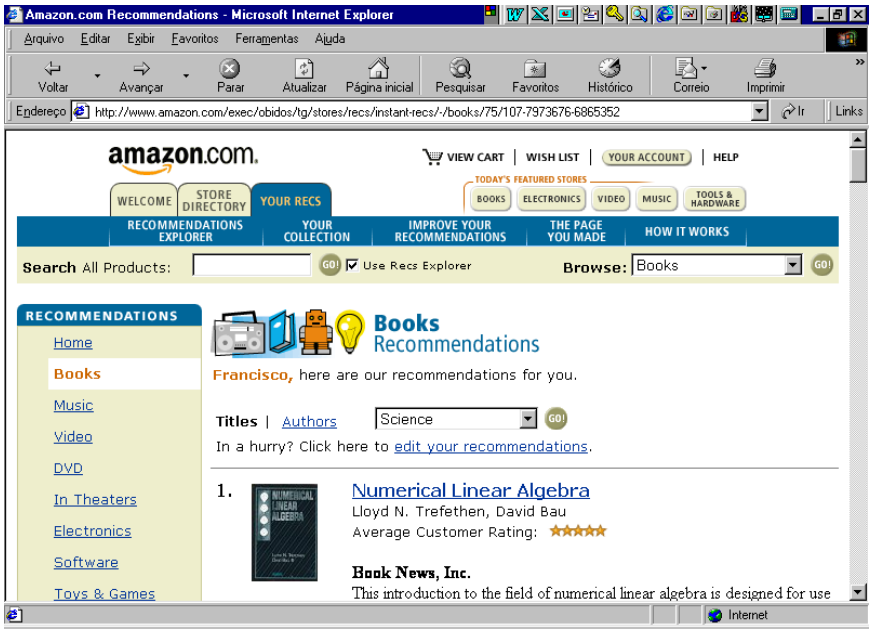


Figura 3 – Interface Principal do Sistema de Recomendações da Amazon.com.

5.3.4.1 Outros exemplos de varejo

Na área de telecomunicações e na indústria cinematográfica, observam-se mais exemplos impulsionados pela Internet e ligados ao consumo:

- a programação de TVs por assinatura é tão extensa em número de canais e de opções de conteúdo que, freqüentemente, o assinante sequer toma conhecimento da disponibilidade de programas de seu interesse; em conseqüência, começam a prosperar serviços como o *Personalized Television* (SMYTH e COTTER, 2000), o qual disponibiliza sugestões personalizadas de programas de TV para 20.000 usuários da Inglaterra e da Irlanda;
- na mesma linha, o sistema *Movie Lens* mantido como ambiente de pesquisa pela Universidade de Minessota, (http://movielens.umn.edu) atualmente oferece sugestões de filmes para cerca de 80.000 usuários, com base em mais de 3 milhões de avaliações individuais.

6. ESCALONAMENTO MULTIDIMENSIONAL (MULTIDIMENSIONAL SCALING – MDS)

Suponha que um conjunto de objetos está sendo analisado e que para cada par de objetos *(r,s)* há uma medida ∂_{rs} de "dissimilaridade" entre eles. Uma definição estreita de Escalonamento Multidimensional é o conjunto de técnicas utilizadas na busca de um espaço de baixa dimensionalidade, usualmente Euclidiano, no qual os objetos são representados por pontos, numa configuração tal que as distâncias entre os pontos no espaço, $\{d_{rs}\}$, correspondam, da melhor maneira possível, às dissimilaridades originais, $\{\partial_{rs}\}$ (COX e COX, 2001).

6.1. Tipos de Dados

Vários tipos de dados prestam-se à análise por MDS; no entanto, tipos diferentes exigem/possibilitam o uso de diferentes técnicas e abordagens.

As principais características relevantes para a especificação da técnica adequada são (COX e COX, 2001): escala de medidas, modos e direções (*ways)*.

6.1.1 Escala de Medida

Dados podem ser classificados quanto à escala de medida utilizada na sua geração. São quatro: nominal, ordinal, intervalar ou racional.

6.1.2 Número de Modos

Cada conjunto de objetos subjacente aos dados é chamado de um **modo**. Assim as dissimilaridades $\{\partial_{rs}\}$ entre livros e usuários da biblioteca apresentados no exemplo da Figura 1 caracterizam-se por serem dados de modo dois. Já as dissimilaridades apenas entre usuários apresentados na base da Figura 2, caracterizam-se por serem de modo um.

6.1.3 Número de Direções

Cada índice numa medida entre objetos é chamada de direção. Portanto, o ∂_{rs} dos exemplos acima apresenta duas direções. Se cada usuário *i* avaliasse a semelhança entre o conteúdo de pares de livros (*r*,*s*), os dados $\partial_{rs,i}$ teriam três dimensões.

6.2. Modelos de Escalonamento Multidimensional

Existem vários modelos de MDS (COX e COX, 2001). O ponto de partida é um conjunto de dados medido em escala intervalar, uni-modal e bi-direcional de medidas de dissimilaridade.

6.2.1 Escalonamento Clássico

Se as distâncias na configuração forem Euclidianas e

$$d_{rs} = \partial_{rs} \qquad r, s = 1,..., n$$

de tal maneira que as dissimilaridades também são precisamente Euclidianas, então é possível encontrar uma configuração de pontos que garanta a igualdade acima.

O escalonamento clássico trata as dissimilaridades $\{\partial_{rs}\}$ diretamente como Euclidianas e faz uso da decomposição espectral de uma matriz de dissimilaridades duplamente centrada.

6.2.2 Escalonamento Métrico por Mínimos Quadrados

Métodos de escalonamento métrico por Mínimos Quadrados fazem corresponder as distâncias $\{d_{rs}\}$ a dissimilaridades transformadas $\{f(\partial_{rs})\}$, onde f é uma função monotônica contínua. Uma configuração é procurada de tal maneira que $\{d_{rs}\}$ se ajuste por Mínimos Quadrados às dissimilaridades transformadas $\{f(\partial_{rs})\}$, por exemplo, minimizando-se a função de perda

$$\frac{\sum_r \sum_s \left(d_{rs} - (\alpha + \beta\partial_{rs})\right)^2}{\sum_r \sum_s {d_{rs}}^2}$$

onde α e β são constantes que devem ser encontradas.

O escalonamento clássico e o escalonamento por MQ são exemplos de escalonamentos métricos, onde "métrico" diz respeito ao tipo de transformação das dissimilaridades e não ao tipo de espaço onde se procura a configuração.

6.2.3 Escalonamento Não-Métrico

Se a natureza métrica da transformação das dissimilaridades é abandonada, chega-se ao escalonamento não-métrico. A transformação f pode agora ser arbitrária, mas deve obedecer à restrição monotônica, que preserva a ordem das dissimilaridades transformadas:

$$\partial_{rs} < \partial_{r's'} \Rightarrow f(\partial_{rs}) \leq f(\partial_{r's'}) \qquad \textit{para} \quad \textit{todo} \quad 1 \leq r, s, r', s' \leq n \,.$$

6.2.4 Análise Procrustes

Suponha que um escalonamento multidimensional tenha sido realizado em dados de dissimilaridade por meio de dois métodos diferentes, dando origem a duas configurações distintas que representam o mesmo conjunto de objetos. A análise Procrustes dilata, translada, espelha e rotaciona uma das configurações para que os pontos se ajustem, da melhor maneira possível, à outra, permitindo a comparação dos resultados.

6.2.5 Outros Modelos

Outros modelos de MDS, como escalonamento unidimensional, *biplots*, *unfolding*, análise de correspondência e modelo de diferenças individuais não são de interesse para este trabalho.

7. ALGUNS DETALHES SOBRE MDS CLÁSSICO E PROCRUSTES

A solução encaminhada neste trabalho utiliza repetidas vezes os algoritmos convencionais de MDS clássico e Procrustes. Apresentamos a seguir alguns detalhes de algoritmos para sua implementação.

7.1. MDS Clássico

Um algoritmo prático para o cálculo do MDS Clássico é sugerido por COX e COX (2001) como:

a) Obtenha as dissimilaridades $\{\partial_{rs}\}$.

b) Calcule a matriz **A** = $\{-\frac{1}{2}\partial_{rs}^2\}$.

c) Calcule a matriz **B** = **HAH** = $\{a_{rs} - a_{r.} - a_{.s} + a_{..}\}$, onde

H é a matriz centralizadora $\mathbf{I} - n^{-1}\mathbf{11}^{\mathbf{T}}$,

I é a matriz identidade,

1 é uma matriz coluna com valor 1 em todas as linhas,

$$a_{rs} = -\tfrac{1}{2}\partial_{rs}^2,$$

$$a_{r.} = n^{-1}\sum_s a_{rs},$$

$$a_{.s} = n^{-1}\sum_r a_{rs} \text{ e}$$

$$a_{..} = n^{-2}\sum_r\sum_s a_{rs}.$$

d) Calcule de **B** os autovalores $\lambda_1, ..., \lambda_{n-1}$ e autovetores associados $\mathbf{v_1}, ..., \mathbf{v_{n-1}}$, com autovetores normalizados de forma que $\mathbf{v_i}^{\mathbf{T}}\,\mathbf{v_i} = \lambda_1$.

e) Escolha um número *p* de dimensões apropriado. (No caso deste trabalho o número de dimensões é 2.)

f) As coordenadas dos *n* pontos nas *p* dimensões do espaço Euclidiano são dadas por $x_{ri} = v_{ir}$ ($r = 1, ..., n$; $i = 1, ..., p$).

7.1.1 Implementação da Função cmdscale no R

No ambiente de processamento do R (veja 13.1), o escalonamento multidimensional clássico está implementado através de uma função disponível na biblioteca *mva* (*multivariate analysis*). O código da função é o seguinte:

```
cmdscale<- function (d, k = 2, eig = FALSE)
{
   if (any(is.na(d)))
      stop("NA values not allowed in d")
   if (is.null(n <- attr(d, "Size"))) {
      x <- as.matrix(d^2)
      if ((n <- nrow(x)) != ncol(x))
         stop("Distances must be result of dist or a square matrix")
   }
   else {
      x <- matrix(0, n, n)
      x[row(x) > col(x)] <- d^2
      x <- x + t(x)
   }
   storage.mode(x) <- "double"
   Tmat <- -0.5 * .C("dblcen", x, as.integer(n), PACKAGE = "mva")[[1]]
   e <- eigen(Tmat, symmetric = TRUE)
   ev <- e$values[1:k]
   points <- e$vectors[, 1:k] %*% diag(sqrt(ev))
   dimnames(points) <- list(dimnames(d)[[1]], NULL)
   if (eig)
      list(points = points, eig = ev)
   else points
}
```

onde

########################## cmdscale ##############################
Parâmetros de entrada:
d = matriz de distâncias ou objeto dist de distâncias

k = número de dimensões desejadas para a solução (default = 2)
eig = valor lógico que indica se a decomposição deve ou não constar da saída da função.

Variáveis:

x	recebe as distâncias ao quadrado;
Tmat	recebe **HAH**, calculada por meio de uma função implementada em C;
e	recebe a decomposição espectral de Tmat;
ev	recebe os autovalores de *e;*
points	recebe as colunas de autovalores relevantes.

Retorna:

points	contendo as coordenadas procuradas
eig	contendo a decomposição espectral, dependendo do parâmetro lógico passado para a função.

7.2. Procrustes

É comum a necessidade de comparar uma configuração de pontos num espaço Euclidiano com outra configuração que mantém com a primeira um mapeamento um-a-um. Se elas não coincidem, assume-se que o espaço em que uma delas está representada sofreu uma deformação. As técnicas de Análise de Procrustes permitem identificar quais foram as deformações e, "desmanchando-as", fazer coincidir as duas configurações. Permitem, também, calcular uma medida do ajuste após as correções.

A solução teórica (COX e COX, 2001) supõe que uma configuração de *n* pontos em um espaço Euclidiano *q*-dimensional, com coordenadas dadas pela matriz $\mathbf{X}_{(n\times q)}$, precisa ser ajustada otimamente a outra configuração **Y** dos mesmos *n* pontos em um espaço Euclidiano *p*-dimensional ($p \geq q$). Assume-se que o *r*-ésimo ponto na primeira configuração pode ser mapeado sobre o *r*-ésimo ponto da segunda.

Inicialmente, *p-q* colunas de zeros são agregadas à matriz **X**, de forma a que ambas as matrizes tenham a mesma dimensionalidade.

A soma das distâncias entre os pontos de **X** e seus correspondentes em **Y** é dada por

$$R^2 = \sum_{r=1}^{n} (\mathbf{y}_r - \mathbf{x}_r)^T (\mathbf{y}_r - \mathbf{x}_r) .$$

Sejam os pontos no espaço **X** dilatados, transladados, rotacionados e refletidos para novas coordenadas $\mathbf{x}'_r$ onde

$$\mathbf{x}'_r = \rho \mathbf{A}^T \mathbf{x}_r + \mathbf{b} .$$

Pode-se demonstrar (COX e COX, 2001) que os movimentos que minimizam R^2 são

$$\rho = tr(\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X})^{\frac{1}{2}} / tr(\mathbf{X}^T \mathbf{X}),$$

$$\mathbf{A} = (\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X})^{\frac{1}{2}} (\mathbf{Y}^T \mathbf{X})^{-1}, \text{ e}$$

$$\mathbf{b} = \mathbf{y}_0 - \rho \mathbf{A}^T \mathbf{x}_0 \quad onde \quad \mathbf{x}_0 = \tfrac{1}{n} \sum_{r=1}^{n} \mathbf{x}_r \quad e \quad \mathbf{y}_0 = \tfrac{1}{n} \sum_{r=1}^{n} \mathbf{y}_0 .$$

Se de interesse, uma medida de qualidade do ajuste, conhecida como estatística de Procrustes, é dada por

$$R^2 = 1 - \left\{ tr(\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X})^{\frac{1}{2}} \right\}^2 / \left\{ tr(\mathbf{X}^T \mathbf{X}) tr(\mathbf{Y}^T \mathbf{Y}) \right\}.$$

7.2.1 Algoritmo de Procrustes

COX e COX (2001) sugerem o seguinte algoritmo para a implementação de uma análise de Procrustes em que uma configuração **Y** deve ser ajustada a uma configuração **X**:

a) Subtraia o respectivo vetor de médias de cada uma das configurações de forma a que os seus centróides coincidam com a origem;

b) Encontre a matriz $\mathbf{A} = (\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})^{\frac{1}{2}}(\mathbf{Y}^T\mathbf{X})^{-1}$;

c) Encontre $\rho = tr(\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})^{\frac{1}{2}} / tr(\mathbf{X}^T\mathbf{X})$; e

d) Calcule $R^2 = 1 - \left\{tr(\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})^{\frac{1}{2}}\right\}^2 / \left\{tr(\mathbf{X}^T\mathbf{X})tr(\mathbf{Y}^T\mathbf{Y})\right\}$.

A implementação desta função encontra-se na subseção 20.1.18, a seguir.

Terceira Parte: Formulação do Problema e Encaminhamentos

Nesta terceira parte do trabalho apresentamos três arquivos de transações que serão utilizados como exemplos; encaminhamos o problema de representação da rede propondo a utilização de uma Base Temática e a substituição das arestas por uma Superfície de Coesão; finalmente, ressaltamos aspectos do tipo de rede comum nesta categoria de problema de cooperação indireta, que simplificam os algoritmos propostos para o encaminhamento da solução computacional desenvolvida na seção final do bloco.

8. TRÊS ARQUIVOS COM DADOS PARA EXEMPLO

Para servir de ilustração na discussão que se segue, uma pequena rede foi criada arbitrariamente e suas arestas registradas no arquivo de exemplo EX1.dat. Em seguida, do arquivo DADOS.dat contendo os dados reais da aplicação foram retirados dois subconjuntos de transações, nomeados EX2.dat e EX3.dat.

Os três arquivos têm, antes do pré-processamento, 19, 1.264 e 4.850 registros, respectivamente. O objetivo ao utilizá-los foi manter os exemplos tão pequenos quanto possível em cada etapa da discussão, pois isto facilita a apresentação de matrizes, cálculos e gráficos ilustrativos dos conceitos estudados; foi também observar como é a tendência de aumento na complexidade do problema do ponto de vista computacional à medida que o número de vértices da rede cresce.

8.1. Exemplo 1 – EX1.dat

O primeiro dos arquivos de exemplo, EX1.dat, tem o conteúdo a seguir.

Saída 1 – Conteúdo do Arquivo EX1.dat

```
109 3
100 1
108 2
100 2
101 1
108 1
101 2
101 1
107 2
107 1
101 5
102 1
106 6
102 3
106 2
104 3
105 3
104 4
105 5
```

Como todos os arquivos de entrada da aplicação, neste trabalho, trata-se, basicamente, de um arquivo contendo duas colunas com *nomes* (identificadores ou "id"s) de vértices.

Cada linha representa uma aresta, caracterizada pelo nó de usuário (primeira coluna) e pelo nó de livro (segunda coluna).

O primeiro registro poderia, portanto, ser lido da seguinte maneira: "O usuário 109 retirou o livro 3." O segundo registro pode ser lido como: "O usuário 100 retirou o livro 1." E assim por diante.

Depois de devidamente pré-processado (a finalidade e os procedimentos do pré-processamento serão discutidos na subseção 9.1), o arquivo do Exemplo 1 perdeu seis linhas e recebeu nome nas linhas e colunas, ficando com o formato abaixo.

Saída 2 – Arquivo EX1.dat Pré-Processado

	usuário	livro
1	100	1
2	100	2
3	101	1
4	101	2
5	101	5
6	102	1
7	102	3
8	105	3
9	105	5
10	107	1
11	107	2
12	108	1
13	108	2

Os nós V = { 1, 2, 3, 5, 100, 101, 102, 105 , 107, 108} implícitos na lista acima, e as arestas E = { e_i }, i = 1, ..., 13, contidas no arquivo formam a rede representada graficamente a seguir. Note que as "transações de empréstimo" do arquivo de entradas formam a "lista de arestas" da rede.

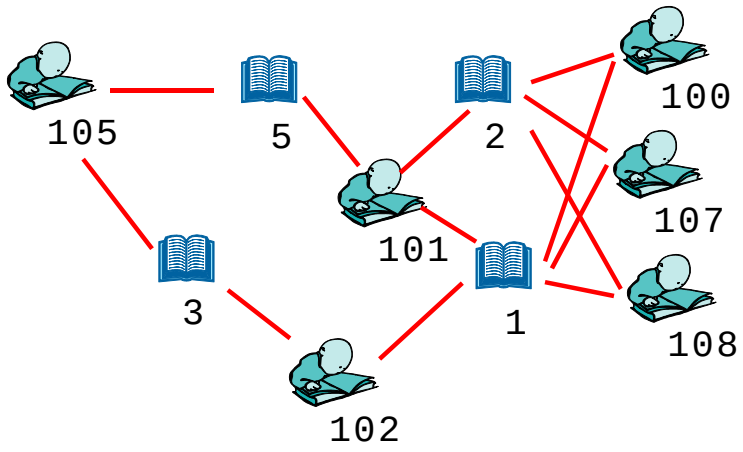


Figura 4 – Representação Gráfica da Rede do Exemplo 1.

A rede pode ser também representada pela matriz de adjacência **A**, da Saída 3, a seguir. Como as redes de nosso interesse não são orientadas, isto é, a aresta (*i*,*j*) é a

mesma aresta que (*j*,*i*), resulta que a matriz de adjacência é simétrica. Assim, por economia, sempre a representaremos apenas na porção subdiagonal.

Saída 3 – Matriz de Adjacência do Exemplo 1.

A

	1	2	3	4	5	6	7	8	9
2	0								
3	0	0							
4	0	0	0						
5	1	1	0	0					
6	1	1	0	1	0				
7	1	0	1	0	0	0			
8	0	0	1	1	0	0	0		
9	1	1	0	0	0	0	0	0	
10	1	1	0	0	0	0	0	0	0

A menos de observação em contrário, as linhas das matrizes de adjacência correspondem aos vértices ordenados crescentemente por id (veja na subseção 13.1.2 comentário sobre as dificuldades trazidas ao algoritmo por conta desta correspondência implícita).

8.2. Resumo Comparativo dos Três Arquivos de Exemplos

Os arquivos de exemplo EX2.dat e EX3.dat são análogos, em termos de estrutura, ao arquivo EX1.dat discutido na subseção anterior. Divergem do primeiro porque são conjuntos de transações reais e porque apresentam números diferentes de transações e nós, nos diversos estágios preliminares de processamento, conforme Tabela 2, a seguir.

Tabela 2 – Resumo Comparativo dos Arquivos de Exemplos

Arquivo	Número Inicial de Registros (TO)	Registos Após Deduplicação (TD)	Registros após Remoção de Cílios (TS)	Vértices de Usuários (TS)	Vértices de Livros (TS)	Linhas na Matriz de Distâncias Temáticas	Linhas Possíveis na Matriz de Distâncias Temáticas
	TO	TD	TS	Usuários	Livros	DT-o	DT-p
EX1.dat	19	18	13	6	4	15	15
EX2.dat	1,264	1,089	175	71	66	147	2,485
EX3.dat	4,850	4,211	2,043	596	702	2,445	177,310

As explicações sobre conceitos e notação implícitos nas colunas de **TO** a **TS** da tabela acima estão na subseção 9.1 a seguir. O número de vértices de usuários

(analogamente, livros) corresponde ao número de usuários (livros) distintos existentes na matriz de entrada deduplicada e sem cílios.

O número de linhas **observado** na matriz de Distâncias Temáticas (DT-o) corresponde ao número de linhas registrado e acompanhado pelo algoritmo, de um total de caselas de Distâncias Temáticas possíveis (DT-p) na matriz de distâncias. As razões que permitem esta "economia" de recursos e suas implicações estão discutidas na subseção 9.3.2.

9. FORMULAÇÃO GERAL DO PROBLEMA

Nesta seção explicamos as principais idéias utilizadas na implementação do algoritmo e damos exemplos dos cálculos realizados em cada etapa.

9.1. Pré-Processamento da Lista de Transações

Como vimos na subseção 8.1, os dados originais para esta aplicação correspondem a transações que ligam Usuários a Livros, isto é, são transações de empréstimo e por este motivo chamamos a matriz correspondente de **TO** (**T**ransações **O**riginais). Num caso geral, os dados de entrada correspondem à Lista de Arestas de uma rede bimodal (no sentido definido em 6.1.2).

Como as transações podem estar duplicadas, isto é, um usuário pode haver retirado o mesmo livro várias vezes, e como esta informação não é de interesse no nosso problema, os registros redundantes são descartados e o resultado é armazenado numa matriz batizada de **TD** (**T**ransações **D**eduplicadas).

Finalmente, as transações em **TD** caracterizam uma rede em cuja periferia tipicamente há muitos vértices tenuemente vinculados à rede, isto é, ligados à rede por apenas uma aresta. Também freqüentemente, estes mesmos vértices "quase soltos" encadeiam-se em longos filamentos, como na Figura 5. Estes "filamentos" não têm posição definida no espaço temático; sua inclusão no problema apenas inflaciona a necessidade de recursos computacionais, sem resultar em entendimento adicional da configuração temática. Por este motivo, uma função foi incluída no algoritmo para remover os "cílios" conservando para análise apenas o "núcleo firme" da rede. O resultado é armazenado na matriz **TS** (**T**ransações **S**em Cílios).

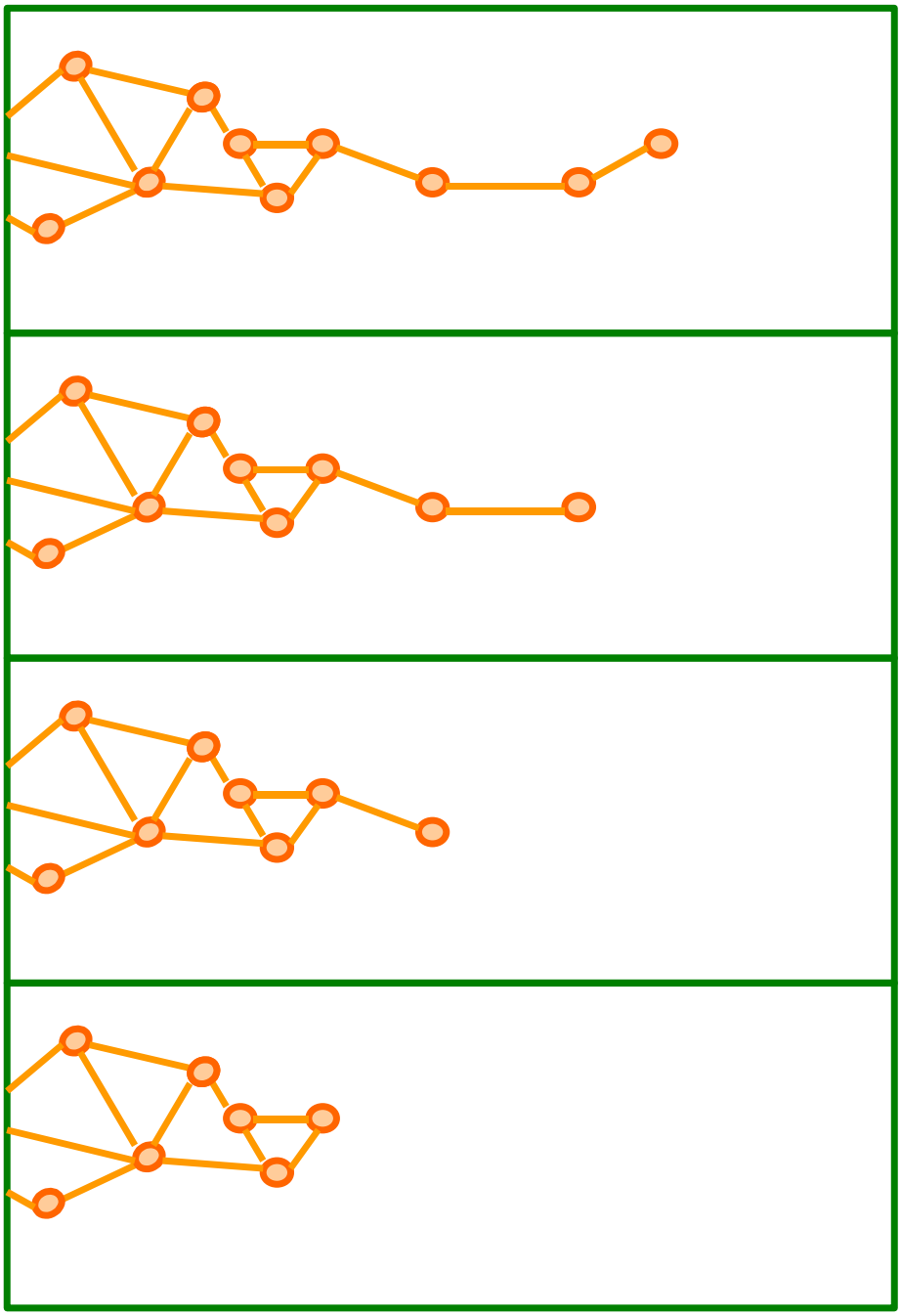


Figura 5 – Exemplo de Remoção de Cílio

O fluxograma da etapa de Pré-Processamento está incluído na Figura 6.

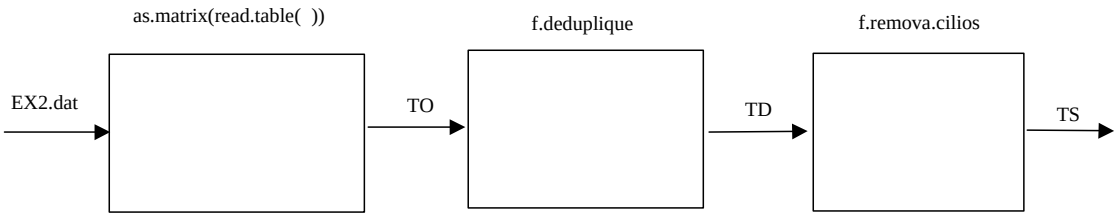


Figura 6 – Fluxograma do Pré-Processamento

9.2. Redes Bimodais e Redes Monomodais

Os dados originais da aplicação sob consideração caracterizam-se por serem bimodais, isto é, terem modo 2. Em outras palavras, subjacentes aos dados, há dois tipos de objetos: usuários e livros. No entanto, neste trabalho, estamos principalmente interessados nas relações dos usuários entre si, ou seja, em vários momentos desejamos representar uma rede monomodal (isto é, de modo 1). Felizmente, podemos passar de uma situação a outra por meio de uma operação muito simples: a partir da matriz de adjacência **A**, construímos a matriz de adjacência em (exatamente) dois passos $\mathbf{A}^2$.

A matriz $\mathbf{A}^2$ mostra as ligações em (exatamente) dois passos que ocorrem na rede original, isto é, mostra se dois usuários são ligados entre si (através de um livro); e mostra também se dois livros estão ligados entre si (através de um usuário). Este último caso, da ligação entre livros, não é de interesse neste trabalho, mas tem muito interesse mercadológico para sugestões de venda cruzada.

A matriz $\mathbf{A}^2$ é, de fato, o quadrado da matriz **A,** com os valores diferentes de 0 substituídos por 1 (o resultado antes da substituição indica de quantas maneiras diferentes podemos passar de um nó a outro através da rede). Dela, no momento, interessa-nos apenas a partição Usuário-Usuário, conforme a Figura 7.

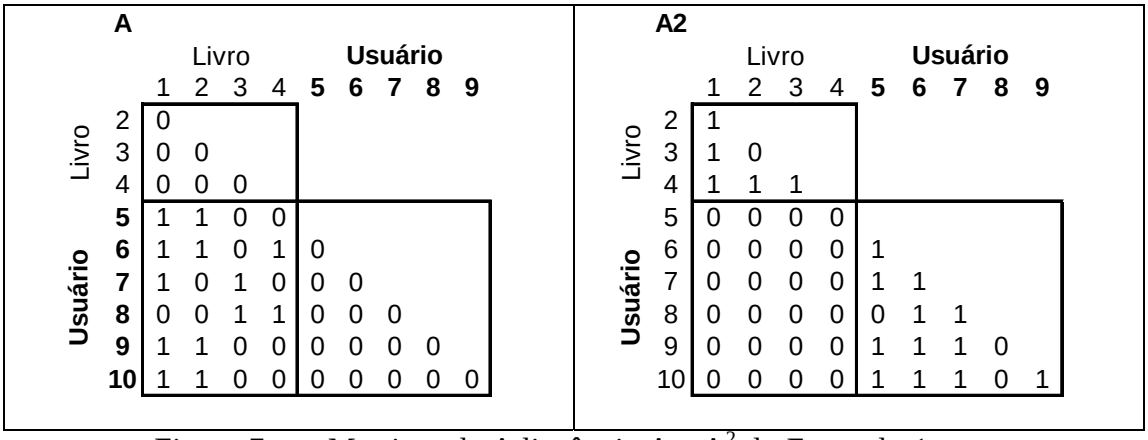


Figura 7 – Matrizes de Adjacência A e A^2 do Exemplo 1, Particionadas segundo o Modo dos Vértices.

Note, na Figura 7, que as partições Usuário-Usuário e Livro-Livro só contém zeros em **A**. Nesta altura, as ligações estão todas na partição Usuário-Livro. Isto, evidentemente, decorre de que apenas usuários podem tomar livros emprestados.

Já em $\mathbf{A^2}$ a situação se inverte; não há ligações entre usuários e livros, em dois passos. Por meio de exatamente duas arestas, livros ligam-se a livros e usuários a usuários.

A representação gráfica de $\mathbf{A^2}$ (na partição Usuário-Usuário) é apresentada na Figura 8, a seguir.

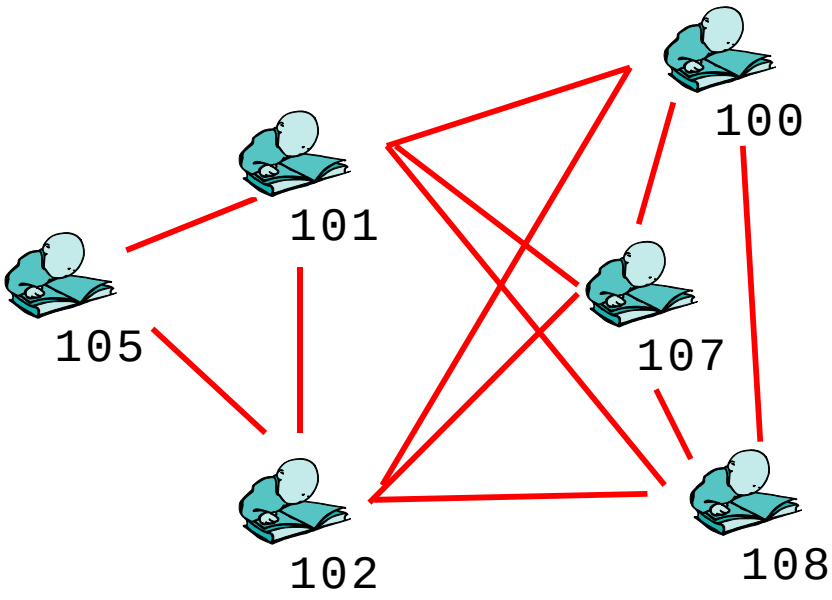


Figura 8 – Representação Gráfica da Rede de Usuários do Exemplo 1.

9.3. Cálculo e Armazenamento das Distâncias Temáticas

De posse das informações de usuários, e para evitar a montagem da matriz de adjacência, podemos organizá-las como uma Lista de Vizinhos de cada vértice.

A Lista de Vizinhos (LV) do Exemplo 1 tem a forma apresentada na Saída 4. Ela é divida em dois blocos: Vizinhos em (exatamente) 1 Passo (LV$V1) e Vizinhos em (exatamente) 2 Passos (LV$V2). Em LV$V1, para cada usuário são listados os livros que ele retirou; por exemplo, o usuário com id = 100 retirou os livros com id = 1 e 2. Em LV$V2, para cada usuário são listados os usuários que retiram os mesmos livros que ele. Assim, aprendemos que os usuários 101, 102 107 e 108 retiraram livros também retirados pelo usuário 100. Confronte a Saída 4 com a Figura 4 e a Figura 8 para confirmar o entendimento da LV.

Saída 4 – Lista de Vizinhos dos Nós de Usuários no Exemplo 1.

```
LV # Lista de Vizinhos
$V1 # Vizinhos de Usuários em 1 passo, portanto, Livros
$V1[[1]] # id = 100
1 2

$V1[[2]] # id = 101
1 2 5

$V1[[3]] # id = 102
1 3

$V1[[4]] # id = 105
3 5

$V1[[5]] #id = 107
 1  2

$V1[[6]] #id = 108
 1  2


$V2 # Vizinhos de Usuários em 2 passos,  portanto, Usuários
$V2[[1]] # id = 100
[1] 101 102 107 108

$V2[[2]] # id = 101
[1] 100 102 107 108 105

$V2[[3]] # id = 102
[1] 100 101 107 108 105

$V2[[4]] # id = 105
[1] 102 101

$V2[[5]] # id = 107
[1] 100 101 102 108

$V2[[6]] # id = 108
[1] 100 101 102 107
```

9.3.1 Cálculo

De posse da Lista de Vizinhos fica fácil calcular a Distância Temática.

Para evitar perda de informação, neste ponto é útil considerar a vizinhança em dois passos (também chamada de vizinhança expandida) de cada vértice. Vejamos, por exemplo, o caso dos vértices 102 e 105.

A vizinhança em exatamente um passo de v_{102} é $\Gamma_{102,1} = \{1, 3\}$; e de v_{105} é $\Gamma_{105,1} = \{3, 5\}$.

A vizinhança em exatamente dois passos de v_{102} é $\Gamma_{102,2} = \{100, 101, 105, 107, 108\}$; e de v_{105} é $\Gamma_{105,2} = \{101, 102\}$.

A vizinhança expandida (isto é, em até dois passos) de v_{102} é $\Gamma_{102} = \{1, 3, 100, 101, 105, 107, 108\}$; e de v_{105} é $\Gamma_{105} = \{3, 5, 101, 102\}$.

A união entre ambas as vizinhanças expandidas é {1, 3, 5, 100, 101, 102, 105, 107, 108}, e a intersecção é {3, 101}. Assim, a distância temática entre estes dois nós é

$1- [\#(\Gamma_{102} \cap \Gamma_{105})/\#(\Gamma_{102} \cup \Gamma_{105})] = 1 - [2/9] = 0.778.$

No caso do Exemplo 1, a lista completa das Distâncias Temáticas é apresentada na Saída 5, a seguir. No exemplo analisado acima, estamos falando do par de vértices (102, 105) que, em termos de índices de linhas e colunas da matriz de distâncias é o par (4,3), cuja DT está assinalada em negrito na matriz da Saída 5.

Saída 5 – Distâncias Temáticas entre Usuários do Exemplo 1

	1	2	3	4	5
2	0.4444444				
3	0.5555556	0.5000000			
4	0.7500000	0.8000000	**0.7777778**		
5	0.2857143	0.4444444	0.5555556	0.7500000	
6	0.2857143	0.4444444	0.5555556	0.7500000	0.2857143

9.3.2 Armazenamento

A distância temática é calculável entre todos os pares de nós de uma rede.

Numa rede com baixa densidade de ligações, no entanto, haverá uma grande quantidade de pares de vértices cujas vizinhanças terão intersecção vazia, o que resulta numa Distância Temática igual a 1 – que é o máximo possível. O algoritmo proposto neste trabalho não registra na Lista de Distâncias temáticas os pares de vértices nessa situação. Isto representa uma enorme economia de recursos de armazenamento.

A Tabela 2 mostra que, conforme o número total de vértices passa de 10 para 137 e 1298, a Lista de Distância Temáticas passa respectivamente de 15 para 481 e 2445 registros, o que corresponde a respectivamente 100%, 18% e 1% das distâncias existentes.

9.4. Cálculo da Coesão

O cálculo da coesão na vizinhança expandida de um vértice é fácil de realizar com base na Lista de Arestas (ou seja, os próprios dados em TS) e da Lista de Vizinhos.

Dado um vértice *i*, digamos o 102, consultamos na LV o tamanho de LV$V1 e de LV$V2. O produto destas duas quantidades corresponde ao total de arestas possíveis naquela vizinhança expandida.

Para o cálculo do número de arestas existentes, basta contar em **TS** (veja a Saída 2, na página 43) as arestas cujos dois extremos (isto é, vértices) pertencem à vizinhança considerada.

No caso do vértice 102, este cálculo resulta em 5/(2*5) = 0.5. A Saída 6 mostra a coesão dos vértices do Exemplo 1. O cálculo relativo ao vértice 102 está assinalado em negrito.

Saída 6 – Coesão dos Vértices do Exemplo 1

```
$Coesao
[1] 0.8750000  0.5333333  0.5000000  0.5000000  0.8750000  0.8750000
```

9.5. Representação da Rede Em Espaço Temático

A representação de uma rede por meio de grafo em um plano é elemento chave de apoio a aplicações em inúmeras áreas do conhecimento (BATTISTA e OUTROS, 1999). Por exemplo:

- **engenharia de software** – gráficos de fluxo de dados, chamadas de sub-rotinas, árvores de encadeamento de programas, hierarquias de objetos;
- **bancos de dados** – relacionamentos entre entidades;
- **sistemas de informação** – diagramas organizacionais;
- **sistemas de apoio à decisão** – redes PERT, árvores de atividade;
- **eletrônica** – desenho de circuitos;
- **inteligência artificial** – diagramas de representação de conhecimento;
- **biologia** – árvores evolutivas;
- **química** – desenho de moléculas;
- **engenharia civil** – desenho de plantas; e
- **cartografia** – desenho de mapas.

Neste trabalho propomos ampliar a lista acima com a inclusão de uma aplicação da representação de redes num problema de cooperação indireta, relevante para as áreas de marketing e de gerenciamento do conhecimento.

A representação geométrica de gráficos vem sendo estudada por matemáticos há séculos (BATTISTA e OUTROS, 1999). Nos anos 60 começou a ser utilizada no auxílio à compreensão de software. O artigo de Knuth (1963, conforme BATTISTA e OUTROS, 1999) sobre o desenho de fluxogramas foi o primeiro a apresentar um algoritmo de desenho de grafo para efeito de visualização. Em conseqüência do

problema ter uma componente combinatorial e geométrica, e devido à diversidade dos campos de aplicação, a pesquisa sobre este assunto tem sido muito diversificada.

O problema de desenhar um grafo de forma a refletir acuradamente sua estrutura é inerentemente mal-definido (SKIENA, 1998). De maneira geral procura-se um algoritmo que mostre a estrutura do gráfico de forma que ela possa ser melhor compreendida e que seja esteticamente agradável. Infelizmente estes critérios demasiado "suaves" não permitem o desenho de um algoritmo otimizante. Segundo eles, é possível fazer representações muito diferentes, cada uma apropriada em um contexto. A Figura 9 traz 3 representações distintas de um grafo de ordem 10 e tamanho 15.

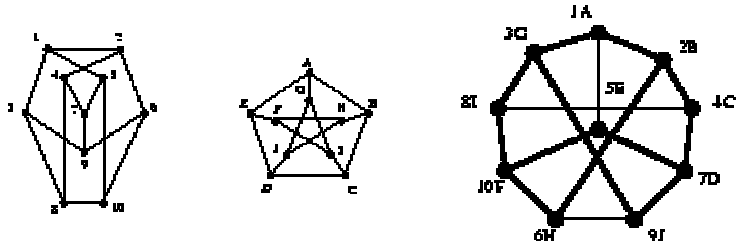


Figura 9 – Diversas Representações de um Grafo de Ordem 10 e Tamanho 15

Por outro lado, critérios mais "duros" e específicos podem ser definidos para medir a "qualidade" de um gráfico:

- **cruzamentos:** busca-se uma representação com um mínimo de cruzamentos de arestas pois elas dificultam a leitura do grafo;
- **área:** busca-se uma representação compacta, mas que garanta uma distância mínima entre os nós;
- **comprimento das arestas:** procura-se evitar arestas longas, pois elas dificultam a leitura do grafo;
- **proporções:** procura-se respeitar as proporções da mídia utilizada para a saída (papel, tela, etc).

Estes objetivos são mutuamente contraditórios; e a busca de uma solução ótima quase sempre resulta num problema computacionalmente intratável.

9.5.1 Arestas como Distâncias Temáticas

Para a representação dos grafos de redes, neste trabalho propomos que as arestas sejam segmentos de reta (isto é, não sejam curvas), e que tenham comprimento proporcional à Distância Temática entre os nós. Os vértices, portanto, estarão em posição coincidentes (distância zero) se tiverem a mesma vizinhança; e estarão a uma distância de 1, se não tiverem vizinhos comuns.

Esta propriedade implica que o espaço (plano) onde os vértices são representados passa a ser um "espaço temático" no sentido de que usuários com interesses comuns serão desenhados próximos.

Implica também que a posição relativa dos vértices pode ser encontrada por Escalonamento Multidimensional a partir da Matriz de Distâncias Temáticas calculada para os nós da rede.

A Figura 10 e a Figura 11 representam a rede do Exemplo 1, conforme esta estratégia, respectivamente sem e com legenda de pontos. Note que os vértices 100, 107 e 108 ocupam a mesma posição.

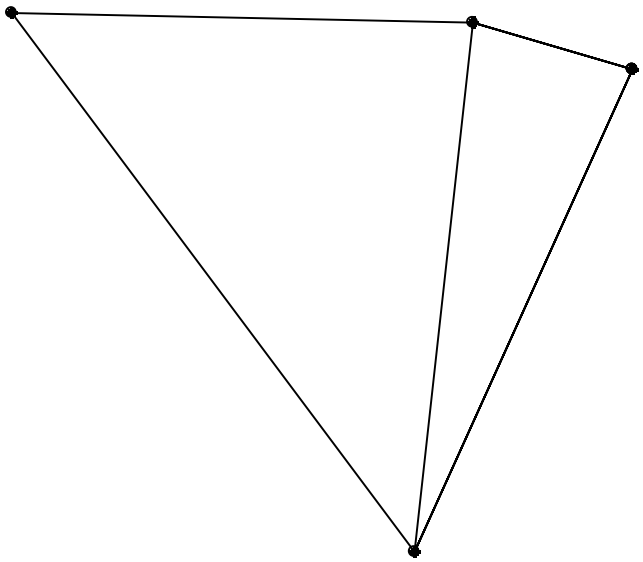


Figura 10 – Rede do Exemplo 1 em Espaço Temático (sem Legenda de Pontos)

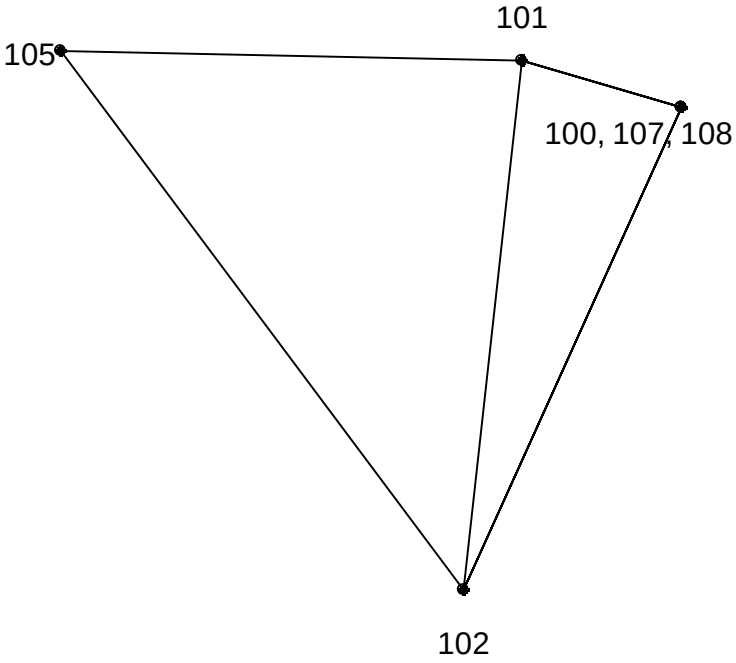


Figura 11 – Rede do Exemplo 1 em Espaço Temático (com Legenda de Pontos)

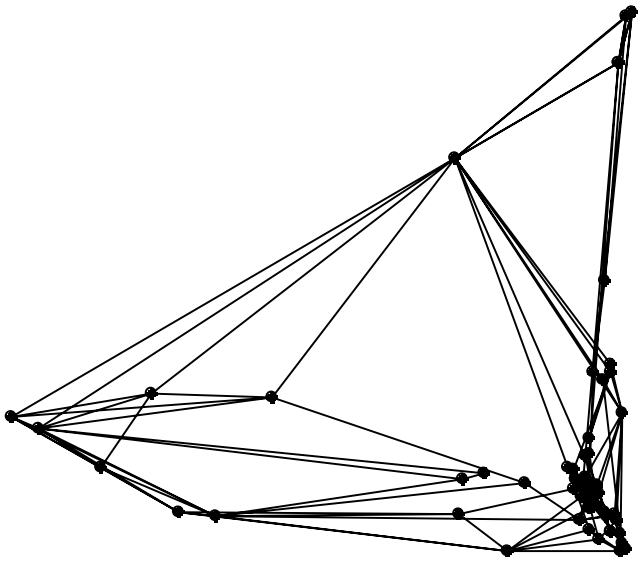


Figura 12 – Rede do Exemplo 2 em Espaço Temático (sem Legenda de Pontos)

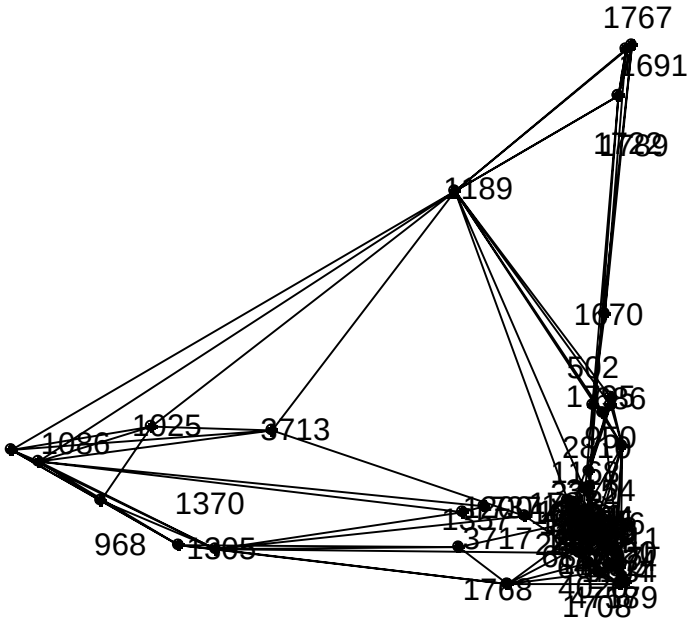


Figura 13 – Rede do Exemplo 2 em Espaço Temático (com Legenda de Pontos)

9.6. Substituição de Arestas por Superfície de Coesão

Um dos problemas na representação gráfica de redes diz respeito à resolução da imagem produzida.

Conforme seu número aumenta, vértices e arestas começam a se superpor, dificultando a leitura. Um gráfico completo de 100 vértices, por exemplo, contém aproximadamente 5.000 arestas; num monitor de 1.000 por 1.000 pixels, cerca de 200 pixels poderiam ser alocados por aresta; o desenho tornar-se-ia um borrão.

Este efeito já pode ser notado ao passarmos do Exemplo 1 (6 vértices) para o Exemplo 2 (73 vértices), como vimos na Figura 10 e na Figura 12. Fica ainda mais agravado quando se deseja identificar os nós, como se nota na Figura 11 e na Figura 13.

Neste trabalho, propomos a estratégia de abandonarmos a representação de arestas individuais em redes grandes, substituindo-as por uma **Superfície de Coesão**.

Esta superfície é obtida fazendo-se associar às coordenadas de cada vértice (definidas conforme a subseção anterior) uma altura proporcional à coesão no vértice. Os demais pontos são obtidos por interpolação ou pelo ajuste de uma superfície suave aos pontos definidos sobre os nós.

A estratégia faz sentido porque, de um lado, os pontos próximos compartilham a mesma vizinhança; e de outro, porque, se a coesão for alta, os vértices na vizinhança serão muito interligados entre si; se for baixa, eles serão pouco interligados.

Troca-se, assim, precisão na informação por representabilidade.

Num sentido "frouxo", o procedimento corresponde a tratar a rede, uma entidade tipicamente discreta, como uma entidade contínua, isto é, como se em cada ponto da superfície temática existisse (potencialmente) um nó.

A Figura 14 e a Figura 15, a seguir, apresentam a superfície de coesão para os Exemplos 1 e 2.

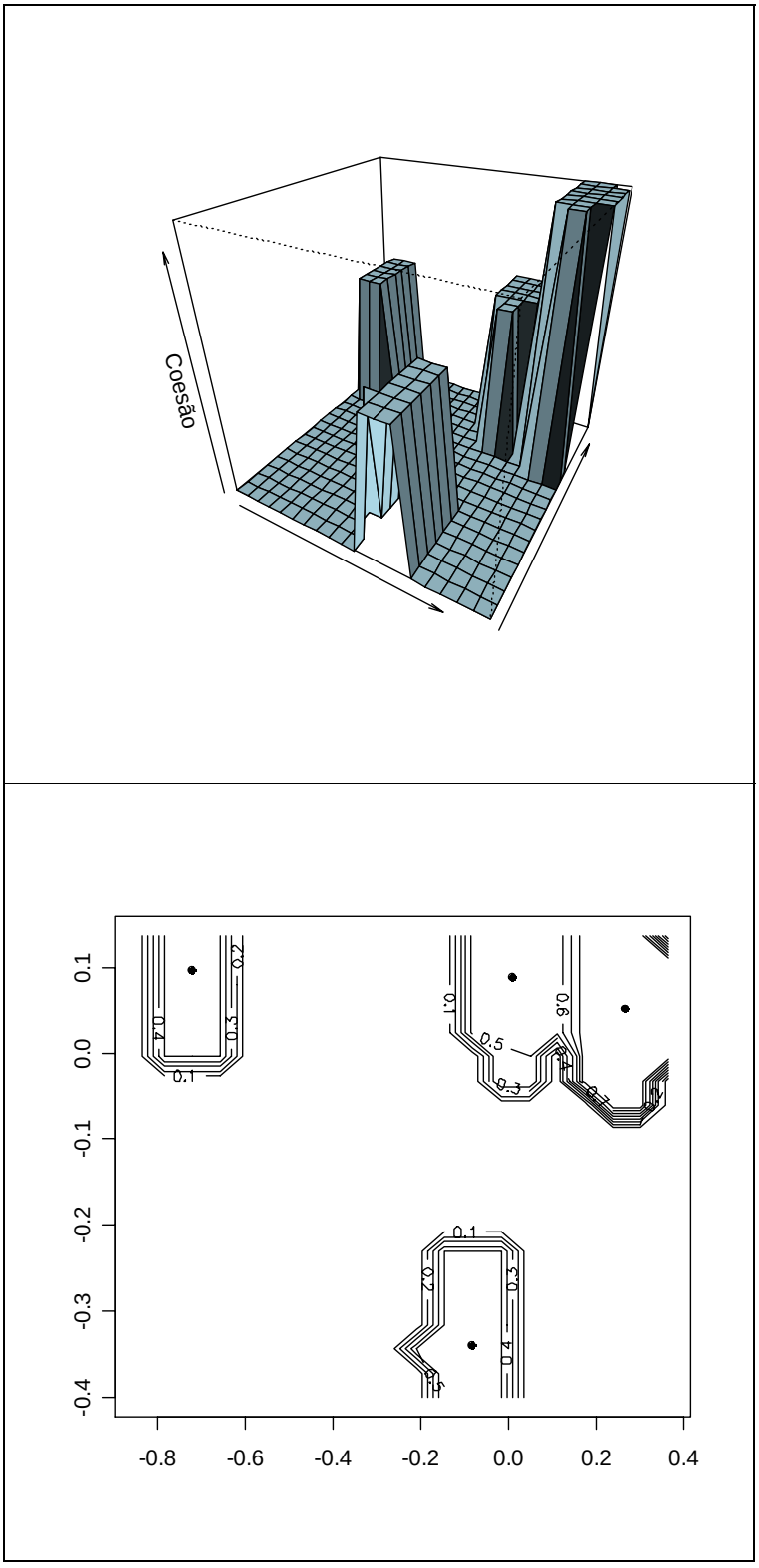


Figura 14 – Superfície de Coesão do Exemplo 1.

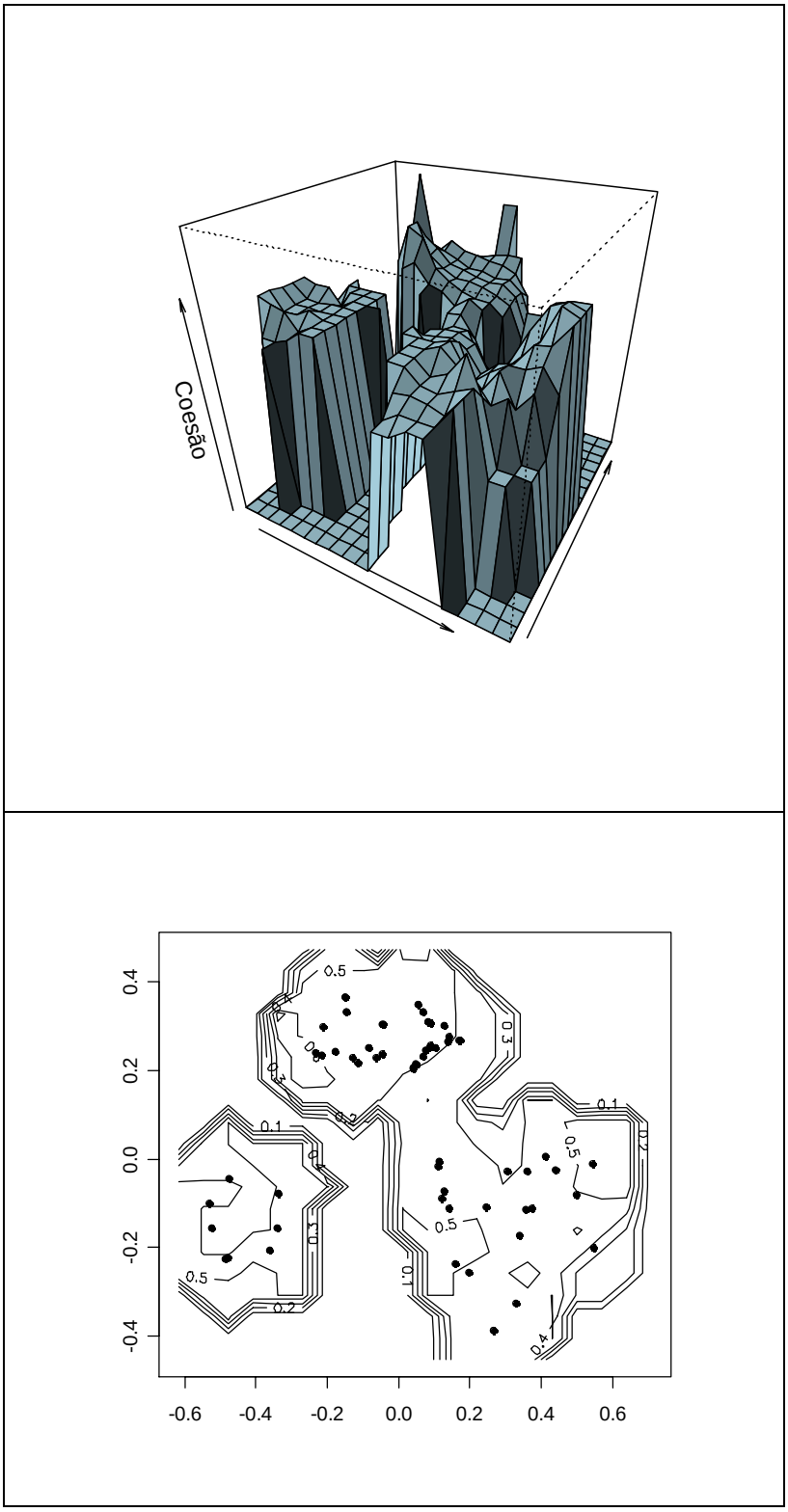


Figura 15 – Superfície de Coesão do Exemplo 2.

10. BASE ESCALONADA

O problema que surge na montagem da base temática (escalonada) para a representação da rede é que os algoritmos tradicionais de escalonamento multidimensional consomem recursos na ordem O(n^2). No computador com processador Pentium-S, 32 MB de ram e 2 Gyga de disco, em que foi processado este trabalho, já não é possível escalonar diretamente os cerca de 600 nós da rede, por esgotamento da memória: não é possível alocar a matriz de distância para submetê-la à função *cmdscale*.

Uma estratégia para enfrentar o problema envolve a utilização de computadores com capacidade de processamento cada vez maiores. Seria a solução pela força bruta. Um projeto recentemente divulgado (http://www.unt.edu/ir/bigmds/, *link* válido em 30/01/2003) pretendia adaptar algoritmos de MDS para permitir processamento paralelo a ser realizado por rede de computadores das bibliotecas públicas nos EUA; o resultado do processamento seria projetado na cúpula de um planetário, numa área de 15m por 15m, ou seja, numa tela de 225 m^2.

Em nossa abordagem, ao contrário, adotou-se o princípio da economia de recursos computacionais e se optou pela estratégia de "dividir para conquistar" (AHO, HOPCROFT e ULLMAN, 1974), isto é, de dividir o problema em partes menores, encontrar solução para as partes, e então combinar a solução das partes para obter uma solução para o todo.

As principais etapas desenhadas para o escalonamento por partes são:

- subdividir a Lista de Arestas em pequenos conjuntos relativos a pontos (implicitamente) próximos;
- "trocar" cada grupo de pontos por um "representante", formando um conjunto de representantes suficientemente pouco numeroso para ser escalonado; e
- escalonar, na seqüência, os pontos inicialmente substituídos, usando como referência para a junção das soluções parciais os respectivos pontos "representantes", a esta altura já com coordenadas fixadas.

Vejamos nas subseções a seguir, em maior detalhe, o que isso significa.

10.1. Quebrando o Problema em Partes Hierarquizadas (Agrupando de Baixo para Cima)

Com base no princípio de que o agrupamento de vizinhos coesos tem "relevância local" (veja 12.4), foram adotados os seguintes critérios para agrupar os vértices nos subconjuntos de processamento:

- A Lista de Vizinhos foi ordenada por Coesão e Grau, criando-se um índice que permite recuperar a posição original da vizinhança;
- Em cada passo, é considerado o vértice cuja vizinhança é a mais coesa; este vértice e os participantes de sua vizinhança ainda não alocados são atribuídos a um grupo; os membros do grupo são escalonados; cria-se um centróide do grupo ("pai"), que o representará no próximo nível de processamento (veja Figura 16, a seguir);
- Os vértices já alocados não são mais considerados;
- De forma iterativa são tratadas as vizinhanças progressivamente menos coesas;
- Vértices que "sobram" sozinhos em uma vizinhança que resulta com apenas dois elementos disponíveis são alocados no grupo de um vizinho deles (grupos de menos de três elementos não são permitidos pois a solução para seus "filhos" não pode ser ajustada à solução global);
- Vértices soltos, isto é, "donos" de vizinhanças em que todos os vizinhos foram alocados são deixados para o final e incluído em um grupo de "sobras".

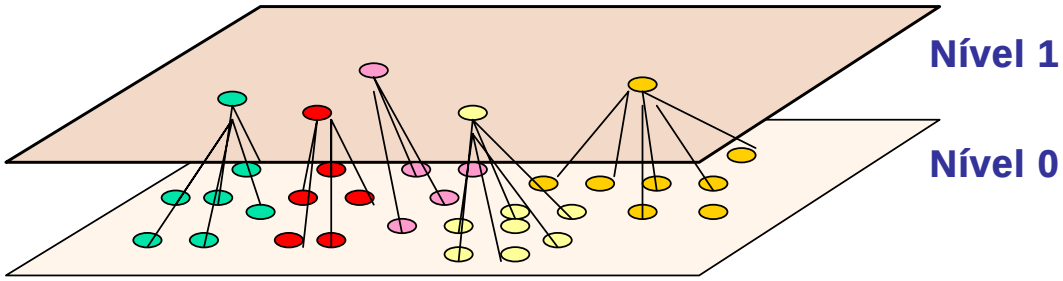


Figura 16 – Alocação de Vértices a Grupos e Criação de Representantes.

Para controle do processo, os vértices originais são considerados como pertencentes a um “Nível 0”; já os centróides criados no processo de agrupamento de nós do Nível 0 são considerados pertinentes ao Nível 1. Na iteração 0, os novos vértices são “pais”, em relação aos nós do Nível 0; na iteração 1, os centróides criados no Nível 1 passam a ser “filhos” e ganham “pais” no Nível 2, e assim por diante, conforme a Figura 17.

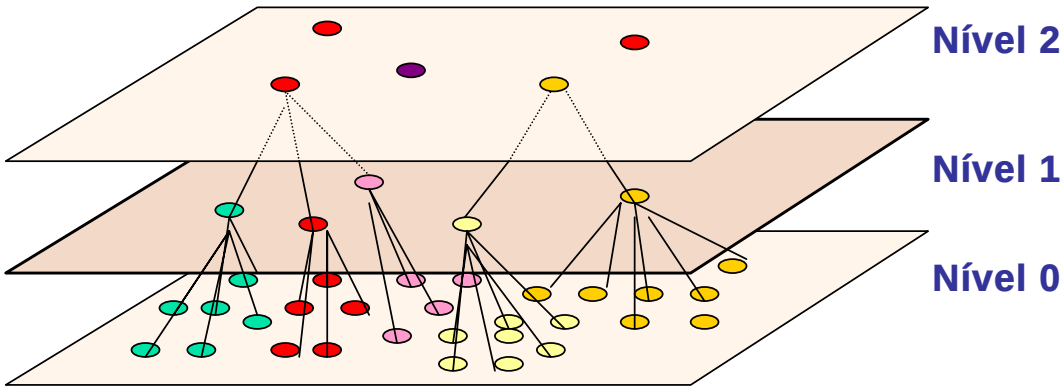


Figura 17 – Redução do Número de Vértices em Iterações Sucessivas.

O número de iterações “de subida” é controlada pelo código no corpo do programa (veja subseção 20.2), e foi ajustado manualmente em cada processamento. Caso haja interesse em automatizar uma regra de parada, seria simples, por exemplo, interromper as iterações quando o número de “representantes” obtido puder ser escalonado.

A Figura 18 mostra como os vértices da rede do Exemplo 2 foram alocados em 16 grupos. Os pontos vazados são os vértices originais e os pontos sólidos são os centróides criados para representar o grupo no Nível 1. A posição de todos os pontos é a que resultou ao final da aplicação do algoritmo proposto neste trabalho.

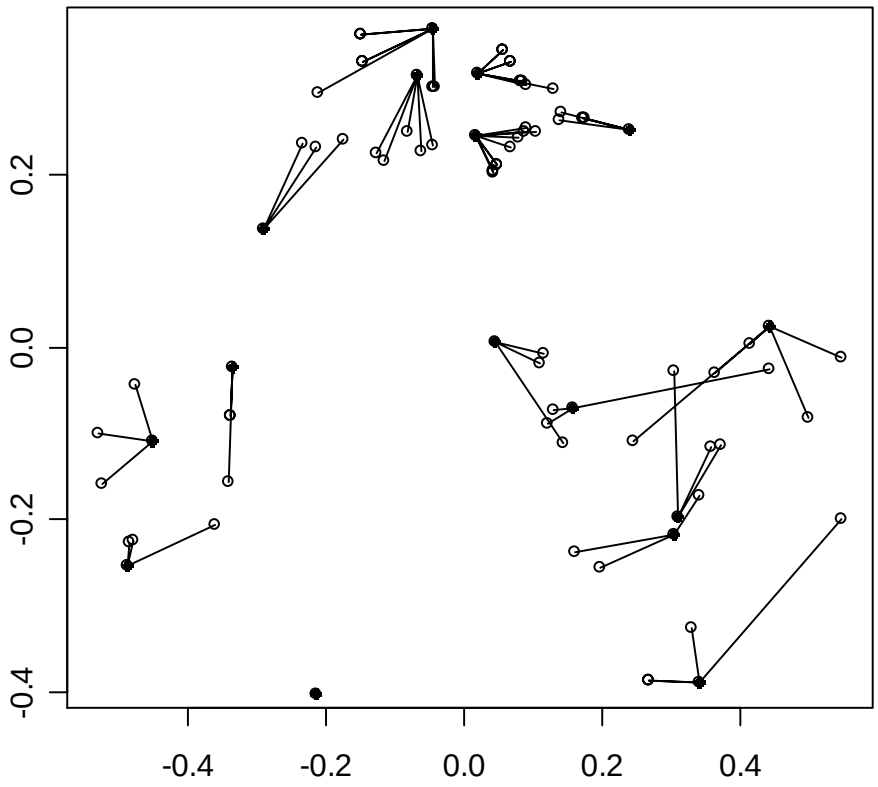


Figura 18 – Arquivo de Exemplo 2: Vértices da Rede (Nível 0, em Círculo Vazado) Ligados ao Centróide de Seu Grupo de Processamento (Nível 1, em Círculo Sólido).

10.2. Coleta das Listas de Arestas

As arestas entre os vértices necessárias à implementação do algoritmo são coletadas em diversos arquivos.

10.2.1 dp0p0

Com base na informação inicial de transações, disponível em **TS**, é calculada a Distância Temática entre os nós de usuários (os vértices de livros não são de interesse imediato neste trabalho, embora possam ser analisados e interpretados de maneira mercadologicamente útil).

Esta é a etapa computacionalmente "mais cara" de todo o algoritmo. Em uma solução completa, para n vértices, n^2 distâncias deveriam ser calculadas. No entanto, duas estratégias de economia foram adotas:

- com base no princípio da relevância local (subseção 12.4), foram calculadas distâncias temáticas apenas entre usuários vizinhos imediatos entre si (isto é, vizinhos em exatamente 2 passos na rede original);
- e, além disso, só foram armazenadas distâncias inferiores a 1.

As distâncias obtidas entre os vértices originais são equiparadas a arestas e armazenadas na matriz *dp0p0* ("distâncias de parentes 0 com parentes 0", isto é, distância entre vértices do Nível 0).

10.2.2 dp1p0

Uma vez formados os grupos do Nível 0 (no caso do Exemplo 2, em número de 16), e calculados seus centróides, a distância entre o centróide ("pai") e os vértices do grupo ("filhos") são calculadas com base em posições obtidas por escalonamento multidimensional clássico do grupo, e coletadas na matriz *dp1p0* ("distância de parentes de nível 1 e parentes de nível 0").

Estas arestas vinculam os filhos a seu pai, e são importantes na junção das soluções parciais por grupo.

10.2.3 dp1p1

A distância entre os "pais" v_i e v_j, respectivamente centróides do grupo i e do grupo j, é calculada como a menor distância entre um "filho" de v_i e um filho de v_j. São duas as razões desta escolha:

- a primeira diz respeito ao conteúdo de informação de cada medida; no contexto da distribuição das distâncias temáticas que ocorrem na aplicação em foco, as distâncias menores são as mais informativas (veja subseção 12.3);
- a segunda razão é de natureza computacional: é "barato" computacionalmente consolidar as distâncias temáticas originais para obter esta medida entre centróides.

As distâncias entre centróides criados no Nível 1 são armazenadas na matriz e no arquivo *dp1p1*.

10.2.4 Níveis Mais Agregados

O processo de agrupamento de vértices, criação de representantes de nível mais elevado, cálculo e coleta de distâncias entre pais e filhos, e entre parentes de mesmo nível continua iterativamente até que os centróides resultantes sejam suficientemente pouco numerosos para serem escalonados por MDS clássico. Em cada iteração vão sendo formados sucessivamente as matrizes e os arquivos *dp1p2*; *dp2p2*; *dp2p3*; *dp3p3*; etc.

10.3. Escalonando (de Cima para Baixo)

Tendo atingido um nível de agregação suficiente para viabilizar o MDS clássico, inicia-se a solução do problema propriamente dita.

10.3.1 Configuração do Último Nível (Nível Mais Elevado)

Quando o conjunto de centróides de um nível é suficientemente pouco numeroso, pode-se fixar coordenadas por meio do escalonamento multidimensional das distâncias entre eles.

Vejamos um exemplo. No caso do conjunto de dados Ex2.dat, isto já é possível após uma única iteração[1].

[1] Na verdade, o conjunto original de 71 vértices poderia ter sido escalonado diretamente, sem a utilização do algoritmo proposto neste trabalho; esta possibilidade será utilizada no seção 14 para a comparação dos dois métodos.

Com a aplicação do algoritmo, os 71 vértices originais foram reunidos em 16 grupos, representados por seus centróides. As distâncias entre os centróides foram coletadas na matriz *dp1p1*. Por sua vez, esta matriz, no formato de Lista de Arestas, foi transformada numa matriz quadrada de distâncias, assumindo-se que as distâncias faltantes têm valor 1. Em seguida, por MDS clássico fixou-se a configuração de centróides de Nível 1 e suas coordenadas foram armazenadas na matriz **X1**, cuja representação gráfica está na Figura 19.

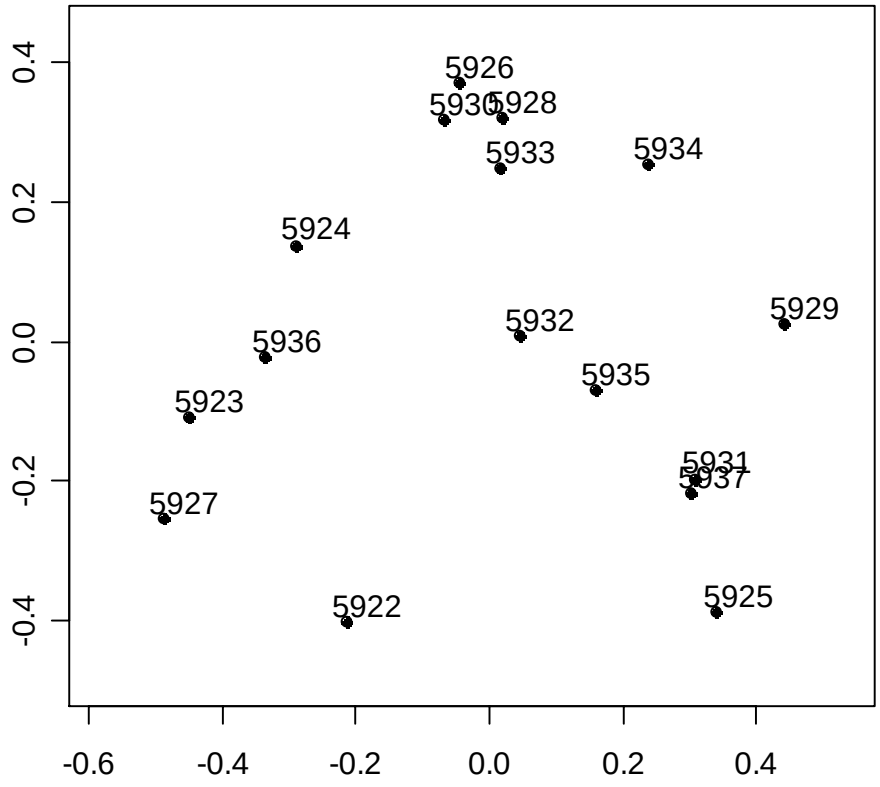


Figura 19 – Centróides de Nível 1 do Exemplo 2, com Rótulos de Vértices.

10.3.2 Triangulando e Encontrando Soluções Parciais

Para a busca de soluções parciais, adotou-se a estratégia de **processar em grupos de três** os vértices da configuração encontrada para um nível de iteração mais elevado já resolvido (no caso do Exemplo 2, estamos nos referindo à configuração representada na Figura 19, e obtida após apenas 1 iteração), juntamente com pontos do nível de baixo, ainda sem coordenadas.

Os motivos para isso são:

- Levando-se em consideração as distâncias entre 3 pais; entre estes pais e seus filhos; e entre todos os filhos destes 3 pais, é possível fixar, por escalonamento, a posição relativa de pais e filhos;
- O conjunto de distâncias obtidos desta maneira é suficientemente pequeno para ser tratado por MDS clássico (o algoritmo garante isso implicitamente porque as vizinhanças são pequenas; testes explícitos e rotinas de tratamento de eventuais exceções devem ser incorporadas ao programa em versões futuras);
- Como as posições absolutas dos pais já estava determinada, a posição absoluta dos filhos pode ser também determinada.

Cada grupo de 3 vértices é formado entre os vizinhos próximos, pois são estes que potencialmente têm influência mútua acentuada.

10.3.2.1 Algoritmos de Triangulação

O particionamento do interior de um conjunto de pontos em triângulos é um problema fundamental em geometria computacional, pois é o primeiro passo para a manipulação de objetos geométricos complicados. No plano, a triangulação é obtida pela inserção entre os vértices de arestas sem intersecção. A solução deve levar em consideração se o formato dos triângulos importa, a escala em que se vai trabalhar, restrições quanto ao traçado das arestas (por exemplo limitadas ao interior da figura), possibilidade de inclusão de pontos novos, etc. Os algoritmos mais simples "custam" $O(n^2)$; algoritmos práticos andam em $O(n \log n)$; e algoritmos de interesse apenas teórico consomem tempo linear (SKIENA, 1998).

O algoritmo “Triangle”, de Jonathan Shewchuk da Universidade de Carnegie-Mellon, é um código em C que gera triangulações de Delaunay, triangulações de Delaunay restritas, e triangulações de Delaunay restritas aderentes a padrões de qualidade (por exemplo, de tamanho de ângulo). É rápido e robusto. O código está disponível em www.cs.cmu.edu/~quake/triagle.html. Foi nosso escolhido entre outras alternativas identificadas: GEOMPCK, de Barry Joe da Universidade de Alberta (ftp://ftp.cs.ualberta.ca/pub/geompack); Voronoi-2D de Steve Fortune (http://netlib.bell-labs.com/netlib/voronoi/ index.html).

A implementação do código Triangle no ambiente de R, no entanto, apresentou muitos problemas que fogem ao escopo deste trabalho, e por este motivo, foi abandonada; ao final, optamos por implementar o nosso próprio algoritmo simplificado de triangulação, que, inclusive, leva em conta a inconveniência para este trabalho do compartilhamento de arestas entre os triângulos formados, típica das soluções de triangulação convencionais. Limites (veja subseção 18.1) e oportunidades de aperfeiçoamento (veja subseção 19.1.1) resultaram das nossas escolhas de implementação.

10.3.2.2 Triangulação no Exemplo 2

Nos dados do Exemplo 2, o resultado da triangulação do vértices do Nível 1 está representado na Figura 20.

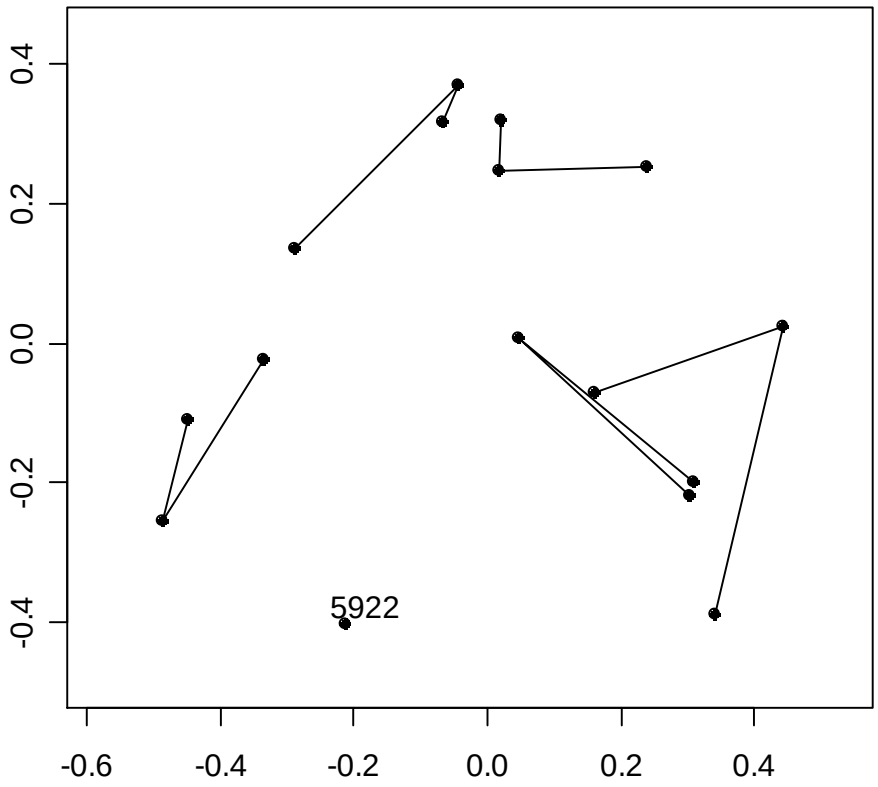


Figura 20 – Triangulação dos Vértices do Nível 1 do Exemplo 2.

Inúmeras exceções relativas ao número de vértices disponíveis para triangulação no final do procedimento tiveram que ser tratadas caso a caso no código. Por exemplo, no caso do arquivo EX2.dat, a última "triangulação" dos centróides do Nível 1 precisa reunir 4 pontos, para evitar que um deles fique isolado. A lógica geral do programa não permite o ajuste da solução parcial para grupos de distâncias com menos de 3 pais.

10.3.2.3 Fixando as Coordenadas de Pais e Filhos em Um Triângulo

Para cada conjunto de três centróides formado, as distâncias entre os próprios vértices do "triângulo", as distâncias entre os vértices do triângulo e seus respectivos filhos e as distâncias entre todos dos filhos destes pais são reunidas numa matriz quadrada de distâncias e escalonadas por MDS clássico.

No caso do Exemplo 2, Nível 1, e considerando-se o conjunto de vértices 5924, 5926 e 5930, as distâncias reunidas são as seguintes (Tabela 3):

Tabela 3 – Distâncias do Primeiro "Triângulo"
Processado no Nível 1 do Exemplo 2.

Origem	Vértice i	Vértice j	Distância
dp1p1	5930	5924	0.28481
	5930	5926	0.05884
	5926	5924	0.33807
dp1p0	5924	684	0.515201
	5924	2723	0.515201
	5924	4680	0.606040
	5926	494	0.533259
	5926	502	0.542315
	5926	686	0.542315
	5926	1078	0.354635
	5926	1347	0.354635
	5926	1724	0.533259
	5926	4540	0.354635
	5930	108	0.294828
	5930	950	0.432863
	5930	1168	0.601237
	5930	2810	0.432863
	5930	3815	0.549015
dp0p0	2810	502	0.76471
	2810	686	0.80000
	2810	950	0.76471
	2723	684	0.83333
	2723	1347	0.40000
	1724	494	0.54545
	1347	684	0.83333
	1168	502	0.77778
	1168	686	0.72727
	950	502	0.60000
	950	686	0.69231
	686	108	0.92308
	686	502	0.45455

Escalonados os dados acima, obtemos a configuração representada na Tabela 4, e na Figura 21.

Tabela 4 – Configuração Processada no Triângulo 1 do Exemplo 2.

Vértice	X1	X2
108	-0.15069	0.22345
494	0.17629	-0.37207
502	-0.36301	-0.23422
684	0.29347	0.27525
686	-0.35103	-0.22427
950	-0.43230	0.06576
1078	0.12198	-0.23116
1168	-0.25447	0.03832
1347	0.36211	-0.00631
1724	0.17629	-0.37207
2723	0.38844	0.27326
2810	-0.36803	0.11475
3815	-0.08595	0.19501
4540	0.12198	-0.23116
4680	0.14252	0.19017
5924	0.31552	0.36430
5926	0.18764	-0.40862
5930	-0.28075	0.33961

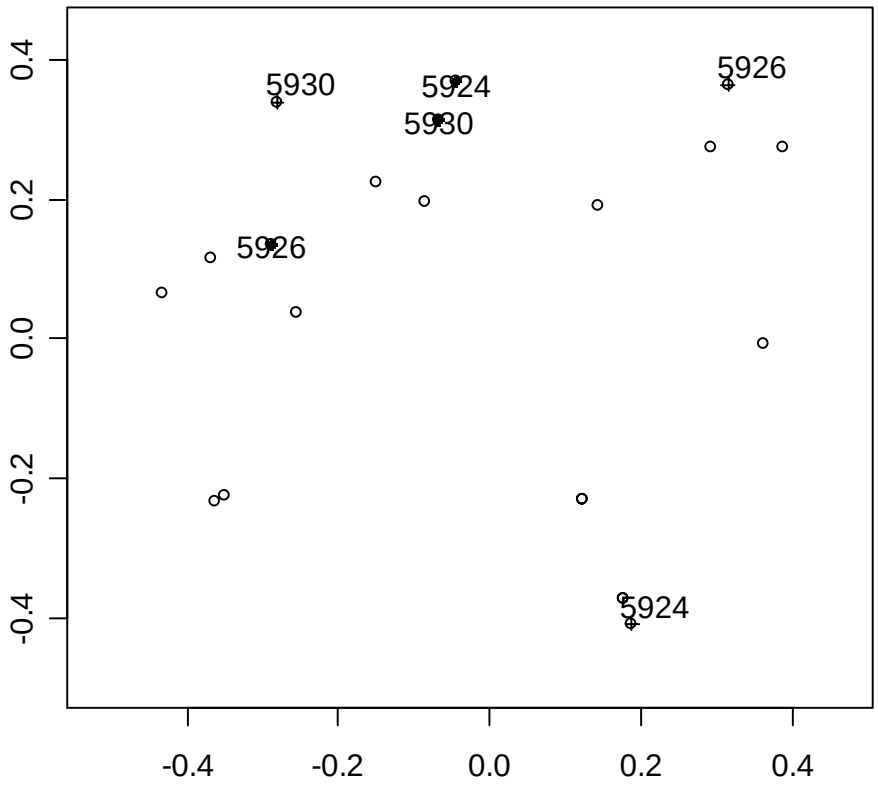


Figura 21 – Configuração no Triângulo 1 do Exemplo 2
(Pontos Vazados) e Coordenadas Originais dos
Nós 5924, 5926 e 5930 (Pontos Sólidos).

10.3.3 Juntando as Soluções Parciais

Na Figura 21, além dos pontos vazados representando os pais (vazados com cruz) e filhos (vazados simples) processados na iteração, também foram incluídos pontos sólidos representando os vértices (pais) fixados no nível anterior. Observa-se que estas posições não coincidem.

Assume-se, então que o espaço em que a solução parcial foi encontrada é uma deformação do espaço fixado no nível de cima. Uma análise de procrustes apenas dos pontos duplicados (isto é, dos pais) permite encontrar a transformação que dá um ajuste ótimo entre as duas configurações, no plano.

No Exemplo 2, a matriz de rotação **A** obtida é

```
           [,1]       [,2]
[1,] -0.9652716  0.2611063
[2,] -0.2611383 -0.9653003,
```

o fator de dilação ρ é

```
[1] 0.2233054,
```

e a translação rígida **b** é

```
           [,1]       [,2]
[1,] -0.1331414  0.2731969.
```

A qualidade do ajuste neste triângulo deixa um pouco a desejar: o coeficiente de procrustes é

```
[1] 0.5614142.
```

As configurações antes e depois do ajuste estão representadas na Figura 22 e na Figura 23, a seguir.

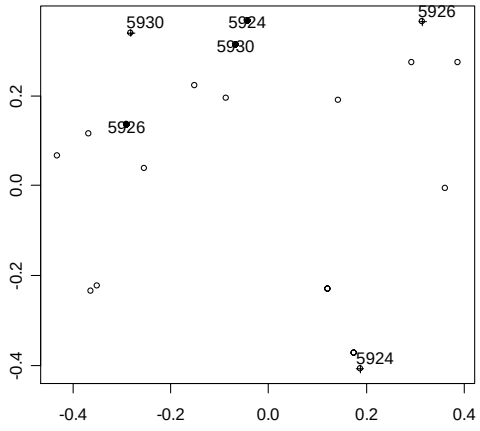


Figura 22 – Configuração Antes do Ajuste

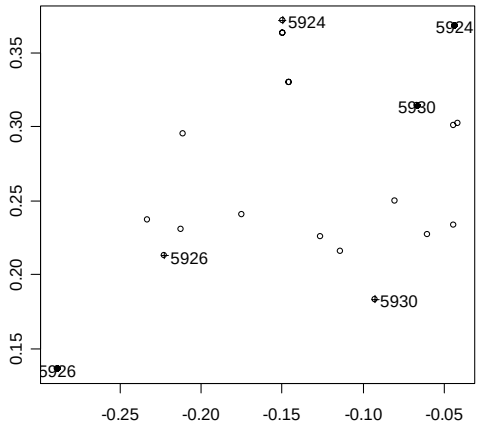


Figura 23 – Configuração Depois do Ajuste.

Após o ajuste, assume-se que as soluções parciais podem ser todas reunidas num único espaço daquele nível.

Aperfeiçoamentos para esta etapa de ajustes são sugeridos na subseção 19.1.2.

11. SUPERFÍCIE DE COESÃO

Conforme já discutimos quando da apresentação do problema (veja subseção 9.6), o aumento da ordem e do tamanho da rede torna impraticável sua representação gráfica. Propusemos, então a troca de precisão na informação por representabilidade: em vez de desenhar as arestas, fazemos associar a cada vértice da rede escalonado sobre o espaço temático, uma altura que representa a coesão na vizinhança do vértice.

Por suavização e interpolação, calculam-se coordenadas para todos os pontos de um *grid* colocado sobre a base temática, obtendo-se a caracterização da superfície temática.

Pontos elevados na superfície temática indicam que naquela vizinhança, os pontos são intimamente ligados entre si: “morros” e “montanhas”, portanto, indicam grupos usuários com interesse parecido. “Vales”, por sua vez, indicam transição entre interesses temáticos.

Nós posicionados sobre picos caracterizam usuários com interesses especializados; nós posicionados sobre vales indicam usuários com interesses diversificados.

A Figura 24, a seguir, mostra que no relevo temático dos usuários do Exemplo 2 há três “morros” de interesse.

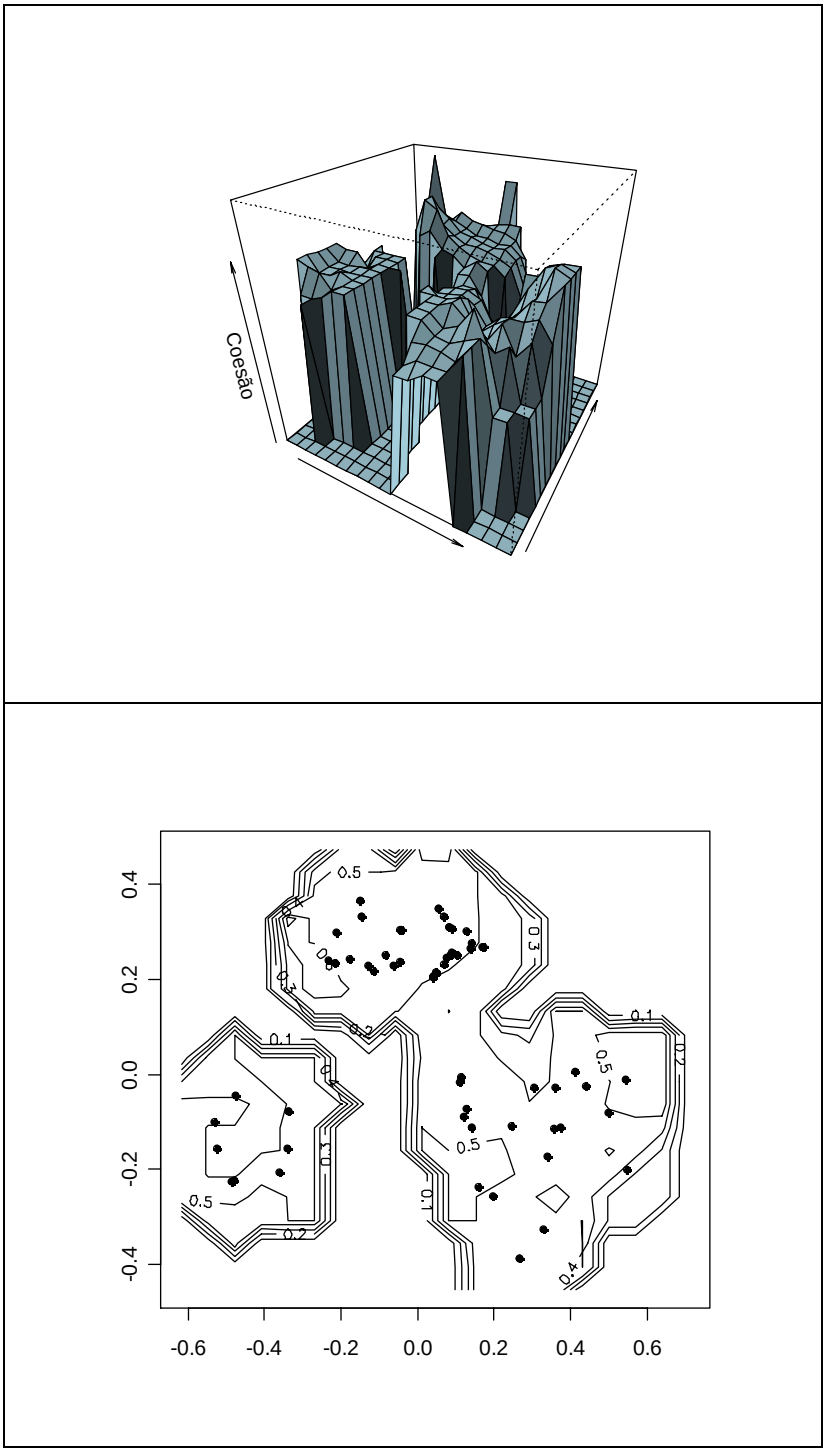


Figura 24 – Relevo Temático do Exemplo 2.

11.1. Interpretação do Relevo

Com ajuda da Figura 25 e dos dados originais do problema, podemos interpretar o conteúdo temático das três elevações do relevo do Exemplo 2.

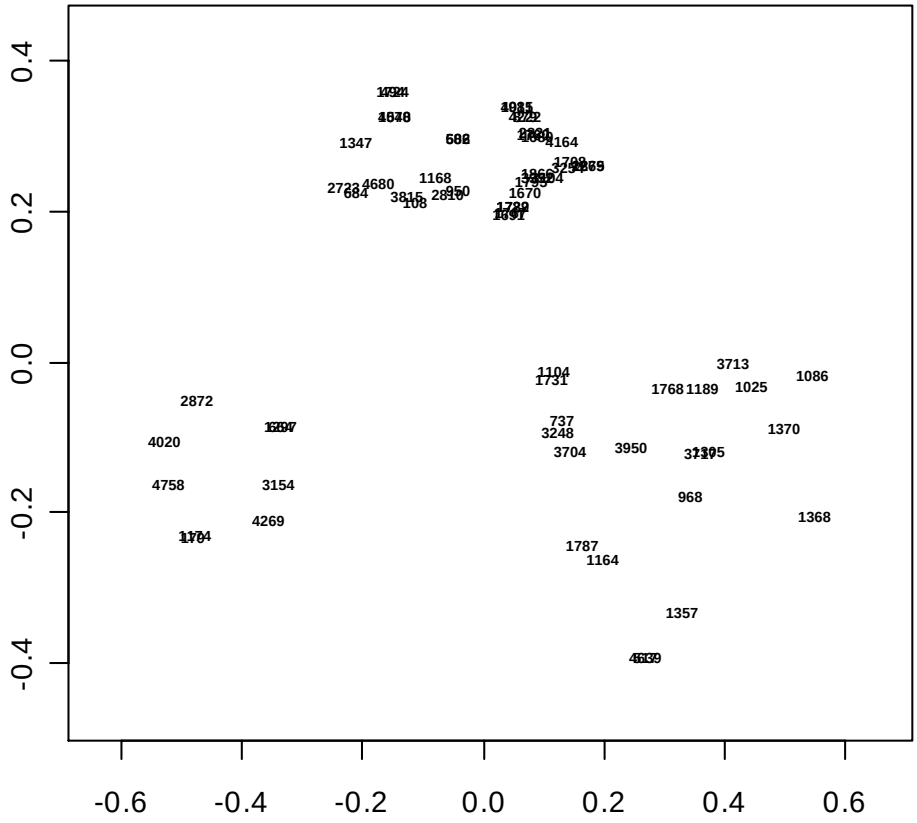


Figura 25 – Legenda dos Vértices da Rede do Exemplo 2.

O conjunto isolado, à esquerda (a Sudoeste), contém, os usuários 179, 664, 1174, 1297, 2872, 3154, 4020, 4269 e 4758. Os livros mais retirados por estes usuários

(isto é, com 3 ou mais ocorrências) estão listados na Tabela 5 e demonstram uma alta coesão em torno do assunto de Mercados Financeiros.

Tabela 5 – Livros Mais Retirados no "Morro Sudoeste"

Qt	Assunto	Autor	Título
14	Balanço : análise	Dante C Matarazzo	Analise financeira de balanços
8	Mercados financeiros futuros	J C Hull	Introdução aos mercados futuros e de opções
7	Moeda e política monetária	Joao do Carmo Lopes	Economia monetária
6	Estatística - matemática	Prem S Mann	Statistics for business and economics
5	Administração financeira : empresas	Stephen A Ross	Princípios de administração financeira
5	Investimento de capital em títulos	Elli Gifford	Investor's guide to technical analysis : predicting price ac
5	Mercado de ações brasileiro	Eduardo Fortuna	Mercado financeiro : produtos e serviços
4	Mercados financeiros futuros	J C Hull	Introdução aos mercados futuros e de opções
4	Levantamento de fundos - Assoc.sem fins lucrativos	James M Greenfield	Fund-raising fundamentals : a guide to annual giving for pro
4	Mercado de ações brasileiro	Eduardo Fortuna	Mercado financeiro : produtos e serviços
4	Organização : mudança	Michael Hammer	Reengenharia : revolucionando a empresa em função dos cliente
4	Estatística - matemática	Paul Newbold	Statistics for business and economics
3	Levantamento de fundos - Assoc.sem fins lucrativos	L Peter Edles	Fundraising : hands-on tactics for nonprofit groups
3	Administração de investimento	Robert A Haugen	Modern investment theory
3	Investimento de capital em títulos	George Soros	The alchemy of finance : reading the mind of the market
3	Investimento de capital em títulos	Fundos de Investimento Performance Rentabilidade	Fundos de investimentos : performance & rentabilidade
3	Mercado de ações futuro		Curso de futuros e opções
3	Mercadologia : pesquisa	Harper W Boyd	Pesquisa mercadológica : texto e casos
3	Mercados financeiros futuros	J C Hull	Introduction to futures and options markets
3	Planejamento empresarial - Processamento de dados	Don Tapscott	The digital economy : promise and peril in the age of networ
3	Mudança social	Paul M Kennedy	Preparando para o século XXI
3	Balanço : análise	Dante C Matarazzo	Analise financeira de balanços : abordagem básica e gere
3	Organizações não-governamentais - Administração	Lester M Salamon	The emerging nonprofit sector : an overview
3	Administração financeira : empresas	Lawrence J Gitman	Princípios de administração financeira
3	Organização : mudança		Reengenharia das empresas : passando a limpo
3	Mercadologia	Philip Kotler	Administração de marketing : analise, planejamento, im
3	Economia	Lester C Thurow	O futuro do capitalismo : como as forças econômicas de h
3	Estatística - matemática	Paul Gerhard Hoel	Estatística elementar
3	Contabilidade	Clyde P Stickney	Financial accounting : an introduction to concepts, metho
3	Administração	Gareth Morgan	Imagens da organização
3	Estatística - matemática	Pedro Luiz de Oliveira Costa Neto	Estatística
3	Levantamento de fundos - Assoc.sem fins lucrativos	Joseph R Mixer	Principles of professional fundraising : useful foundations
3	Organização : mudança	Michael Hammer	A revolução da reengenharia : um guia pratico

O morro abaixo e à direita (a Sudeste) é formado pelos usuários 517, 737, 968, 1025, 1086, 1104, 1164, 1189, 1305, 1357, 1368, 1370, 1731, 1768, 1787, 3248, 3704, 3713, 3717, 3950, 4639. Os livros mais retirados por estes usuários estão listados na Tabela 6 e sugerem uma coesão em torno do tema de Administração Financeira e Planejamento.

Tabela 6 - Livros Mais Retirados no "Morro Sudeste"

Qt	Assunto	Autor	Título
15	Administração financeira : empresas	Lawrence J Gitman	Princípios de administração financeira
14	Mercado de ações brasileiro	Eduardo Fortuna	Mercado financeiro : produtos e serviços
11	Cadeias de Markov	William F Stewart	Introduction to the numerical solution of Markov chains
11	Desenvolvimento econômico	Nali de Jesus de Souza	Desenvolvimento econômico
10	Decisão - administração	Barry Render	Quantitative analysis for management
8	Direito do trabalho brasileiro	Octavio Bueno Magano	Manual de direito do trabalho
7	Decisão - administração	James R Evans	Introduction to simulation and risk analysis
7	Administração de investimento	Paulo Roberto Vampre Hummel	Analise e decisão sobre investimentos e financiamentos :
7	Decisão - administração	James R Evans	Introduction to simulation and risk analysis
7	Custos : contabilidade	Masayuki Nakagawa	ABC : custeio baseado em atividades
7	Contabilidade	M W E Glautier	Accounting theory and practice
6	Serviço ao cliente		Mantendo clientes
6	Balanço : análise	Dante C Matarazzo	Analise financeira de balanços : abordagem básica e gere
6	Ciência	Thomas S Kuhn	A estrutura das revoluções cientificas
6	Tecnologia da informação	William H Inmon	Data architecture : the information paradigm
6	Metodologia Científica	Gilberto de Andrade Martins	Manual para elaboração de monografias e dissertações
5	Administração de investimento		Capital budgeting under uncertainty
5	Mercado financeiro	Luiz Fernando Rudge	Mercado de capitais
5	Características gerais da pesquisa	William Josiah Goode	Métodos em pesquisa social
5	Tecnologia da informação	Shao Yong Chu	Banco de dados : organização, sistemas e administração
5	Sociologia da comunicação	Marshall McLuhan	Os meios de comunicação : como extensões do homem
5	Planejamento empresarial - Processamento de dados	Alex Berson	Data warehousing, data mining, and OLAP
5	Estatística - matemática	Paul Newbold	Statistics for business & economics
5	Direito do trabalho brasileiro	Delio Maranhao	Direito do trabalho
4	Mercado de ações brasileiro	Eduardo Fortuna	Mercado financeiro : produtos e serviços
4	Administração de investimento	John D Finnerty	Project financing : asset-based financial engineering
4	Administração de investimento	Frank J Fabozzi	Investment management
4	Administração de investimento	Pierre Jacques Ehrlich	Engenharia econômica : avaliação e seleção de projetos d
4	Estatística - matemática	Mario F Triola	Business statistics : understanding populations and processe
4	Direito do trabalho brasileiro		I Ciclo de Estudos de Direito do Trabalho
4	Mercado financeiro	Luiz Fernando Rudge	Mercado de capitais
4	Macroeconomia	G F Stanlake	Macroeconomia : uma introdução
4	Mercadologia	Philip Kotler	Administração de marketing : (analise, planejamento e cont
4	Valor : administração da produção	Joao Mario Csillag	Analise do valor : metodologia do valor
4	Valor : administração da produção	Joao Mario Csillag	Analise do valor : metodologia do valor
4	Direito do trabalho brasileiro	Amauri Mascaro Nascimento	Iniciação ao direito do trabalho
4	Direito do trabalho brasileiro	Delio Maranhao	Direito do trabalho
4	Administração financeira : empresas	Harold Bierman	The capital budgeting decision : economic analysis of invest
4	Investimento de capital em títulos	Kenneth H Shaleen	Technical analysis & options strategies
4	Planejamento - administração de empresas		Estratégia : a busca da vantagem competitiva
4	Jogos - matemática	Eric Rasmusen	Games and information : an introduction to game theory
4	Desenvolvimento econômico	Nali de Jesus de Souza	Desenvolvimento econômico
4	Macroeconomia	Rudiger Dornbusch	Macroeconomia
4	Administração financeira : empresas	Neil Seitz	Capital budgeting and long-term financing decisions

Finalmente, a elevação mais acima (ao Norte) na Figura 25, inclui 36 usuários: por exemplo, os de *id* 850, 1168, 1347 e 4680. Os livros mais retirados por eles estão

contabilizados na Tabela 7 e mostram uma orientação temática para tecnologia da informação.

Tabela 7 - Livros Mais Retirados no "Morro Norte"

Qt	Assunto	Autor	Título
22	Tecnologia da informação	Richard E Walton	Tecnologia de informação : o uso de TI pelas empresas q
17	Tecnologia da informação	Aguinaldo Aragon Fernandes	Gerência estratégica da tecnologia da informação
11	Organizações não-governamentais - Administração	Julie Fisher	Nongovernments : NGOs and the political development of th
9	Mercadologia	Jill H Ellsworth	Marketing on the Internet
9	Desenvolvimento econômico	John Friedmann	Empowerment : the politics of alternative development
8	Investimento de capital em títulos	Frank K Reilly	Investments
8	Administração financeira : empresas	Lawrence J Gitman	Princípios de administração financeira
8	Tecnologia da informação	Norberto A Torres	Competitividade empresarial com a tecnologia de informação
7	Desenvolvimento econômico	Deborah Eade	The Oxfam handbook of development and relief
7	Investimento de capital em títulos	Lawrence J Gitman	Fundamentals of investing
6	Tecnologia da informação	Jose Davi Furlan	Megatendencias da tecnologia da informação
6	Administração de empresas	Max H Bazerman	Negociando racionalmente
5	Lógica	Irving M Copi	Introdução a lógica
5	Tecnologia da informação		Formato IBICT : formato de intercâmbio bibliográfico e ca
5	Produção : administração	Roger G Schroeder	Operations management : decision making in the operations fu
5	Tecnologia da informação	Luiz Fernando Ballin Ortolani	Produtividade da tecnologia da informação : evidencias
4	Balanço : contabilidade	Umberto Mandarino	Analise de balanço no mercado de capitais
4	Organização : mudança	Richard L Nolan	Destruição criativa : um processo de seis etapas para transf
4	Investimento de capital em títulos	William F Sharpe	Investments
4	Mercadologia	Robert R Reeder	Industrial marketing : analysis, planning, and control
4	Marketing internacional	Vern Terpstra	International marketing
4	Administração	Gareth Morgan	Imagens da organização
4	Estatística - matemática	Paul Gerhard Hoel	Estatística elementar
4	Mercadologia	Alberto Luiz Albertin	Comercio eletrônico : modelo, aspectos e contribuições
4	Contabilidade	Silverio das Neves	Contabilidade básica
3	Administração de empresas	William H Davidow	A corporação virtual : estruturação e revitalização
3	Matemática	Sebastiao Medeiros da Silva	Matemática para os cursos de economia, administração, ciência
3	Economia	Henri Denis	Historia do pensamento econômico
3	Mercadologia : pesquisa	David A Aaker	Marketing research
3	Administração de pessoal	Idalberto Chiavenato	Recursos humanos
3	Contabilidade	Jose Carlos Marion	Contabilidade empresarial

11.2. Estratégia de Implementação da Representação Gráfica

Para a elaboração da superfície temática, adotamos a estratégia descrita a seguir.

Inicialmente demarcamos a área ocupada pelo gráfico contendo os vértices da rede. Em seguida, colocamos sobre esta área um conjunto parametrizado de pontos de uma retícula. No exemplo, utilizamos uma retícula de 20 x 20 pontos (veja Figura 26).

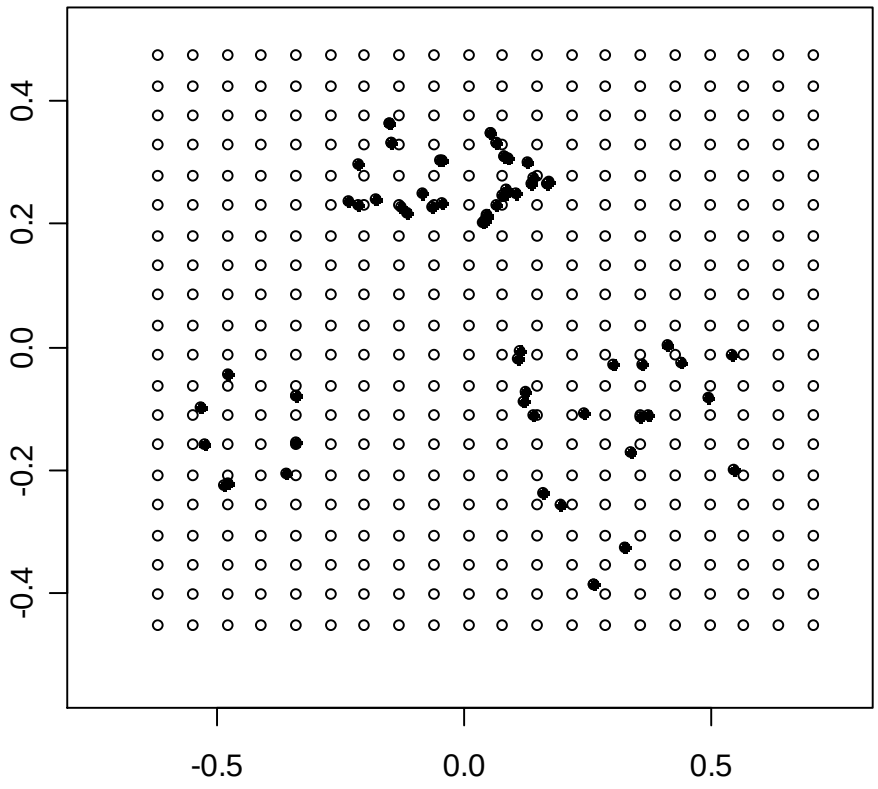


Figura 26 – Retícula sobre os Vértices do Exemplo 2.

Para cada ponto da retícula atribuiu-se a coesão média dos vértices que distavam menos de *d* do ponto em questão (no exemplo d = 2*dx, onde dx é a distância horizontal entre dois pontos na retícula). Os valores de coesão calculados em cada ponto da retícula foram passados para as funções *persp* e *contour* do R, que produzem, respectivamente uma perspectiva 3-D dos dados, e um gráfico de curva de nível.

12. PARTICULARIDADES DO PROBLEMA DE COOPERAÇÃO INDIRETA

Algumas características do tipo de rede formado na nossa aplicação foram repetidamente utilizadas como base de simplificações necessárias à viabilização do algoritmo proposto para sua representação gráfica por meio de escalonamento multidimensional das distâncias temáticas (seção 12). Estas características convenientes são discutidas a seguir.

12.1. Matriz de Adjacência Escassa

Numa matriz de adjacência de rede não orientada, o número de arestas possíveis é *n(n-1)/2*, onde *n* é o número de nós distintos na rede. Assim, com o aumento de nós, o número de arestas possíveis cresce segundo O(n^2).

Admitindo-se que o grau médio *g* dos nós é uma característica estável da rede, isto é, admitindo-se que *g* é constante, o número de arestas observadas é dado por *ng/2*. Assim, com o aumento de nós, as arestas crescem segundo O(*n*).

A densidade de arestas na rede, por sua vez, dada por *g*/(*n-1*) diminui à medida que a ordem da rede aumenta. Isso significa que a matriz de adjacência conterá uma proporção de zeros cada vez maior.

Assim, se de um lado, o tamanho da matriz de distâncias explode conforme aumenta o número de vértices a analisar, tornando-se computacionalmente intratável, por outro lado esta explosão não acontece com o número de caselas contendo o valor "1". Por isso, procuraremos trabalhar controlando apenas o endereço (índices) destas caselas – o que corresponde a trabalhar com a lista de arestas.

Na matriz do Exemplo 1, o grau médio dos nós é 2,6; as arestas observadas são 13; as arestas possíveis, 45. A densidade de arestas é 0,29. Cerca de 70% das caselas, portanto, contém zeros. Para economia, só é necessário acompanhar os 30% de caselas contendo uns. A economia aumenta drasticamente conforme aumenta o número de vértices na rede.

12.2. Matriz de Distâncias Temáticas Escassa

Da mesma forma que na matriz de adjacência predominam os zeros (indicando falta de ligação), também na matriz de distância temática dominam as caselas que indicam ausência de relação temática; apenas, agora, estas caselas assumem o valor da distância máxima, isto é, o valor 1. Podemos economizar recursos acompanhando numa Lista de Distâncias, logicamente equiparável a uma Lista de Arestas, apenas os valores diferentes de 1.

12.3. Distribuição das Distâncias Temáticas

Mesmo entre as Distâncias Temáticas calculadas e acompanhadas, predominam amplamente as distâncias grandes. Elas são, portanto, pouco informativas. Assim, sempre que possível, valorizamos no algoritmo a conservação da informação relativa a arestas pequenas.

12.3.1 Matriz D de Distâncias Temáticas

A matriz **D** de distâncias temáticas (5.1) entre os nós da rede do Exemplo 1, em forma de lista e calculada com base na Lista de Vizinhanças (LV), é apresentada na Saída 7. O histograma dos distâncias temáticas está apresentado na Figura 27.

Saída 7 – Matriz **D** de Distâncias (Temáticas) entre Usuários do Exemplo 1.

D

i	**X1**	**X2**	**X3**
1	100	101	0.4444444
2	100	102	0.5555556
3	100	105	0.7500000
4	100	107	0.2857143
5	100	108	0.2857143
6	101	102	0.5000000
7	101	105	0.8000000
8	101	107	0.4444444
9	101	108	0.4444444
10	102	105	0.7777778
11	102	107	0.5555556
12	102	108	0.5555556
13	105	107	0.7500000
14	105	108	0.7500000
15	107	108	0.2857143

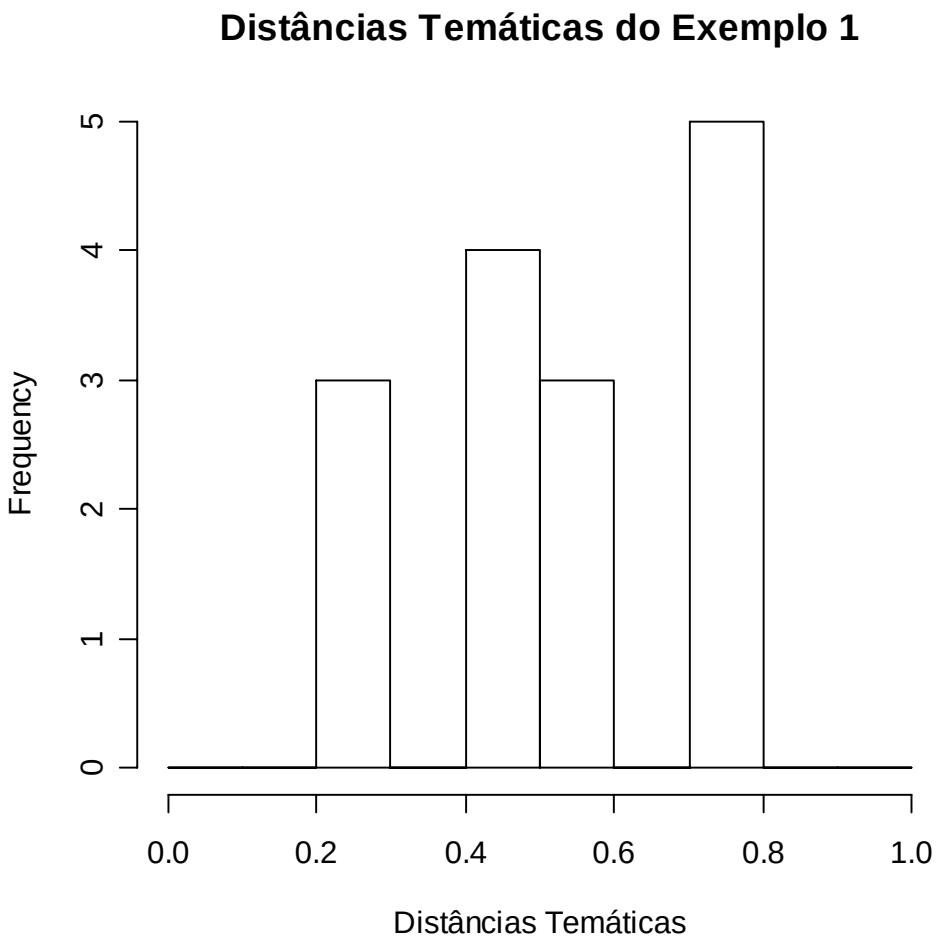


Figura 27 – Histograma das Distâncias Temáticas entre Usuários do Exemplo 1.

Como temos 6 usuários na rede, o número de distâncias entre eles é 6*5/2 = 15. A lista de Distâncias Temáticas tem também 15 entradas, uma para cada aresta; está, portanto, completa.

O arquivo para o Exemplo 2 tem originalmente 1264 registros, que se reduzem a 175 após o pré-processamento. Implicitamente na lista de transações há 71 usuários. Assim, há 71*70/2 = 2485 distâncias entre os nós. No entanto, a matriz de distâncias (temáticas) retornada pelo programa tem apenas 147 linhas, conforme reprodução parcial da Saída 8, representada no histograma da Figura 28, a seguir.

Saída 8 – Distâncias Temáticas entre Nós de Usuários do Exemplo 2.

D i	X1	X2	X3
1	108	1269	0.9230769
2	108	686	0.9230769
3	179	1866	0.9000000
4	179	1174	0.7000000
5	179	4020	0.7500000
6	179	4758	0.5000000
7	494	1724	0.5454545
8	494	1164	0.8461538
9	494	3254	0.8333333
10	494	3104	0.8333333
...	...	...	...
137	3248	4540	0.8000000
138	3352	4540	0.7777778
139	3950	4164	0.7000000
140	4011	4222	0.7777778
141	4020	4680	0.6666667
142	4020	4758	0.7000000
143	4269	4846	0.7500000
144	4639	4846	0.8181818
145	4682	4834	0.4000000
146	4682	5921	0.4000000
147	4834	5921	0.4000000

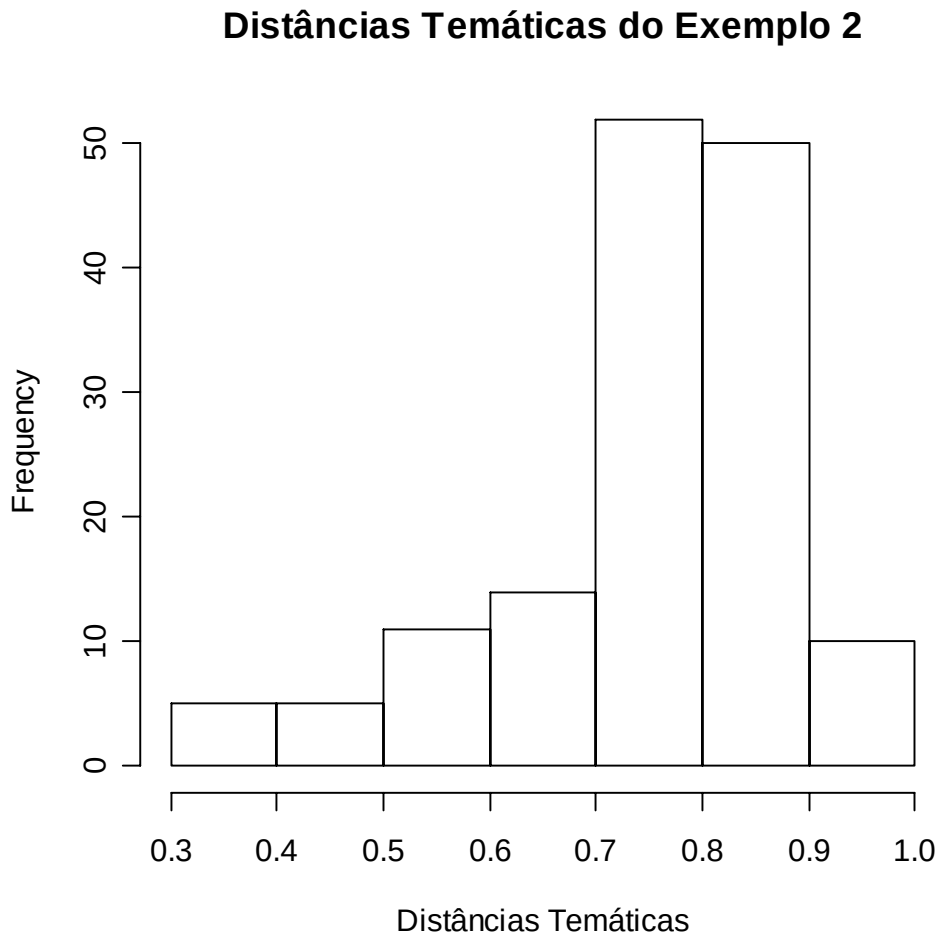


Figura 28 – Histograma das Distâncias Temáticas do Exemplo 2

O arquivo para o Exemplo 3 tem originalmente 4850 registros, que se reduzem a 2043 após o pré-processamento. Implicitamente na lista de transações há 596 usuários. Assim, há 596*595/2 = 177310 distâncias entre os nós. No entanto, a matriz de distâncias (temáticas) retornada pelo programa tem apenas 2445 linhas, conforme reprodução parcial da saída de computador, a seguir (Saída 9). O histograma correspondente está na Figura 29.

Saída 9 – Distâncias Temáticas entre Nós de Usuários do Exemplo 3

D i	**X1**	**X2**	**X3**
1	55	1106	0.6315789
2	55	1211	0.6111111
3	55	2381	0.6666667
4	55	3092	0.6818182
5	55	3382	0.7586207
6	55	3401	0.7666667
7	55	4639	0.8444444
8	55	1868	0.9285714
9	55	3248	0.9230769
10	55	1439	0.8888889
...	...	...	...
2436	4773	5931	0.7307692
2437	4778	4790	0.8695652
2438	4778	4838	0.8636364
2439	4781	4827	0.9090909
2440	4790	4838	0.5384615
2441	4806	4814	0.8095238
2442	4814	4815	0.9310345
2443	4834	5938	0.8846154
2444	4834	5175	0.8518519
2445	4834	5921	0.6000000

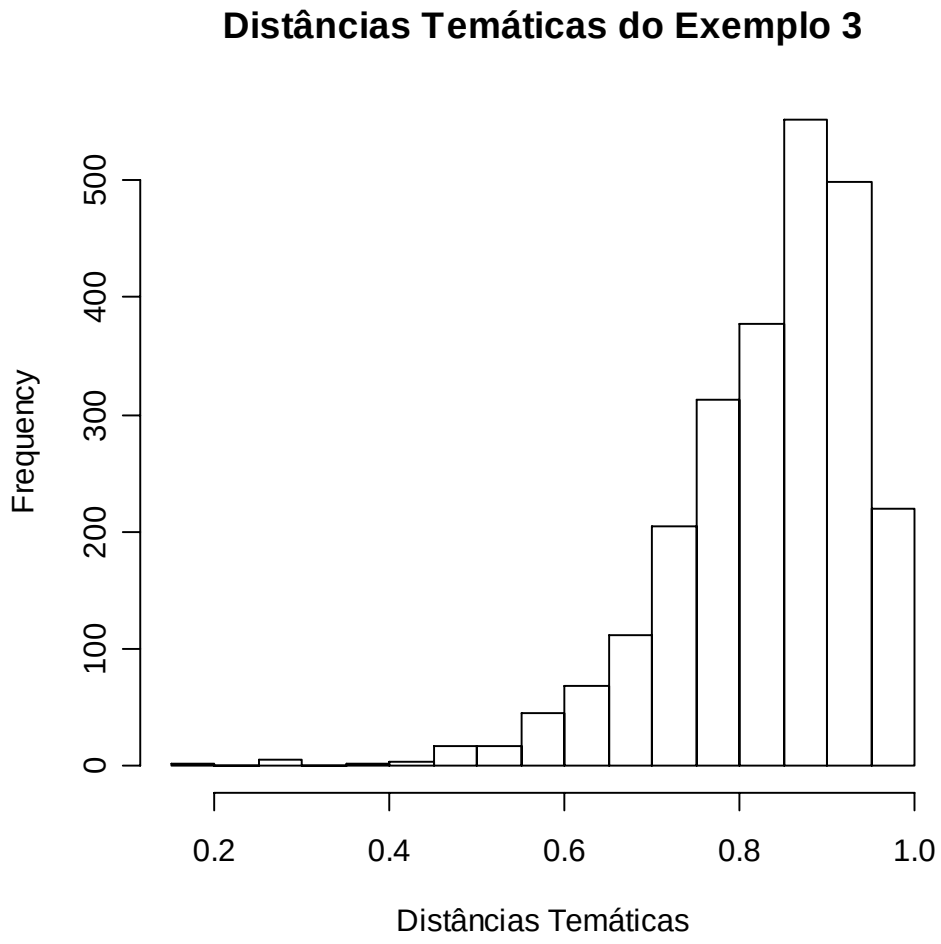


Figura 29 – Histograma das Distâncias Temáticas do Exemplo 3.

Pela comparação dos histogramas podemos constatar, a tendência para a ocorrência de distâncias temáticas grandes se acentua com o aumento do número de vértices. Isto é natural: quantos mais usuários são analisados, mais pares deles podem ser formados com pouco interesse comum.

12.4. Relevância Local

Uma idéia importada do Marketing Geográfico é que o impacto de uma ocorrência georeferenciada sobre a sua vizinhança decai com a distância. Transportada para o contexto de redes, pode-se assumir que vértices separados por muitos passos têm

progressivamente menos informação um sobre o outro. Com base nesta idéia, sempre que possível, montaremos a matriz de adjacência apenas nas vizinhanças dos vértices sob consideração em cada iteração do algoritmo.

Também no cálculo da distância temática o conceito foi importante para redução do número de distâncias a calcular. Por outro lado, a utilização de vizinhanças em mais de um passo pode representar um aperfeiçoamento do algoritmo (veja a subseção 19.1.2).

13. CÓDIGO COMENTADO

Nesta seção discutimos o algoritmo em si, em termos de sua programação.

13.1. R

O R é uma linguagem e um ambiente para computação estatística e gráfica. É similar à linguagem S desenvolvida nos Laboratórios Bell (anteriormente AT&T e no momento Lucent Technologies) por John Chambers e colegas. O R pode ser considerado uma outra implementação do S. Há diferenças de conceito importantes, mas a maior parte do código escrito em S "roda" em R.

É um *freeware*, e pode ser baixado a partir de www.r-project.org.

13.1.1 Vantagens

Além de fornecer uma grande variedade de funções estatísticas e gráficas, o R é **altamente flexível** para o desenvolvimento de extensões. Grande cuidado foi tomado para que os gráficos produzidos por este aplicativo tenham qualidade de impressão e sejam fáceis de manipular. Finalmente, tanto o S quanto o R são extremamente poderosos e econômicos na manipulação de vetores e matrizes. Por estes motivos, o S e o R freqüentemente são a linguagem de escolha na pesquisa estatística.

13.1.2 Desvantagens

A principal desvantagem que identificamos na utilização do R para a solução do nosso problema diz respeito ao gerenciamento de relacionamentos entre tabelas.

Por não ter funcionalidade relacional, do tipo utilizado em bancos de dados, grande cuidado deve ser tomado na manipulação de tabelas para manter a correspondência implícita na ordem dos resultados de manipulações com os dados originais armazenados em matrizes e vetores.

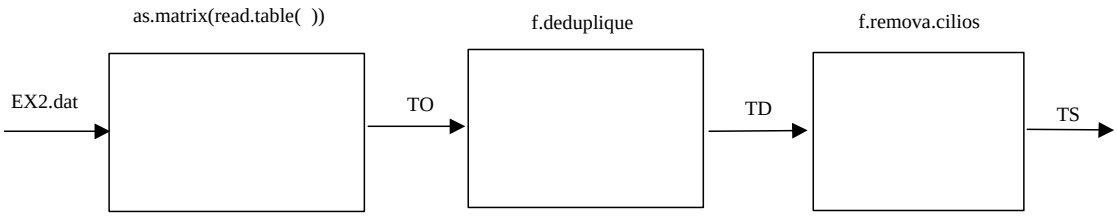
Em outras palavras, apenas a posição relativa das caselas em matrizes e vetores indica quais valores dizem respeito a quais objetos. Isso obriga o programador a

manter registros de índices que possibilitem associar o resultado dos cálculos aos objetos corretos.

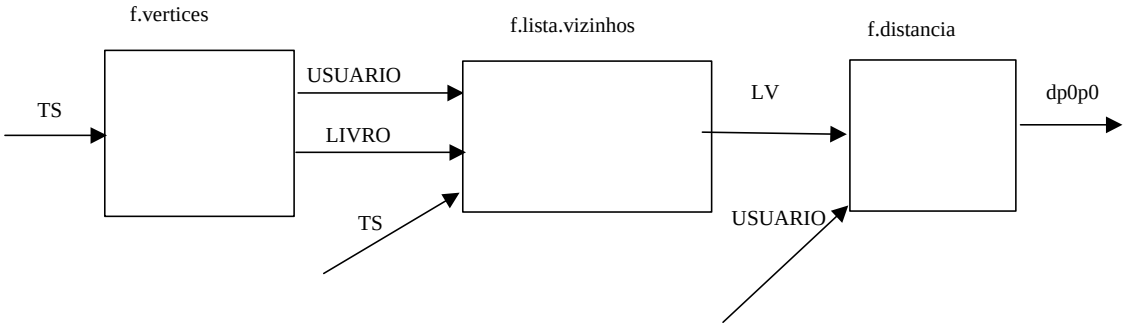
Por exemplo, no nosso caso, qualquer descuido poderia ter levado a associar a coesão calculada para um vértice ao vértice errado. Esta questão tornou-se muito delicada quanto constatamos um *bug* de programação que leva, durante o processamento, ao "desaparecimento" de alguns vértices (veja 10.3.2 e 19.1.1).

13.2. Fluxograma

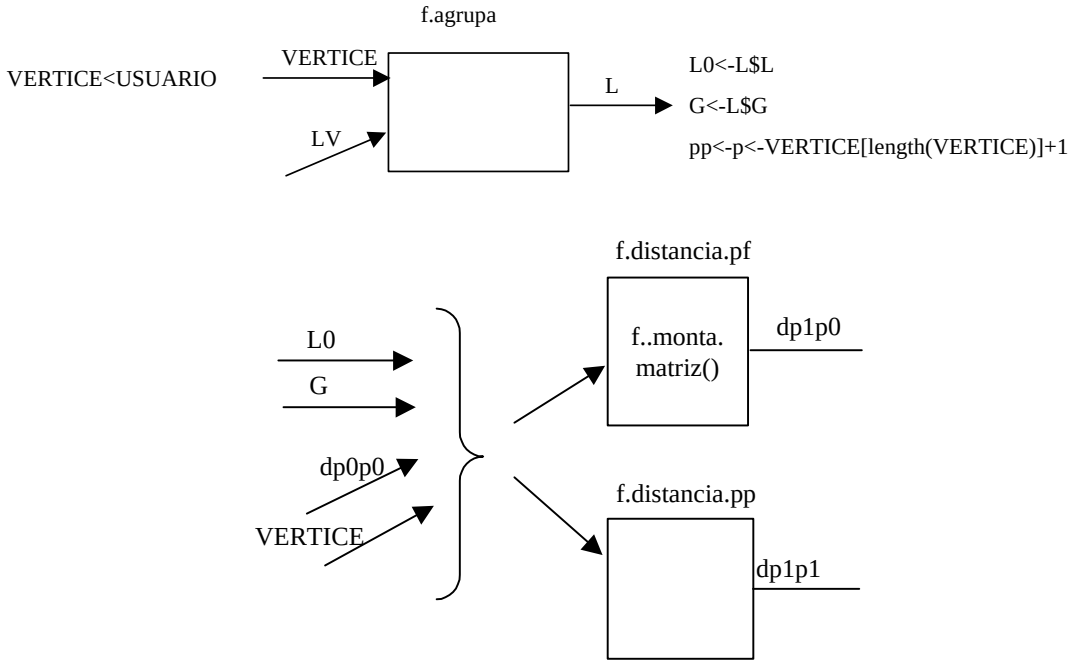
13.2.1 Pré-Processamento



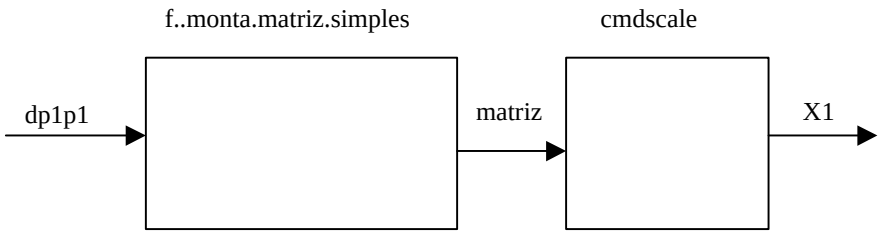
13.2.2 Lista Vizinhos e Distância Temática

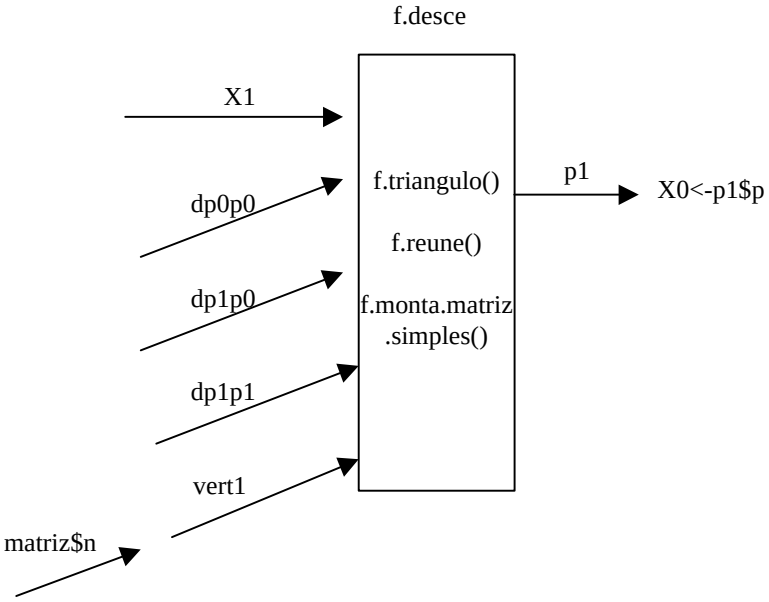


13.2.3 “Sobe”



13.2.4 “Desce”





13.3. Pseudo Código

13.3.1 Pré-Processamento

- Da matriz de dados de entrada, na forma de Lista de Arestas, remova os registros duplicados;
- Remova os "cílios"
- Identifique os vértices dos dois modos.

13.3.2 Lista Vizinhos e Distância Temática

- Com base na lista de arestas, produza uma Lista de Vizinhanças;
- Com base nas vizinhanças, calcule as Distâncias Temáticas entre os nós.

13.3.3 "Sobe"

- Agrupe os nós (filhos) em grupos próximos;
- Crie um representante (pai) para o grupo no nível de cima;
- Calcule as distâncias entre pais e filhos;
- Calcule as distâncias entre pais e pais.

- **Itere** o "Sobe" até o número de pais criados ser escalonável;
- Tendo atingido um nível em que os pontos são escalonáveis, escalone-os e fixe suas coordenadas.

13.3.4 "Desce"

- Tome três vértices do nível de cima; reúna a distância entre eles, a distância entre eles e seus filhos, e a distância entre estes filhos;
- Escalone este conjunto;
- Deforme o espaço da solução de maneira que as coordenadas dos pais se ajustem às coordenadas fixadas no nível de cima;
- **Itere** (dentro do "Desce") até todos os vértices do nível de cima, e por conseqüência todos os filhos neste nível, tenham sido escalonados;
- **Itere** (para baixo) até todos os níveis terem sido processados, isto é, até os pontos do Nível 0 terem sido fixados.

14. COMPARAÇÃO DE DUAS SOLUÇÕES: MDS CLÁSSICO E ALGORITMO PROPOSTO

A Figura 30 e a Figura 31 a seguir trazem, respectivamente, o escalonamento das distâncias temáticas por MDS Clássico para 2 dimensões (k = 2); e o escalonamento obtido pela aplicação do algoritmo proposto neste trabalho. Apesar da impressão de que uma rotação de cerca 135º da primeira figura a tornaria bastante equivalente à segunda, na verdade a correspondência entre elas é fraca: o coeficiente de procrustes entre ambas é de 0,73.

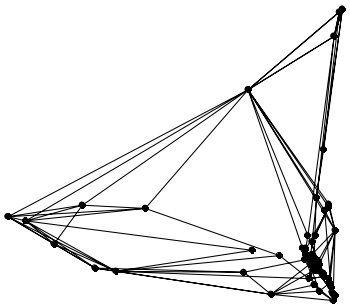


Figura 30 – MDS Clássico para k = 2

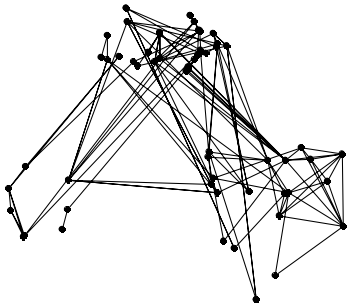


Figura 31 – Algoritmo Proposto

14.1. Para a Aplicação, o MDS Clássico Não É Boa Referência

Apesar do que pode parecer à primeira vista, no entanto, isto não é um problema. Na verdade, nas condições da aplicação para a qual o algoritmo foi desenvolvido, soluções próximas às obtidas pelo MDS Clássico não são desejáveis. Em primeiro lugar porque são muito arbitrárias; em segundo porque apresentam um stress (veja maiores detalhes na subseção 14.1.2) maior para as arestas de interesse.

14.1.1 MDS Clássico Fornece Solução Praticamente Arbitrária

Já constatamos em seções anteriores ser a matriz de distâncias entre os pontos que desejamos escalonar amplamente dominada por distâncias de valor 1; e que as distâncias diferentes de 1 são valores altos, isto é, próximos de 1.

Se cada distância é considerada uma aresta e cada ponto um vértice, a configuração corresponderia a um poliedro **quase regular** e com dimensão *n-1* (ou muito próxima desta valor), onde *n* é o número de vértices.

Estamos, portanto, tratando de um objeto geométrico volumoso, e projetando-o num plano. Há uma enorme perda de informação.

No caso do Exemplo 2, estamos trabalhando com um objeto de 66 vértices, que, no máximo, precisaria de um espaço 65-dimensional para ser acomodado integralmente. Conforme argumentamos, espera-se que o volume efetivamente ocupado atinja as 65 dimensões.

A análise por Componentes Principais das coordenadas fixadas por escalonamento clássico em 65 dimensões confirma esta expectativa: revela que cada dimensão contribui quase igualmente, com 1,5% da informação sobre a posição dos pontos. O *scree-plot* da Figura 32 reflete esta situação. Note que a escala do eixo vertical está truncada em 0,996.

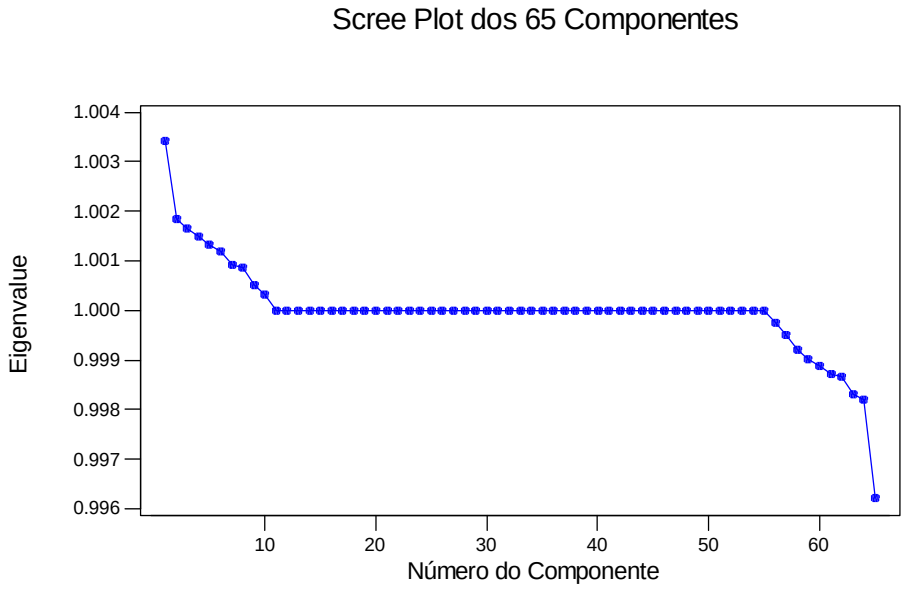


Figura 32 – Scree-plot dos Componentes do Exemplo 2

Disso conclui-se que, apenas acompanhando as direções dos eixos, há 65*64/2 = 2080 projeções ortogonais bidimensionais alternativas e equivalentes em termos de variância, para a configuração estudada.

14.1.2 Algoritmo Proposto Produz Stress Menor Onde Interessa

Definimos *stress* para efeitos de discussão nesta subseção como uma medida de desajuste entre a Distâncias Temática (∂) e a distância calculada com base na configuração fixada por escalonamento (*d*).

$$stress = \sum(\partial - d)^2$$

Como já discutimos (veja subseção 12.3) as distâncias que carregam informação para o problema são aquelas diferentes de 1 e que constam de *dp0p0*. Assim, calculando-se o *stress* para as duas soluções apresentadas na Figura 30 e na Figura 31, obtemos respectivamente 61,52 e 47,04, com nítida vantagem para o algoritmo proposto.

De fato, mesmo considerando-se o escalonamento clássico nas 65 dimensões, e estudando-se o *stress* nas possíveis projeções de 2 dimensões ao longo dos eixos daquele espaço, obtemos a distribuição refletida no histograma da Figura 33. Nota-se que o *stress* obtido com o algoritmo proposto (47,04) é menor do que aquele obtido em qualquer das 2080 projeções no espaço escalonado por MDS Clássico. Dentre estas últimas, o menor valor é de 55,64.

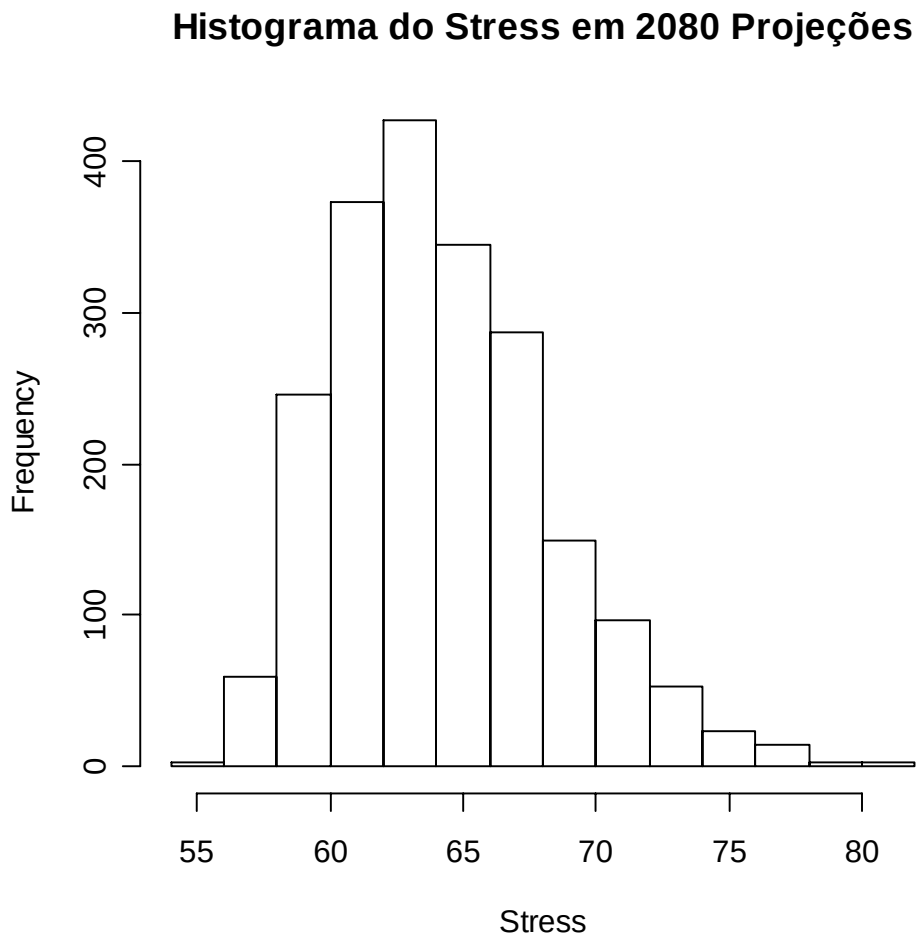


Figura 33 – Distribuição do *Stress*
nas Projeções ao Longo dos Eixos do Espaço de 65 Dimensões
Tomados 2 a 2.

Na verdade, os motivos para isso parecem claros: como o MDS Clássico necessita que a matriz de distância seja montada integralmente, o peso relativo das distâncias diferentes de 1 é ali menor do que nas soluções parciais, depois consolidadas, calculadas pelo algoritmo proposto neste trabalho.

Desta característica do algoritmo proposto resulta que os vértices tematicamente mais próximos ficam mais próximos na configuração obtida; as distâncias mais distorcidas são as que não interessam.

Finalmente, depreende-se que este fenômeno se acentua quando o número de vértices aumenta, o que torna o algoritmo proposto ainda mais desejável (sem mencionar que o MDS Clássico torna-se computacionalmente inaplicável).

Quarta Parte: Estabilidade da Solução

15. ESTABILIDADE DAS REPRESENTAÇÕES DE REDES PRODUZIDAS PELA APLICAÇÃO DO ALGORITMO RGR

Além dos aspectos de utilidade, facilidade de uso, escalabilidade e economia de recursos, pelo menos duas preocupações adicionais devem ser endereçadas quando se propõem e desenvolvem algoritmos para visualização de grandes bases de dados descritoras de redes.

A primeira é que a representação gráfica seja "iluminadora", isto é, que produza compreensão nova e relevante sobre o conjunto de dados em análise. Nos itens 11 e 14 procuramos demonstrar que o algoritmo proposto atende este requisito: discutimos as Superfícies de Coesão e sua interpretabilidade; e argumentamos que a performance de escalonamento do algoritmo é melhor, em vários aspectos, do que o MDS clássico.

Nesta seção debruçamo-nos sobre a segunda preocupação: a preocupação de que a representação gerada pelo algoritmo reflita uma realidade dos dados e não um produto da técnica ou uma variação aleatória introduzida pela amostragem. Quando o banco de dados, ainda que grande, é apenas uma amostra de um conjunto maior, é preciso que a representação gerada pelo algoritmo não seja demasiadamente impactada pela variação amostral. Em outras palavras, e especificamente no caso do algoritmo RGR, é preciso que a posição relativa dos nós da rede não seja demasiadamente alterada pela inclusão ou exclusão (aleatória) de um ou outro nó; e que, analogamente, a superfície de coesão não mude muito em decorrência das transações incluídas, ou não, na análise.

A discussão destes últimos dois pontos está organizada nas sete subseções a seguir.

15.1. Estratégia Global da Averiguação da Estabilidade da RGR

Inicialmente, postulamos que a representação construída com aplicação do algoritmo de Representação de Grandes Redes (RGR) proposto neste trabalho,

para ser útil, precisa ser robusta e resistir bem à falta de informação inerente à representação de redes incompletas. Se, de uma amostra para outra o analista encontrar resultados que variem radicalmente, corre o risco de começar a interpretar como sinal o que é apenas ruído aleatório.

Para investigar o efeito da variação amostral sobre a representação da rede, adotamos a seguinte estratégia:

- Definimos um conjunto de dados como sendo uma rede completa e calculamos sua superfície de coesão sobre sua base multidimensionalmente escalonada. Consideramos este conjunto como sendo a População; e consideramos sua representação como sendo a "verdadeira" representação da rede;

- Definimos um parâmetro α que regula a probabilidade das arestas da rede serem incluídas numa amostra; quanto maior o α, maior a probabilidade de inclusão e maior o tamanho da amostra.

- Para níveis selecionados de α, simulamos a extração de 500 amostras (réplicas) e estudamos a distribuição amostral: das coordenadas de vértices selecionados; da coesão destes vértices; e das superfícies de coesão calculadas. Com base (nos gráficos) destas distribuições tiramos nossas conclusões sobre a estabilidade das representações obtidas com a aplicação do algoritmo.

Vejamos detalhes desta estratégia, a seguir.

15.2. "População"

Para o estudo da estabilidade da saída do algoritmo RGR por veio de simulação de amostras, adotamos como População de usuários e livros os nós integrantes do Exemplo 2, formado, após o pré-processamento, por 71 usuários e 66 livros (veja seção 8). O algoritmo RBR foi utilizado para atribuir coordenadas e medidas de coesão a estes nós; a representação assim obtida foi tomada como a "verdadeira" representação da base de dados. Os

gráficos correspondentes estão na Figura 15, em que se observa claramente a existência de um relevo formado por três "morros": a Sudoeste o grupo temático de Administração Financeira e Planejamento; ao Norte o grupo temático de Tecnologia de Informação; e ao Sudeste o grupo temático de Mercados Financeiros, conforme discussão realizada na seção 11.

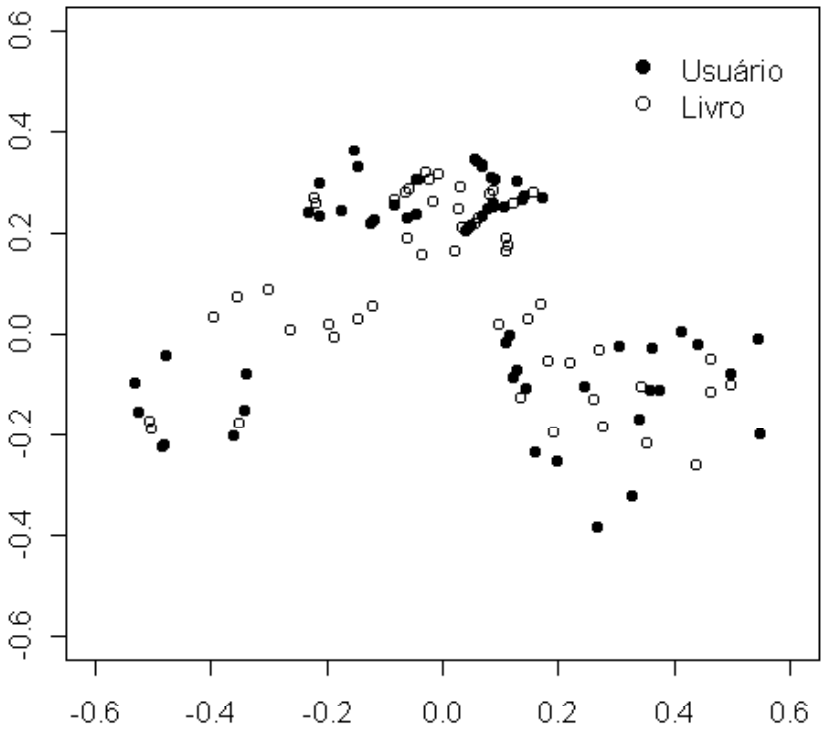


Figura 34 – Posição "Verdadeira" dos Vértices de Usuários e de Livros Integrantes da "População" Formada pelos Nós do Exemplo 2.

A Figura 34, acima, além dos vértices de usuários (círculos cheios), apresenta também a posição dos vértices de livros (círculos vazados). As coordenadas dos livros foram calculadas como a média das coordenadas dos usuários que os retiraram.

Uma analogia ilustra a lógica desta opção. É como se cada livro estivesse ligado ao usuário que o retirou por uma pequena banda elástica, todas do mesmo tamanho; a posição final do livro no espaço temático reflete o equilíbrio destas tensões. O mecanismo pode ser visualizado na Figura 35, através de um gráfico completo (painel superior) e de sua representação simplificada (painel inferior).

No painel superior, temos os usuários representados por pontos cheios (sem legenda), e os livros apenas através de suas legendas (sem pontos). Alguns livros foram destacados para análise. O livro 14241 aparece ligado por arestas aos usuários 2872 e 3815; o livro 14386, ligado aos usuários 1189, 1691, 1722, 1761 e 1789; finalmente, o livro 14282 está ligado aos usuários 517 e 1104. No painel inferior, os três livros e suas ligações foram mantidos; os demais nós foram suprimidos; acrescentaram-se legendas para os nós de usuários e pontos vazados para os nós de livros.

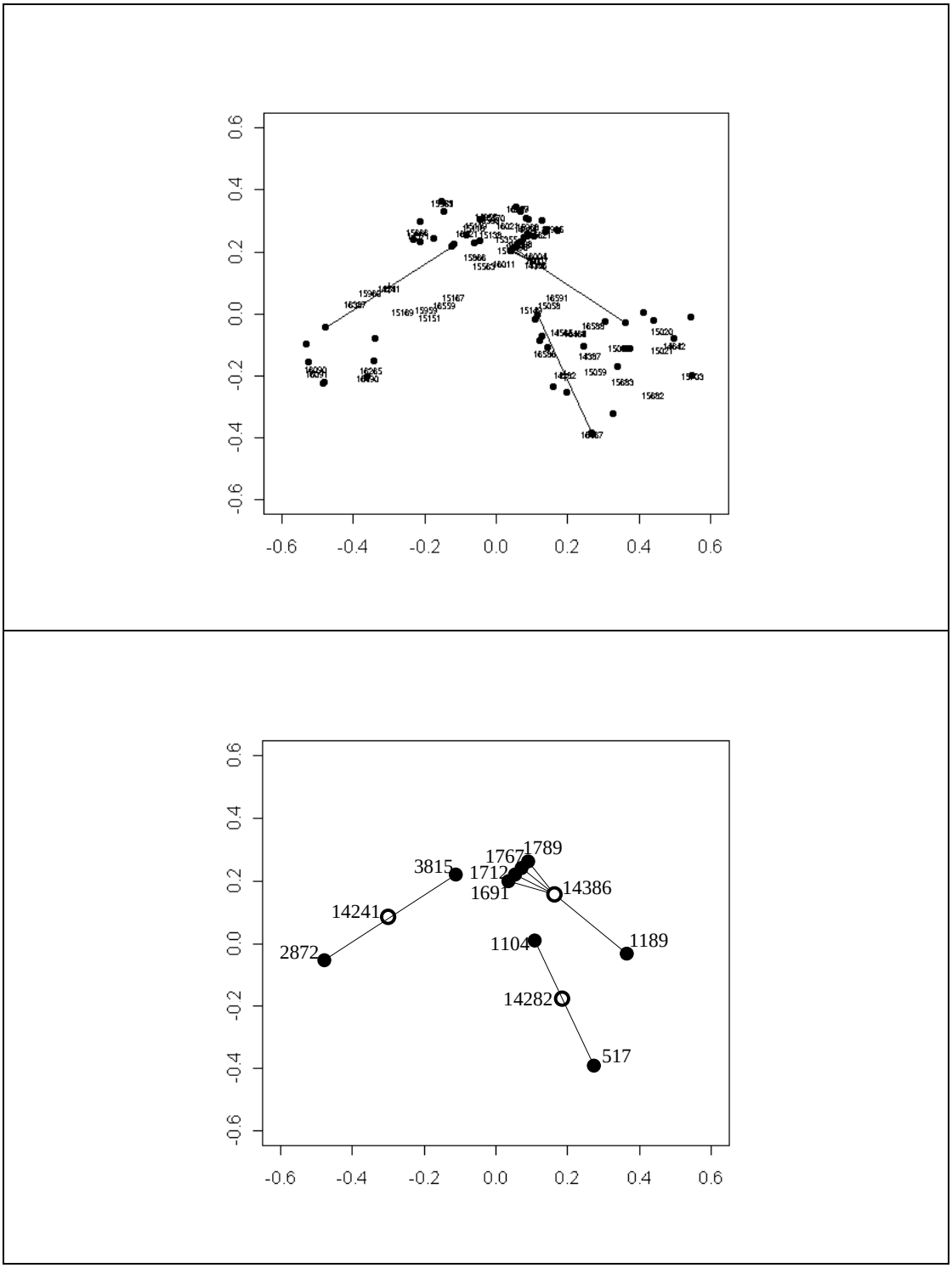


Figura 35 – Posicionamento dos Livros em Relação aos Usuários.

Observa-se nos painéis da Figura 35 que o livro 14241 resultou posicionado a meio caminho entre seus dois usuários; da mesma forma, o livro 14282; já o livro 14386 resultou mais próximo dos usuários 1691, 1712, 1767 e 1789 do que do usuário 1189, por ser mais "puxado" por aqueles do que por este.

Uma medida de interesse no processo de amostragem (seção 15.3, a seguir) é a distância entre cada usuário e todos os livros. Para ilustrar comportamento destes valores, apresentamos na Figura 36 o histograma da distância de todos os livros aos usuários destacados para a elaboração dos gráficos na Figura 35. Ressaltamos que os usuários 1347 e 1168 estão posicionados no "morro" Norte; os usuários 4020 e 3154, no "morro" Sudoeste; e os usuários 1189 e 968, no "morro" Sudeste. Observe como cada par tem uma distribuição mais semelhante dentro dos pares do que entre eles.

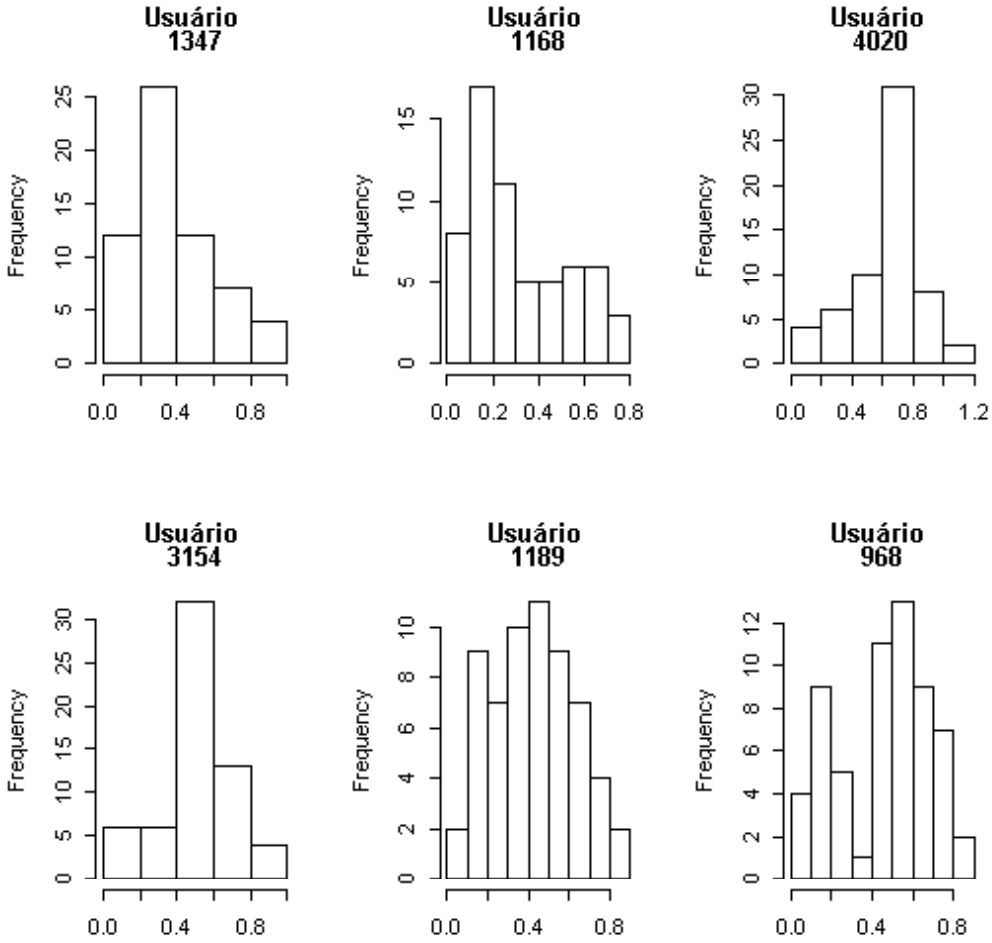


Figura 36 – Histograma das Distâncias Entre Usuários Selecionados e Todos os Livros da População.

15.3. Processo de amostragem

Para avaliar como muda a representação de rede em função da variabilidade amostral, utilizaremos um processo de simulação de amostras, criando um grande número de réplicas extraídas da População definida em 15.2. Submetidas ao algoritmo RGR o que se espera é que a representação para cada réplica seja robustamente semelhante à "verdadeira" superfície da População.

Os critérios adotados para a amostragem foram os seguintes. Toma-se seqüencialmente cada usuário integrante da População. Identifica-se, na População, o conjunto de arestas incidentes sobre ele. A cada uma destas arestas atribui-se uma probabilidade segundo a função:

$$p_{ij} = \frac{\frac{1}{d_{ij}^{\alpha}}}{\sum_{j} \frac{1}{d_{ij}^{\alpha}}}$$

onde

p_{ij} = probabilidade da aresta do usuário *i* ao livro *j* entrar na amostra
d_{ij} = distância temática do usuário *i* ao livro *j*
α = grau de sensibilidade à distância temática, $-\infty \leq \alpha \leq \infty$

A idéia básica foi atribuir a cada aresta existente na População uma probabilidade relativamente pequena de entrar na réplica; e que esta probabilidade diminuísse conforme o livro se afastasse do usuário. O parâmetro α regula a intensidade do efeito deste afastamento. Nos termos originais do problema, este comportamento equivale a simular a possibilidade de um usuário retirar um livro da biblioteca de forma a ser ela tão menor quanto mais afastado tematicamente do usuário estivesse o livro.

Posteriormente a todos os cálculos realizados, constatou-se que o número de livros consultados pelos usuários acabou interferindo de maneira imprevista na atribuição de probabilidades, mas apesar disso consideramos o resultado geral adequado para a simulação, pois alfas crescentes produziram, para um usuário fixado, distribuições de probabilidades com valores relativamente baixas (como desejado) e com médias crescentes (como desejado). Este efeito global está exemplificado na Figura 37, onde se vê o histograma das probabilidades associadas às arestas potenciais do usuário 1347, aos diversos níveis de α utilizados na simulação.

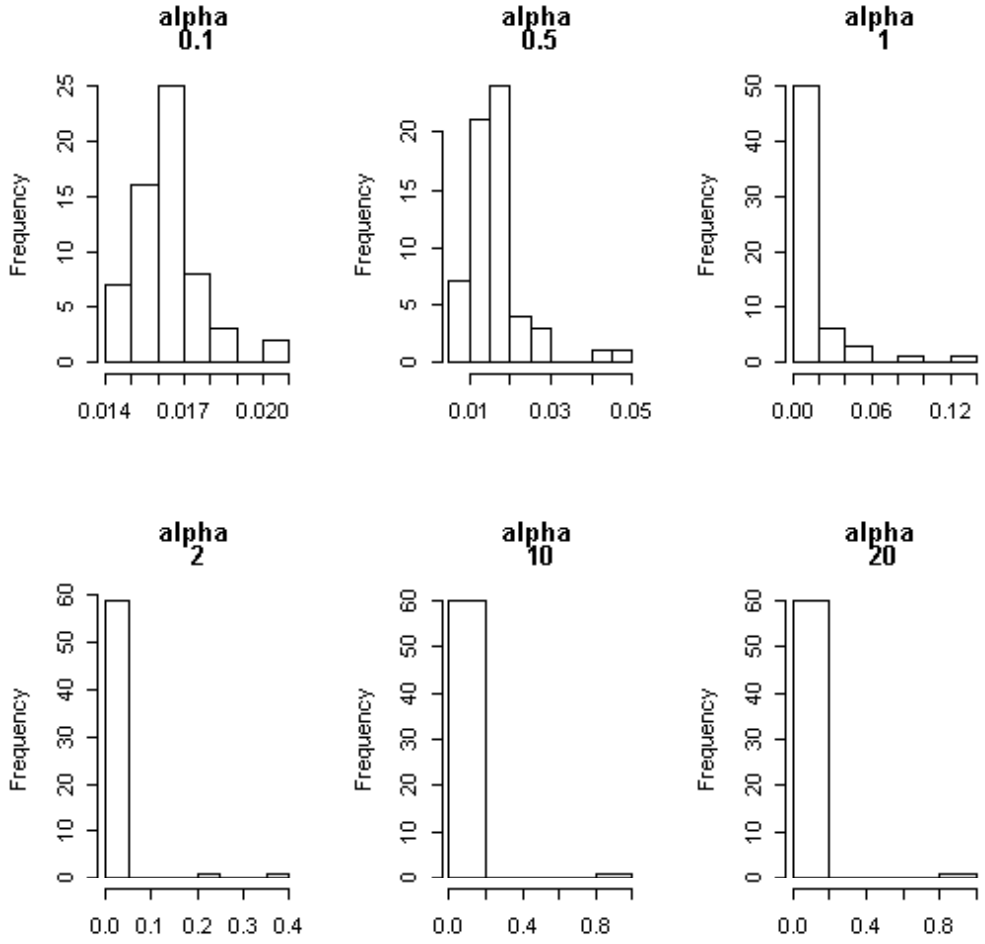


Figura 37 – Histograma das Probabilidades de Livros Serem Sorteados pra o Usuário 1347, Em Vários Níveis de α.

15.4. Número de Réplicas

Para cada nível de α, foram produzidas500 réplicas e submetidas ao algoritmo RGR. Como se sabe, o algoritmo está baseado na aplicação do Escalonamento Multidimensional Clássico e dele "herda" algumas características; destas, é relevante, neste ponto da discussão, o fato de que a representação gráfica da base temática multidimensionalmente escalonada

pode resultar com uma orientação qualquer; isto é, selecionadas as representações de duas amostras quaisquer, para elas serem geometricamente comparáveis, uma delas precisa ser rotacionada, transladada, deformada, ou sofrer uma combinação destes movimentos.

Assim, antes de qualquer comparação, procuramos alinhar a representação da amostra à representação da população, aplicando Procrustes aos pontos comuns entre ambas. Como era de se esperar, o resultado do ajuste foi influenciado pelo nível de α. Veja a Figura 38, a seguir.

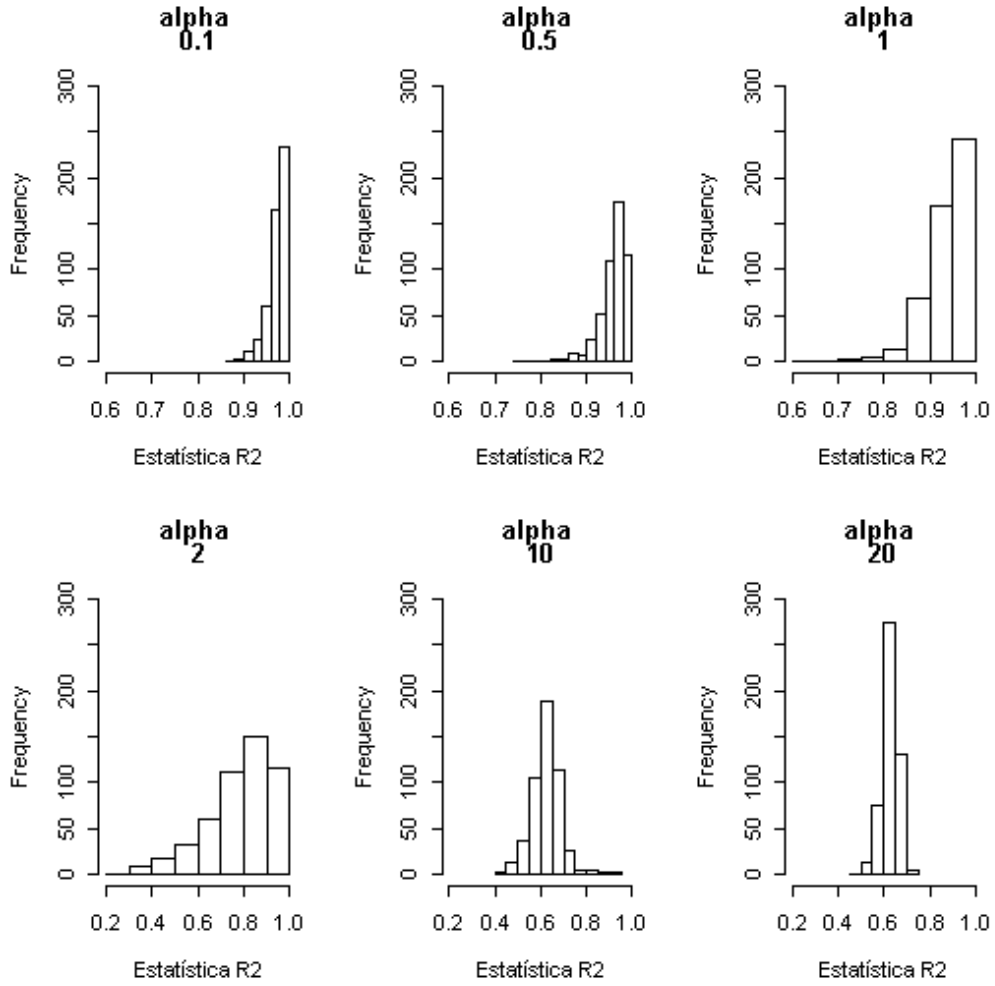


Figura 38 – Histograma do R^2: 500 Réplicas por Nível de α.

Nota-se que, à medida que aumenta o valor de α, e portanto a probabilidade das arestas integrarem a amostra, e portanto o tamanho da amostra, o ajuste de Procrustes da amostra em relação à população torna-se mais estável em torno de um valor pior (0,6). Voltaremos ao tema mais adiante na síntese do capítulo (seção 15.7).

15.5. "Passeio" e Coesão de Usuários Selecionados

Para a análise a seguir, voltamos aos usuários selecionados para exemplificações na seção 15.2. Recordemos que, na seqüência, cada par de usuários

destacados (1347 e 1168; 4020 e 3154; 1189 e 968) pertence aos “morros” Norte, Sudoeste e Sudeste, respectivamente.

A discussão realizada neste tópico está apoiada na Figura 39, que se estende pelas próximas 12 páginas. Duas páginas de gráficos foram dedicadas a cada usuário enfocado.

Para cada usuário foram considerados 6 níveis de α, ordenados do nível que gera amostras menores para o nível que gera amostras maiores.

Em cada nível há três gráficos:

- O primeiro, à esquerda da linha correspondente a um determinado valor de α, representa a posição de todos os usuários da população (círculos vazados) na configuração “verdadeira”; a posição (“verdadeira”) na população do usuário destacado (círculo cheio); a posição do usuário destacado em cada uma das 500 réplicas realizadas (sinal de +). Pode-se, neste gráfico observar onde é representado o usuário de interesse em cada uma das amostras, e seu “passeio” amostral.

- O segundo, no meio da linha, representa a coesão calculada, em cada réplica, para o usuário em foco, plotada em função de coordenada X calculada para o nó. Pode-se, portanto, observar neste gráfico o “passeio amostral” da coesão. A linha horizontal denota o valor “verdadeiro” da coesão naquele nó de usuário.

- Finalmente, o terceiro gráfico, à direita da linha correspondente, é análogo ao anterior e representa a coesão em função da coordenada Y calculada para o nó de usuário destacado.

Uma análise detalhada e comparativa dos gráfico permite propor as seguintes conclusões.

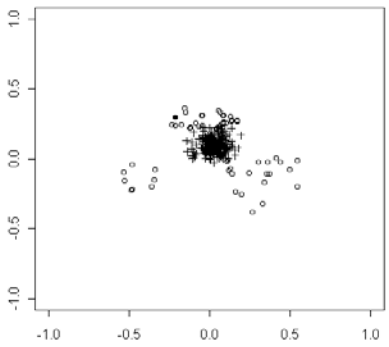
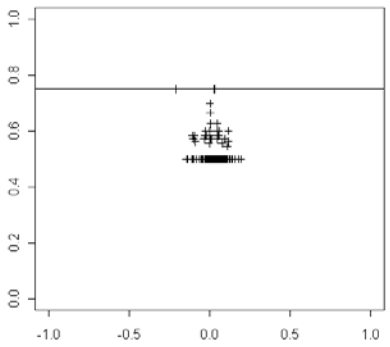
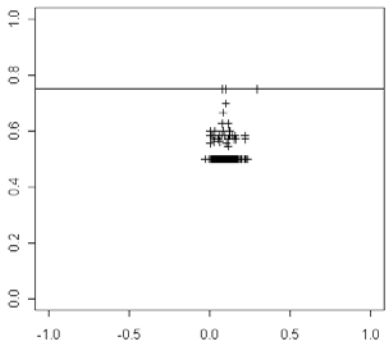
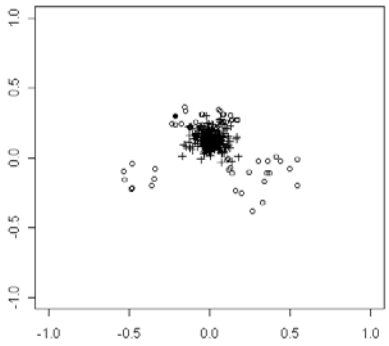
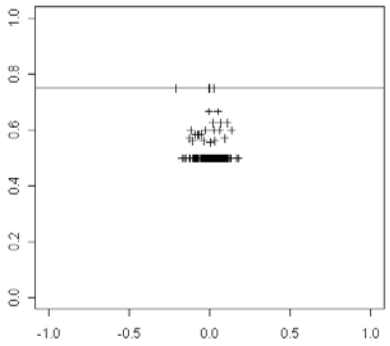
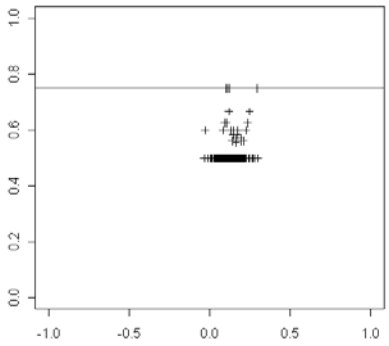
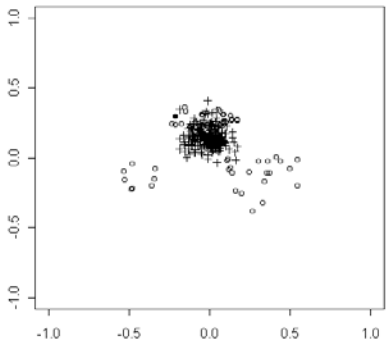
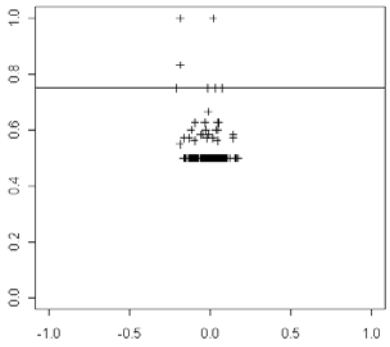
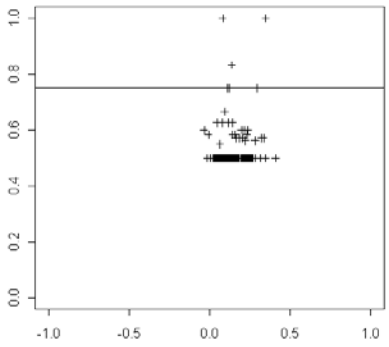
Passeio do Usuário 1347	Coesão em x	Coesão em y
α=0,1		
		
α=0,5		
		
α=1		
		

Figura 39 – “Passeio” de Usuários Selecionados (continua na página seguinte).

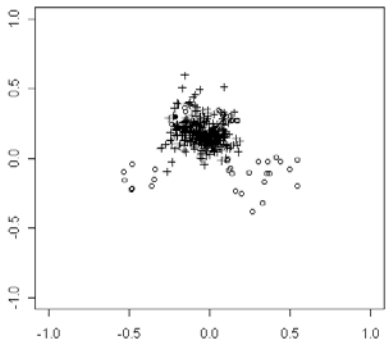
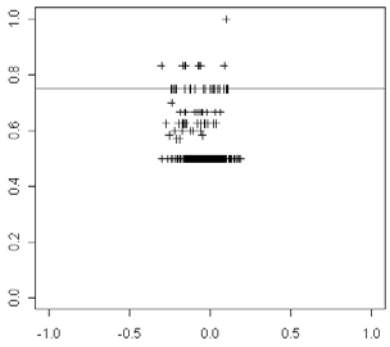
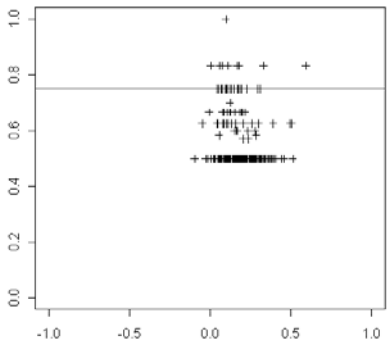
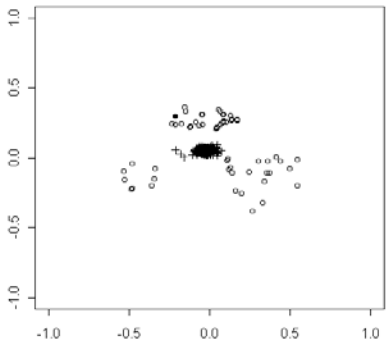
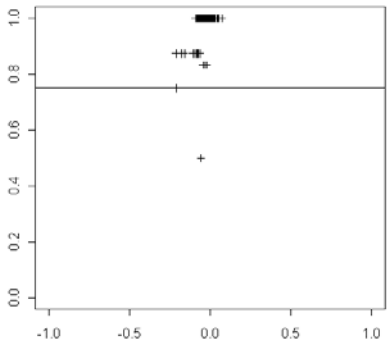
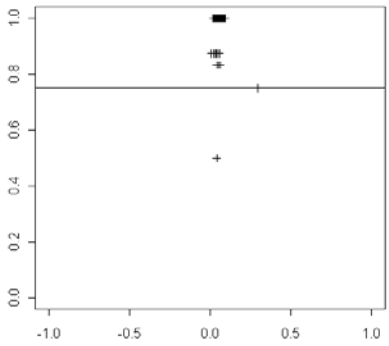
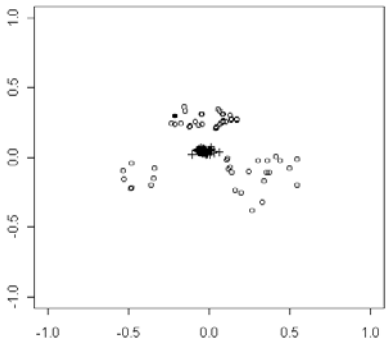
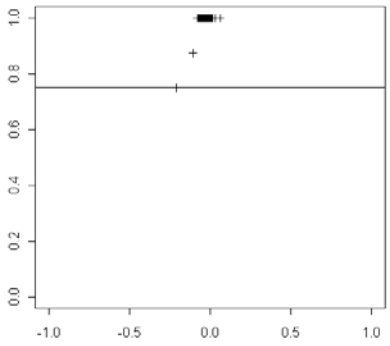
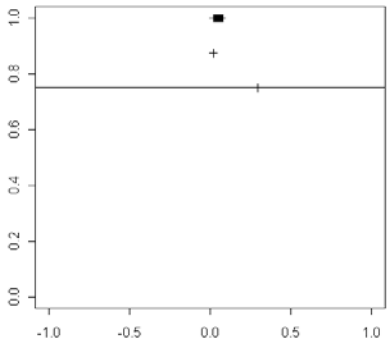
Passeio do Usuário 1347	Coesão em x	Coesão em y
α=2		
		
α=10		
		
α=20		
		

Figura 39 – “Passeio” de Usuários Selecionados (continua na página seguinte).

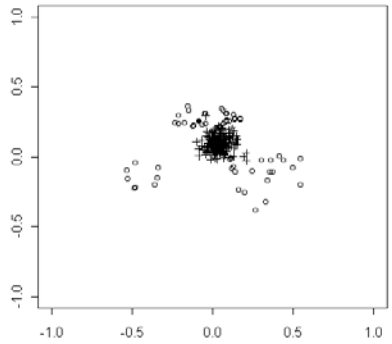
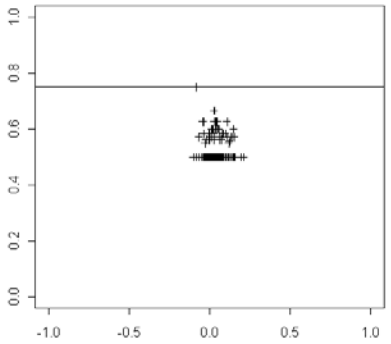
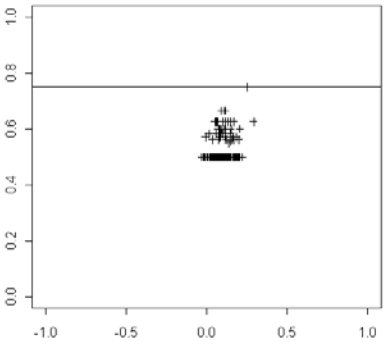
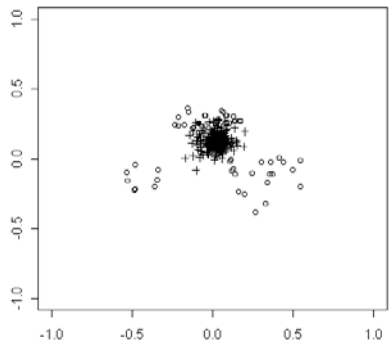
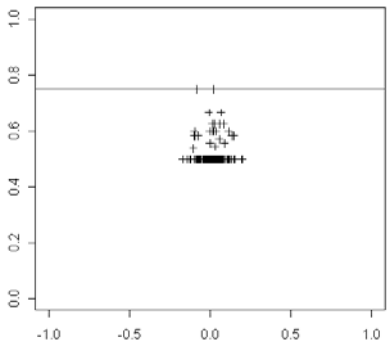
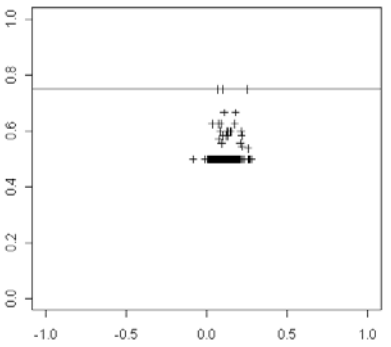
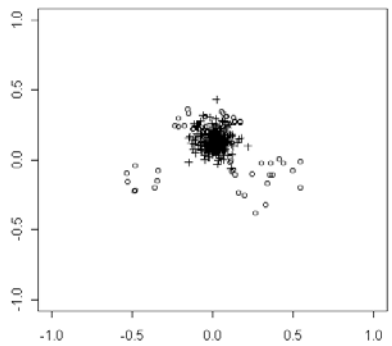
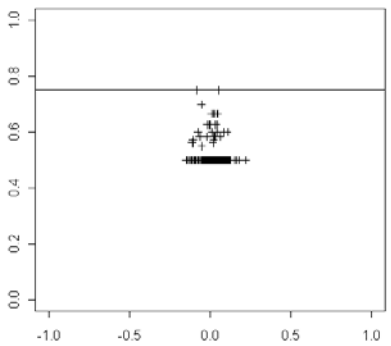
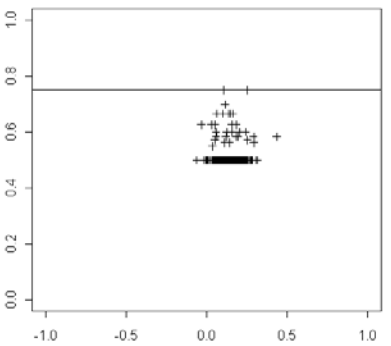
Passeio do Usuário 1168	Coesão em x	Coesão em y
α=0,1		
		
α=0,5		
		
α=1		
		

Figura 39 – “Passeio” de Usuários Selecionados (continua na página seguinte).

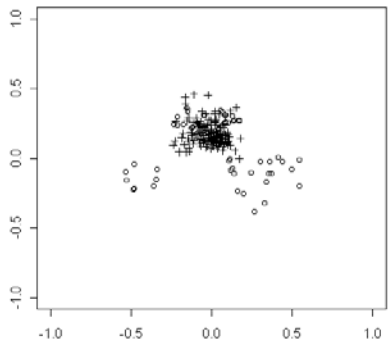
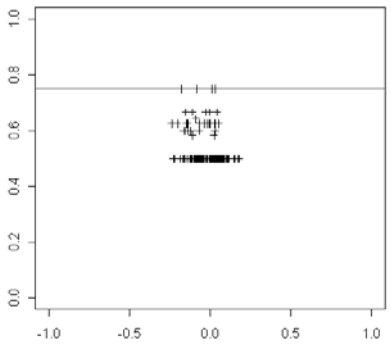
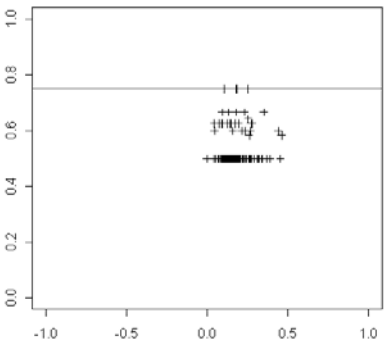
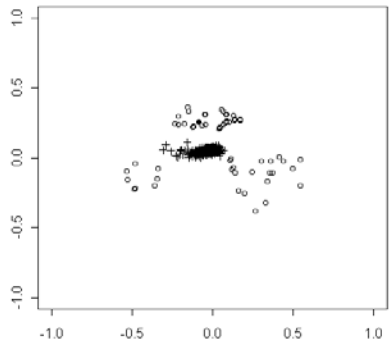
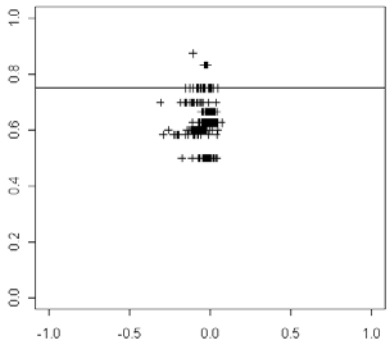
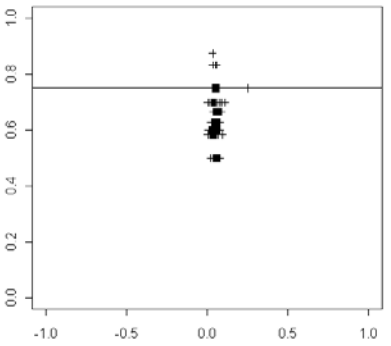
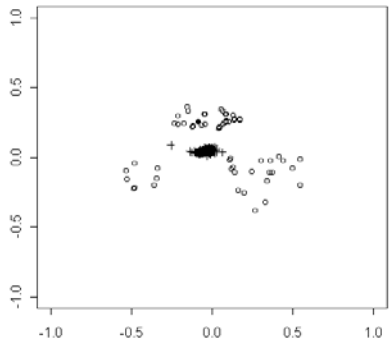
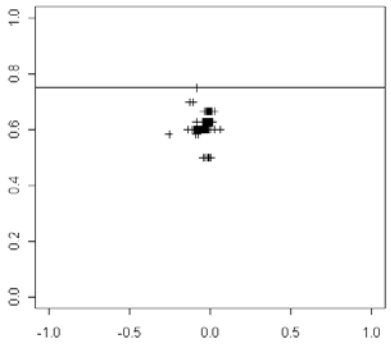
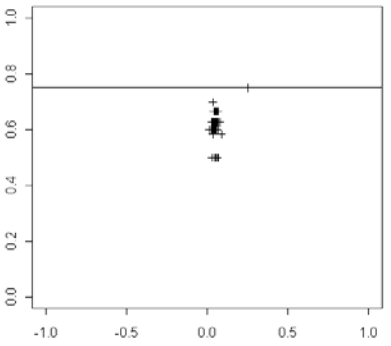
Passeio do Usuário 1168	Coesão em x	Coesão em y
α=2		
		
α=10		
		
α=20		
		

Figura 39 – “Passeio” de Usuários Selecionados (continua na página seguinte).

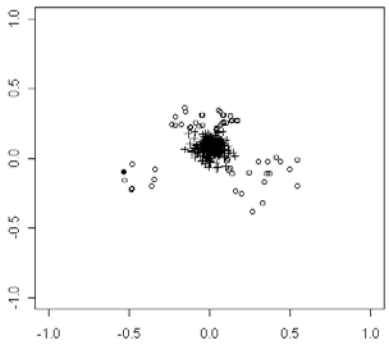
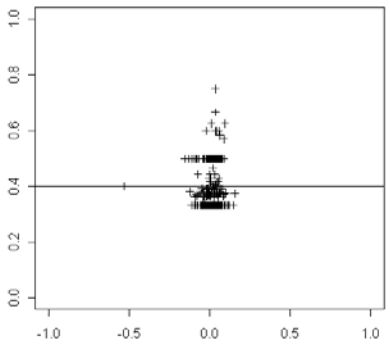
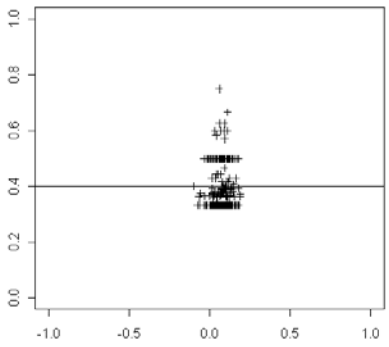
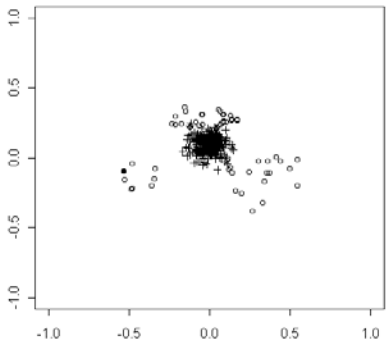
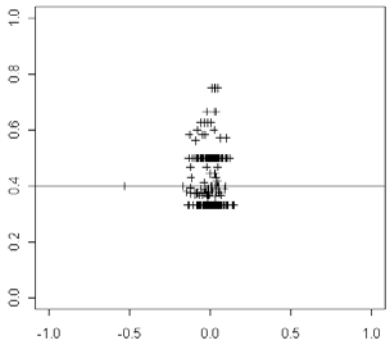
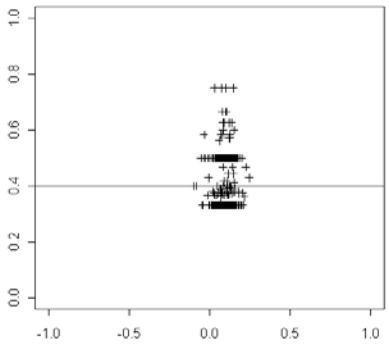
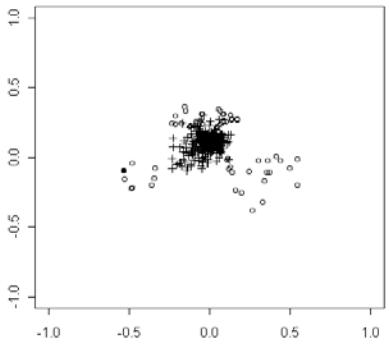
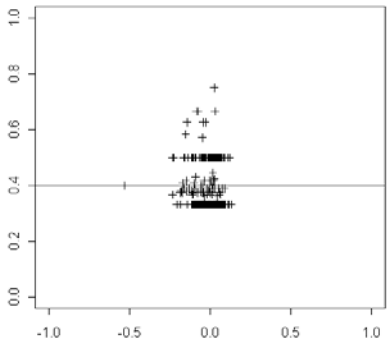
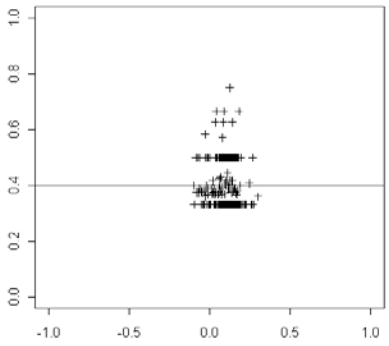
Passeio do Usuário 4020	Coesão em x	Coesão em y
α=0,1		
		
α=0,5		
		
α=1		
		

Figura 39 – “Passeio” de Usuários Selecionados (continua na página seguinte).

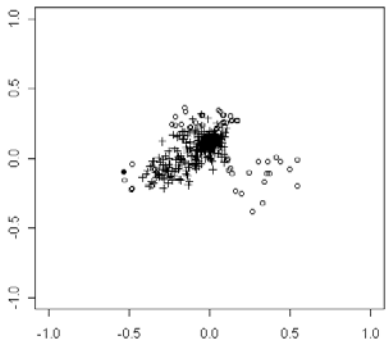
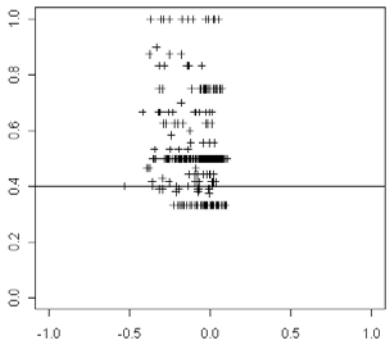
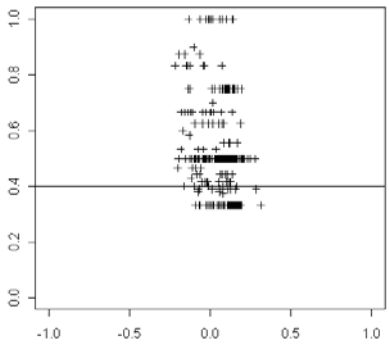
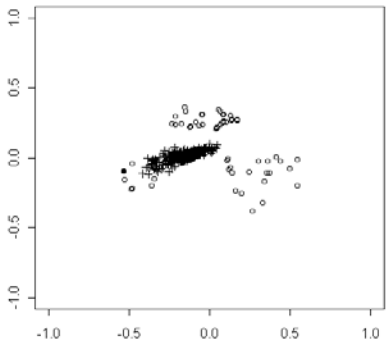
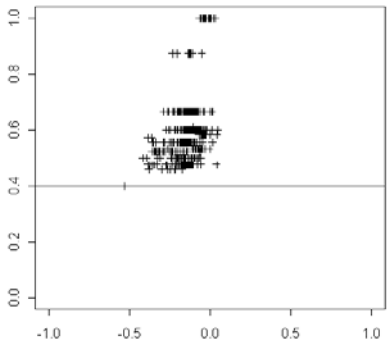
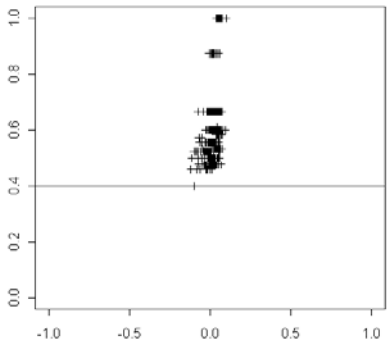
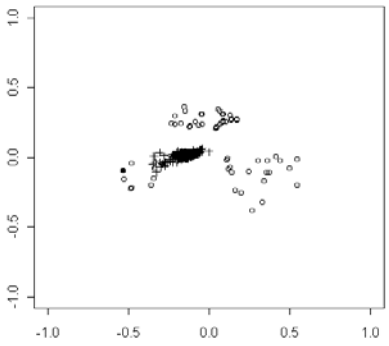
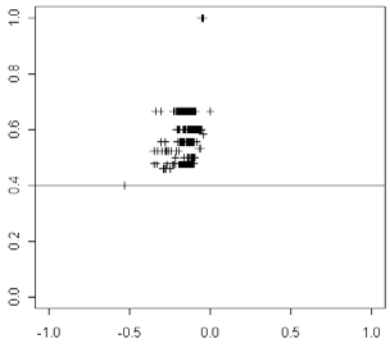
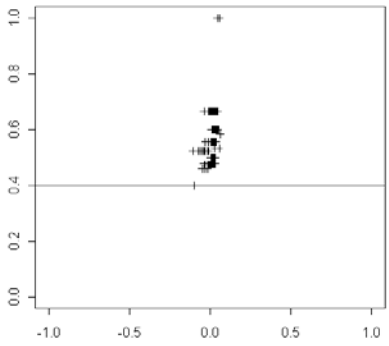
Passeio do Usuário 4020	Coesão em x	Coesão em y
α=2		
		
α=10		
		
α=20		
		

Figura 39 – “Passeio” de Usuários Selecionados (continua na página seguinte).

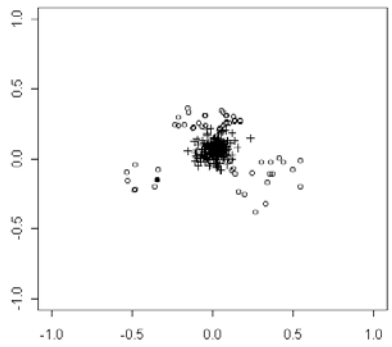
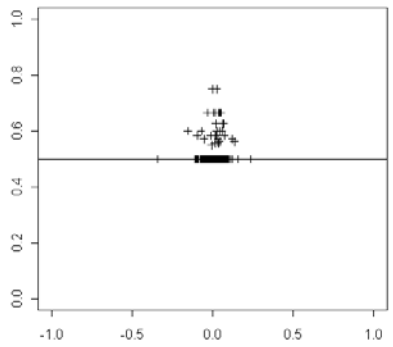
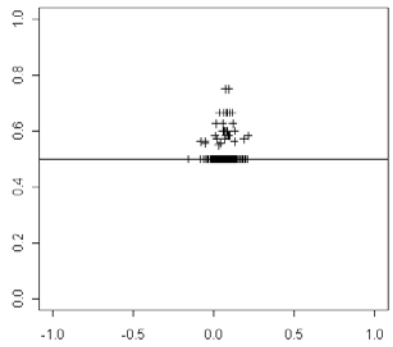
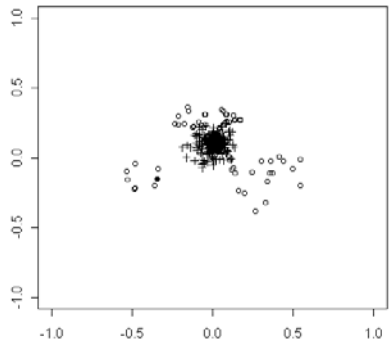
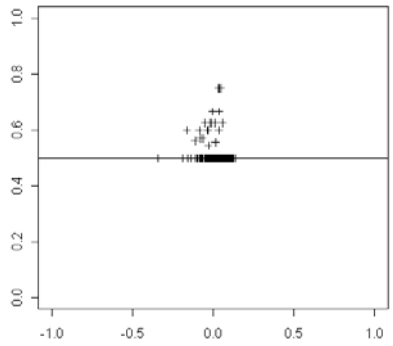
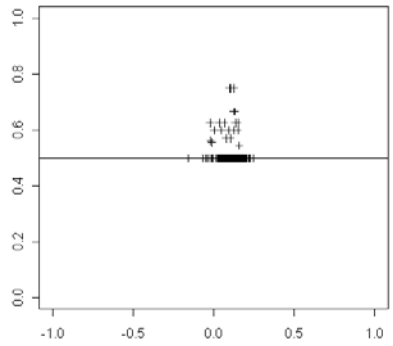
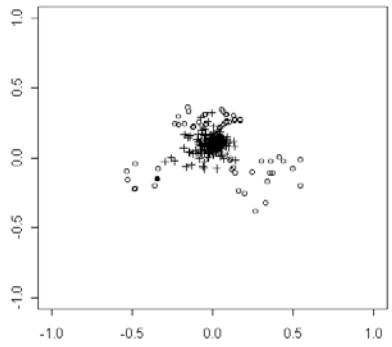
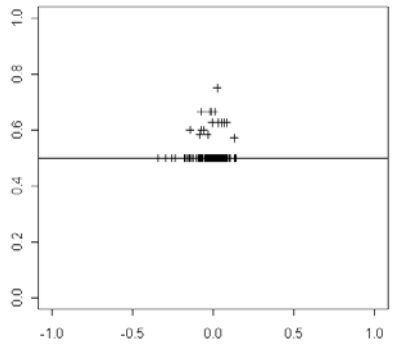
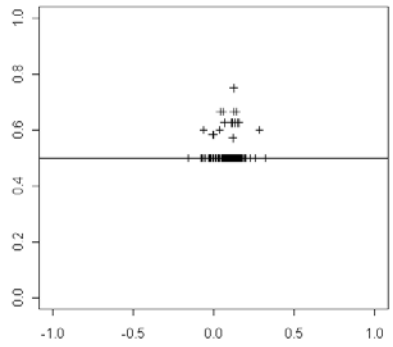
Passeio do Usuário 3154	Coesão em x	Coesão em y
α=0,1		
		
α=0,5		
		
α=1		
		

Figura 39 – “Passeio” de Usuários Selecionados (continua na página seguinte).

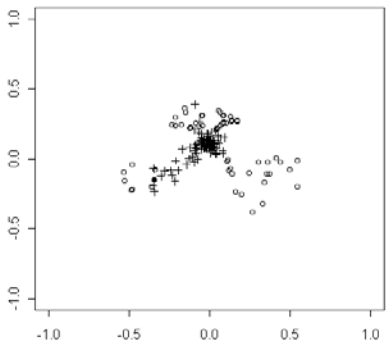
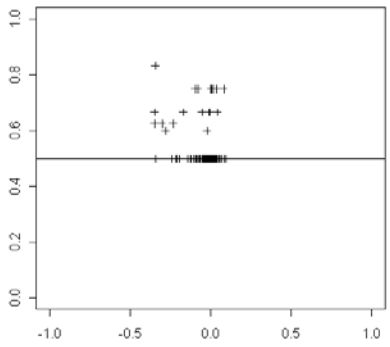
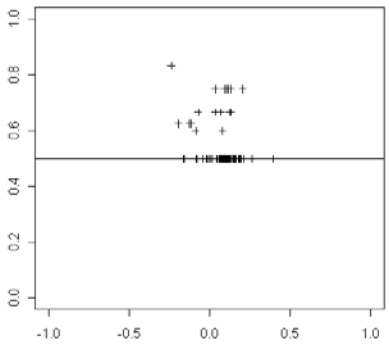
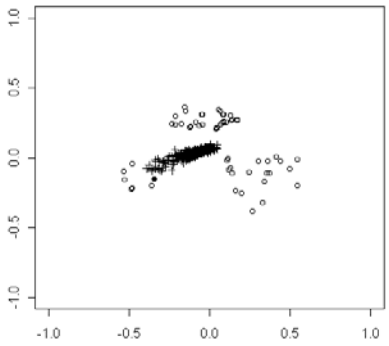
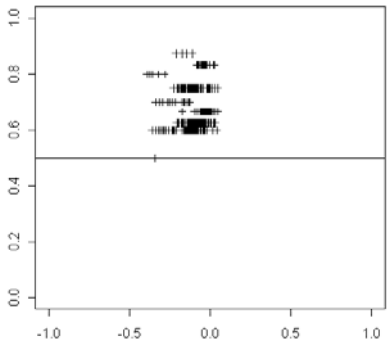
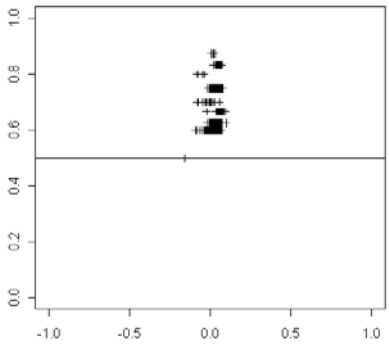
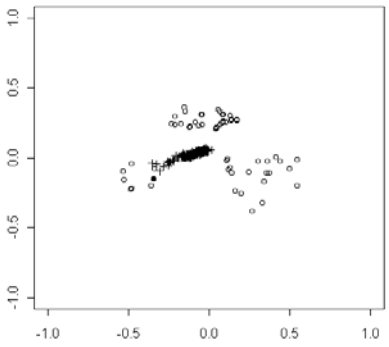
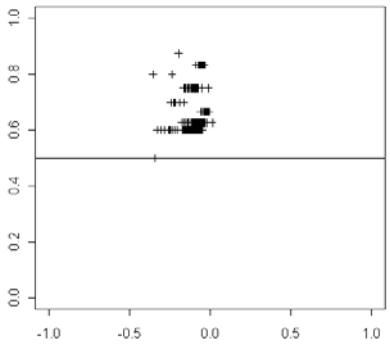
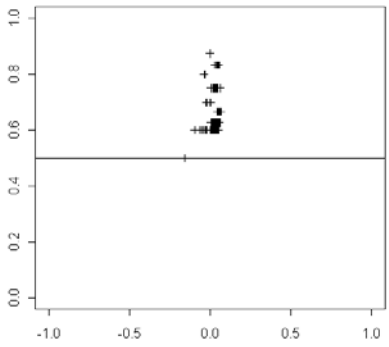
Passeio do Usuário 3154	Coesão em x	Coesão em y
α=2		
		
α=10		
		
α=20		
		

Figura 39 – “Passeio” de Usuários Selecionados (continua na página seguinte).

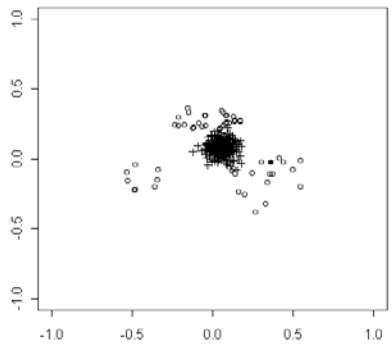
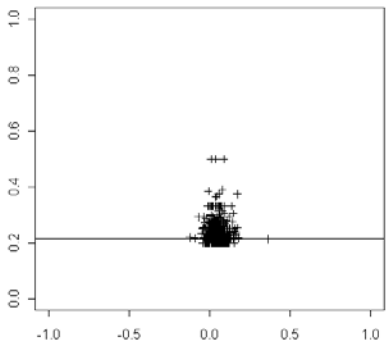
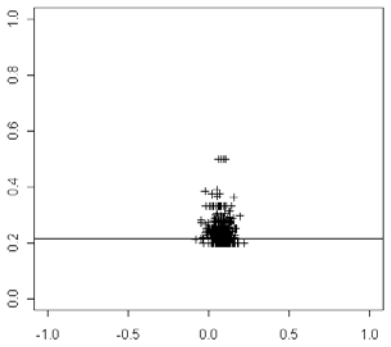
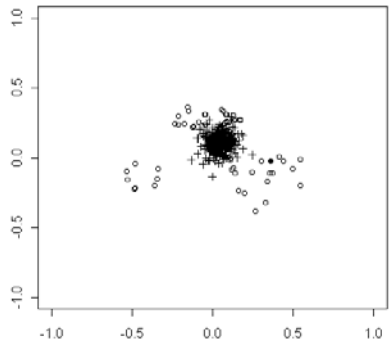
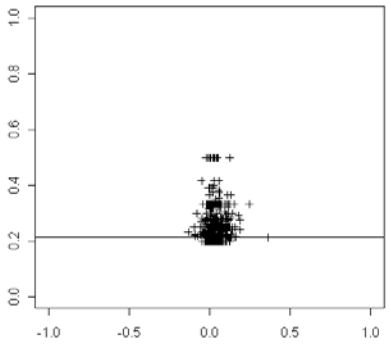
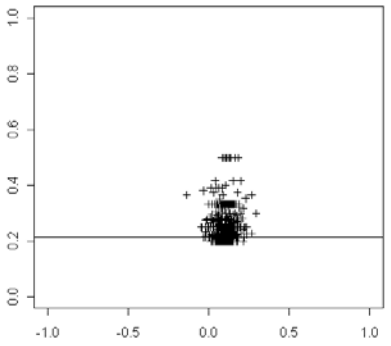
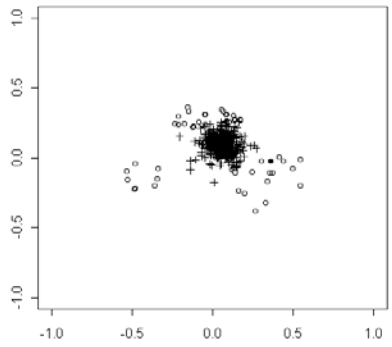
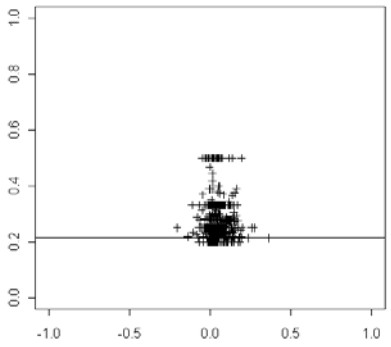
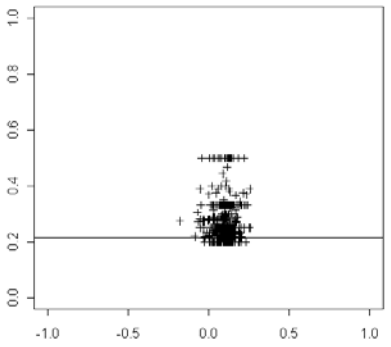
Passeio do Usuário 1189	Coesão em x	Coesão em y
α=0,1		
		
α=0,5		
		
α=1		
		

Figura 39 – “Passeio” de Usuários Selecionados (continua na página seguinte).

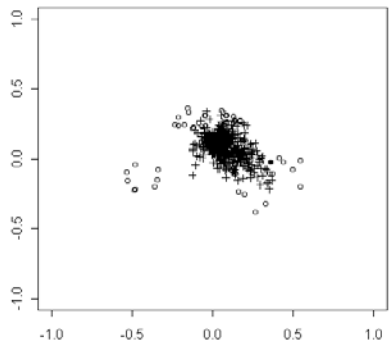
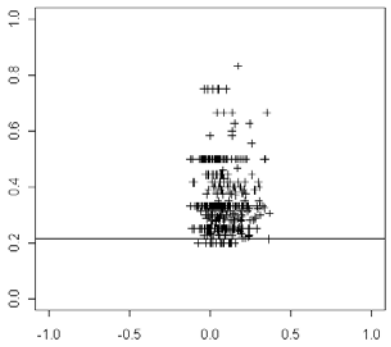
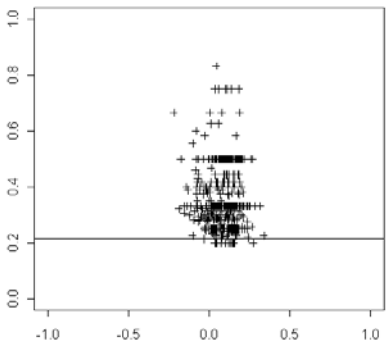
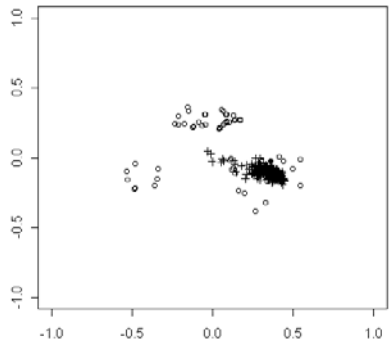
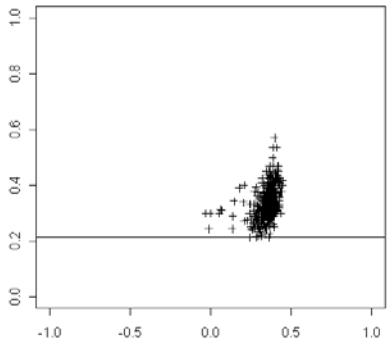
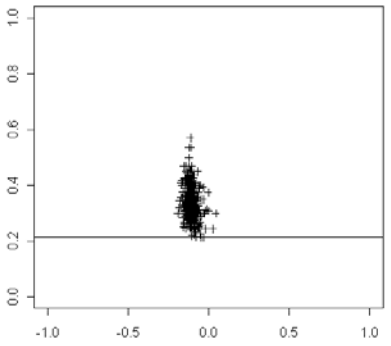
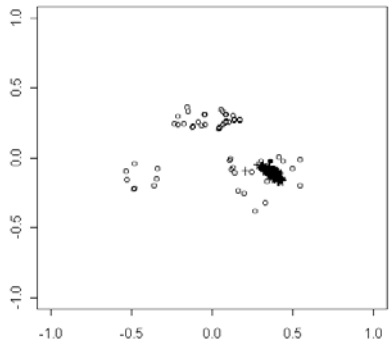
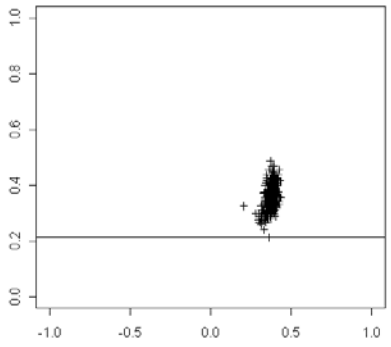
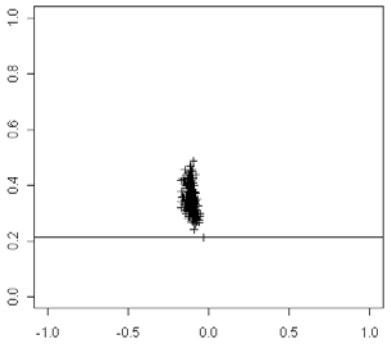
Passeio do Usuário 1189	Coesão em x	Coesão em y
α=2		
		
α=10		
		
α=20		
		

Figura 39 – “Passeio” de Usuários Selecionados (continua na página seguinte).

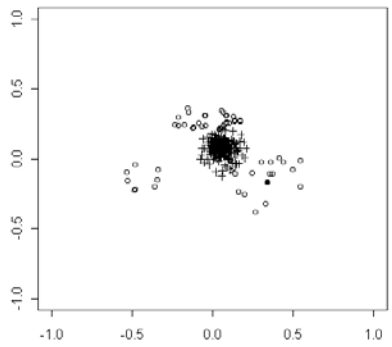
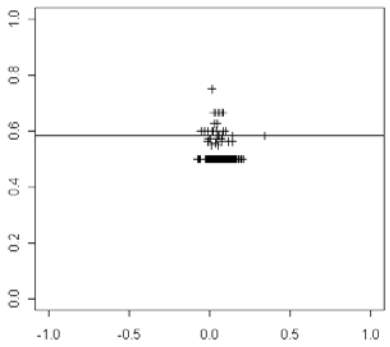
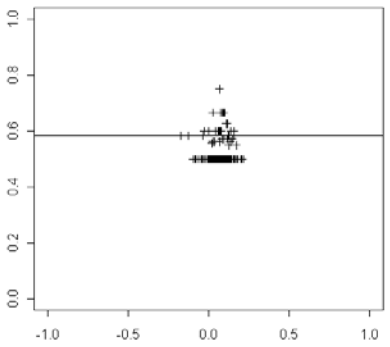
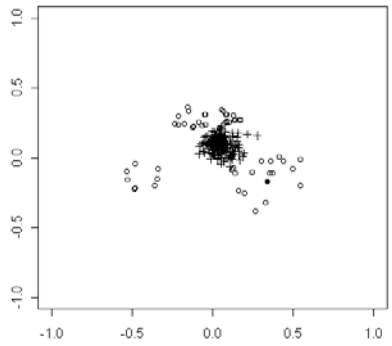
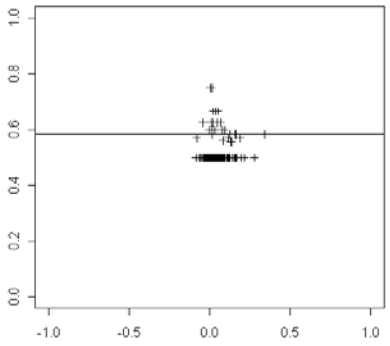
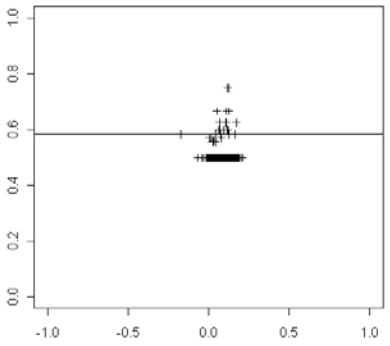
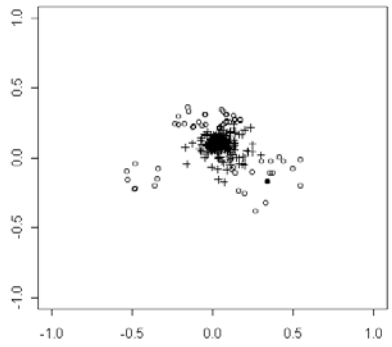
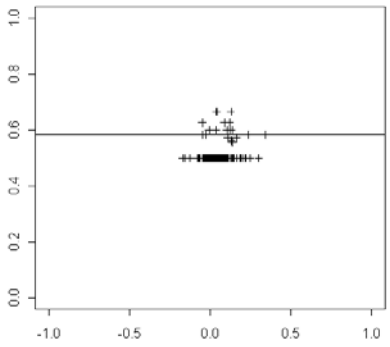
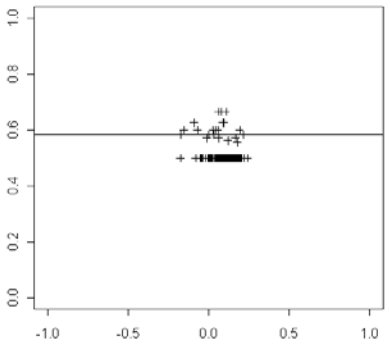
Passeio do Usuário 968	Coesão em x	Coesão em y
α=0,1		
		
α=0,5		
		
α=1		
		

Figura 39 – “Passeio” de Usuários Selecionados (continua na página seguinte).

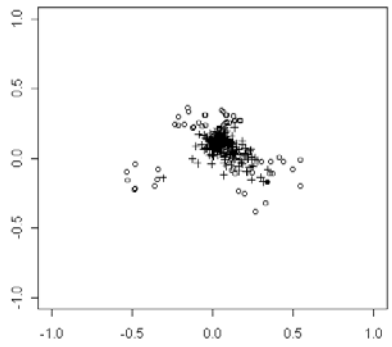
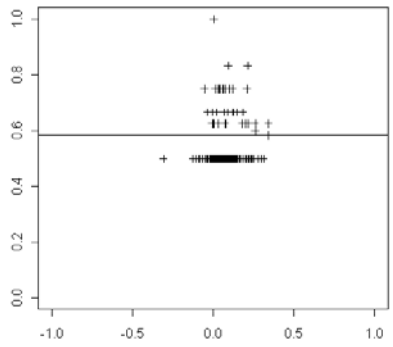
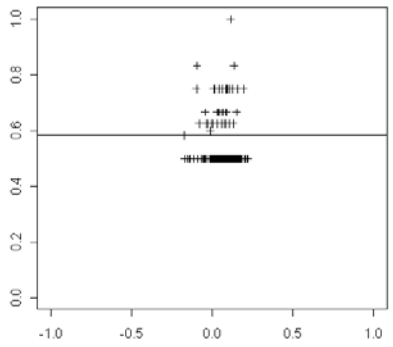
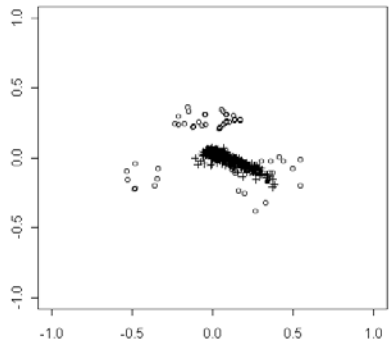
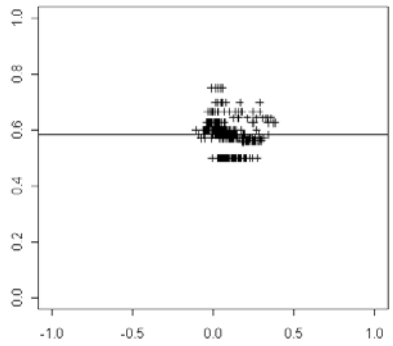
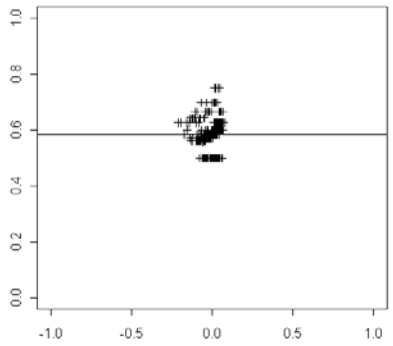
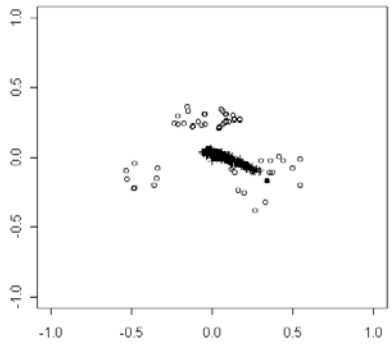
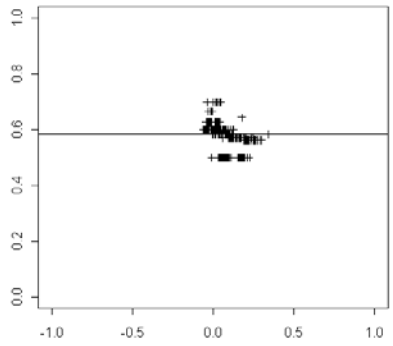
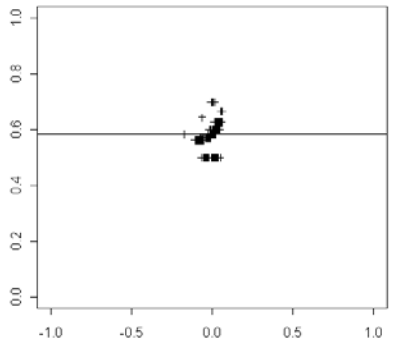
Passeio do Usuário 968	Coesão em x	Coesão em y
α=2		
		
α=10		
		
α=20		
		

Figura 39 – “Passeio” de Usuários Selecionados.

15.5.1 "Regressão" das Coordenadas do Usuário para o Centróide da Configuração

Quanto menor o α (e, portanto, quanto menor a probabilidade de amostragem das arestas e menor o tamanho das amostras) há uma tendência mais acentuada a viezar a posição "estimada" para um usuário em direção ao centróide da configuração.

15.5.2 "Regressão" da Coesão Estimada para 0,5

Quanto menor o α (e, portanto, quanto menor a probabilidade de amostragem das arestas e menor o tamanho das amostras) há uma tendência mais acentuada a viezar a coesão estimada para o usuário em direção a 0,5. Isso significa que valores altos de coesão tendem a ser subestimados, e valores baixo tendem a ser superestimados.

15.5.3 Consistência

Tanto as coordenadas quanto a coesão de um nó de usuário tendem a ser estimadas mais próximas dos valores verdadeiros quanto maior a amostra.

15.6. Relevo de Coesão das Amostras

No seção anterior, analisamos o desempenho do algoritmo com relação à representação individuais de nós da rede. Voltamos agora nossa atenção para a representação da superfície de coesão como um todo.

A discussão nesta seção está apoiada na Figura 40, que se estende pelas próximas 12 páginas, duas páginas dedicadas a cada nível de α. A primeira página do par traz uma representação tri-dimensional de 9 réplicas escolhidas aleatoriamente entre as 500 calculadas para cada nível de α; a segunda traz as mesmas representações na forma de curvas de nível, acrescidas de pontos representado os usuários integrantes da amostra.

Gráfico de perspectiva para alfa igual a 0,1.

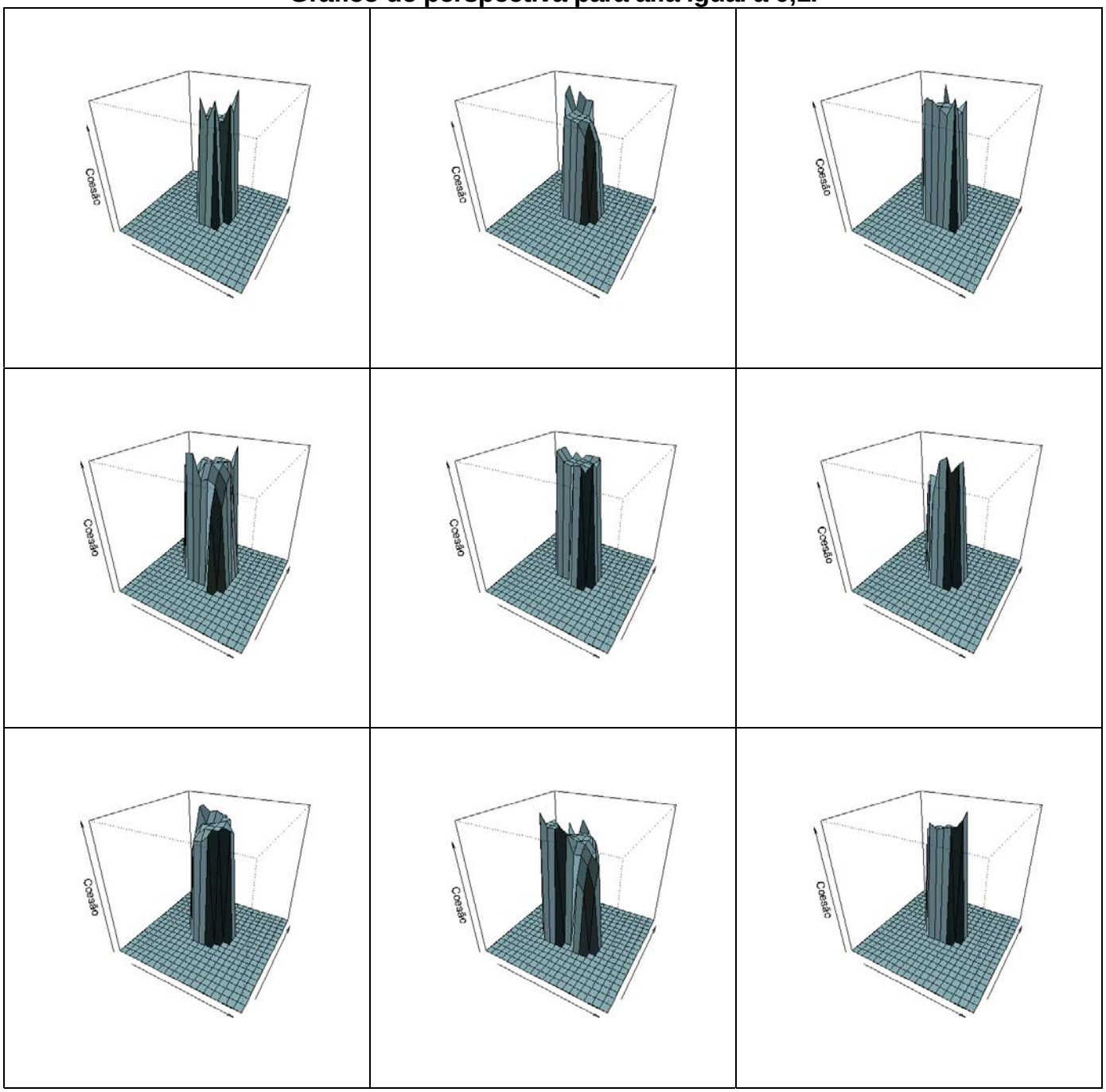


Figura 40 - Representação da Superfície de Coesão para 9 Réplicas em Cada Nível de α.
(Continua na Próxima Página)

Gráfico de contorno para alfa igual a 0,1.

Figura 40 - Representação da Superfície de Coesão para 9 Réplicas em Cada Nível de α .
(Continua na Próxima Página)

Gráfico de perspectiva para alfa igual a 0,5.

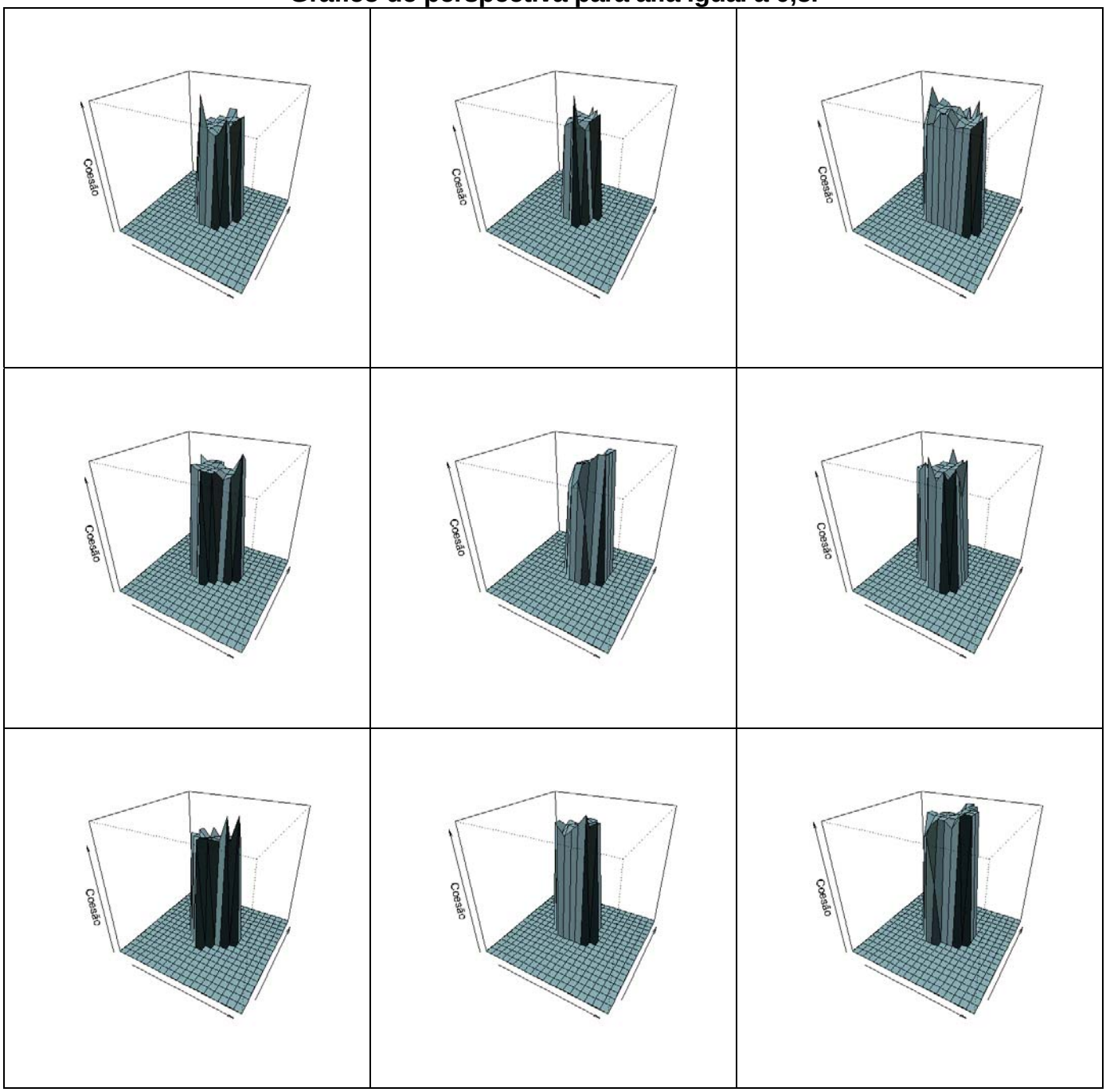


Figura 40 - Representação da Superfície de Coesão para 9 Réplicas em Cada Nível de α .
(Continua na Próxima Página)

Gráfico de contorno para alfa igual a 0,5.

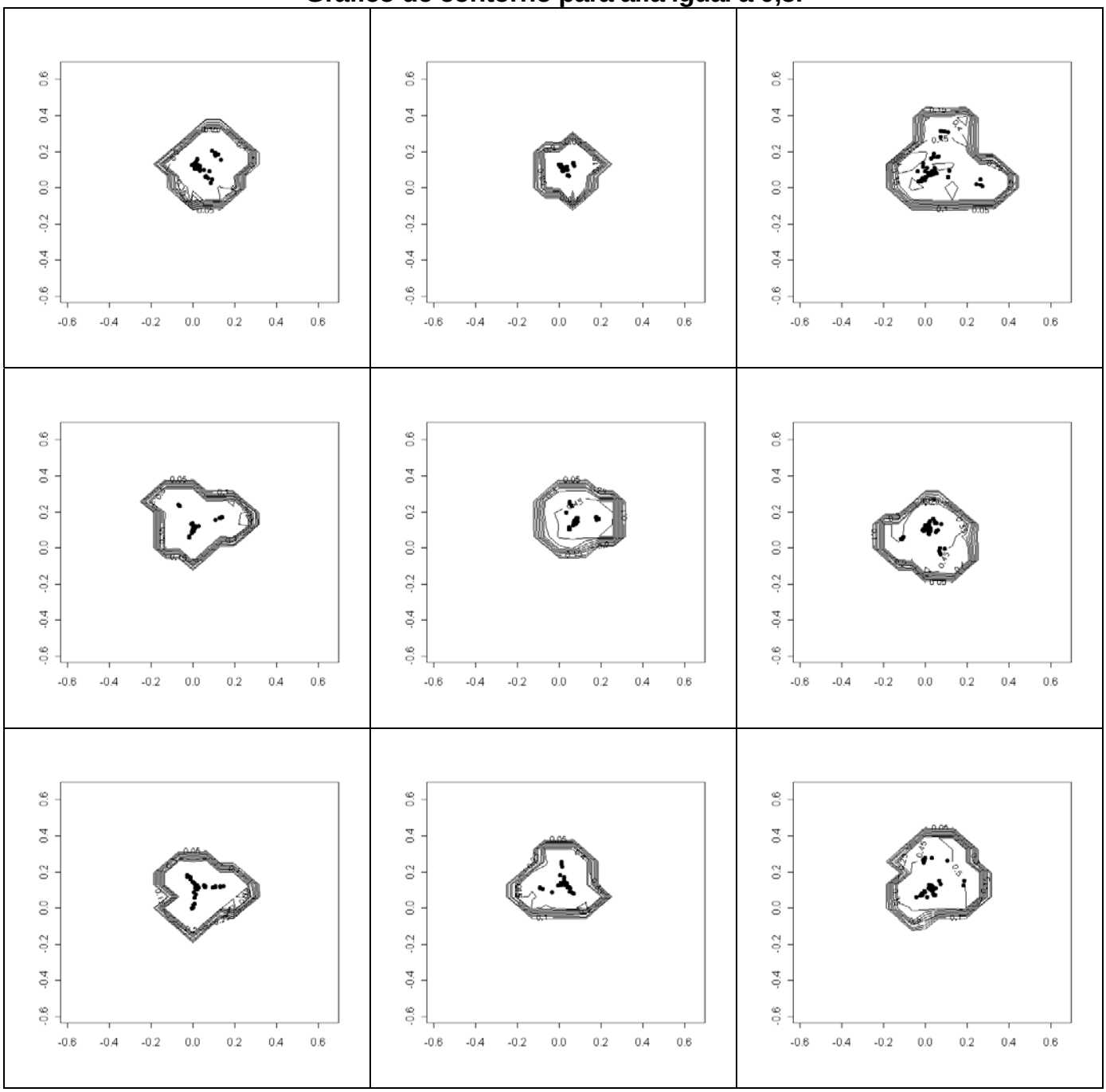


Figura 40 - Representação da Superfície de Coesão para 9 Réplicas em Cada Nível de α .
(Continua na Próxima Página)

Gráfico de perspectiva para alfa igual a 1.

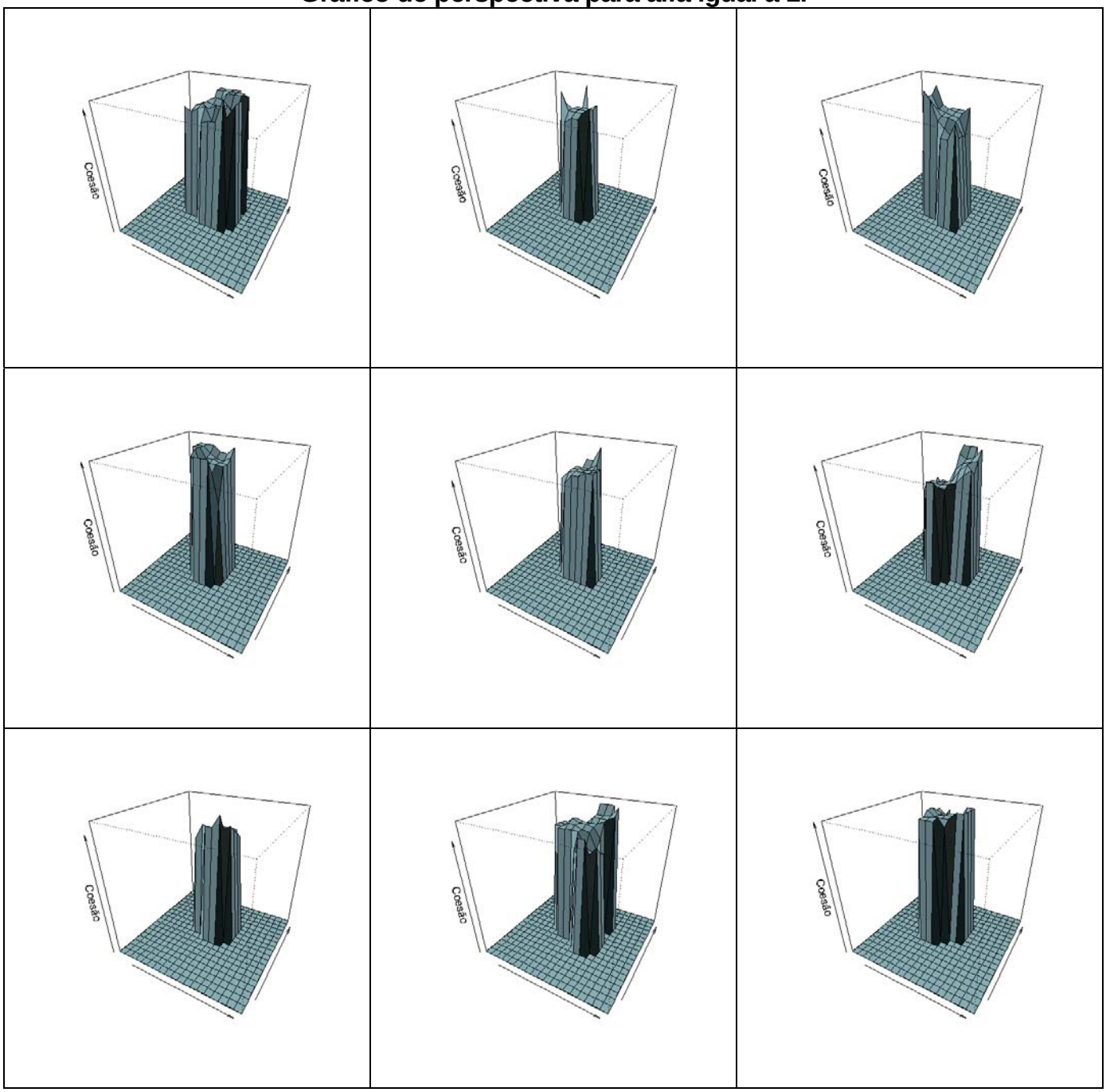


Figura 40 - Representação da Superfície de Coesão para 9 Réplicas em Cada Nível de α .
(Continua na Próxima Página)

Gráfico de contorno para alfa igual a 1.

Figura 40 - Representação da Superfície de Coesão para 9 Réplicas em Cada Nível de α .
(Continua na Próxima Página)

Gráfico de perspectiva para alfa igual a 2.

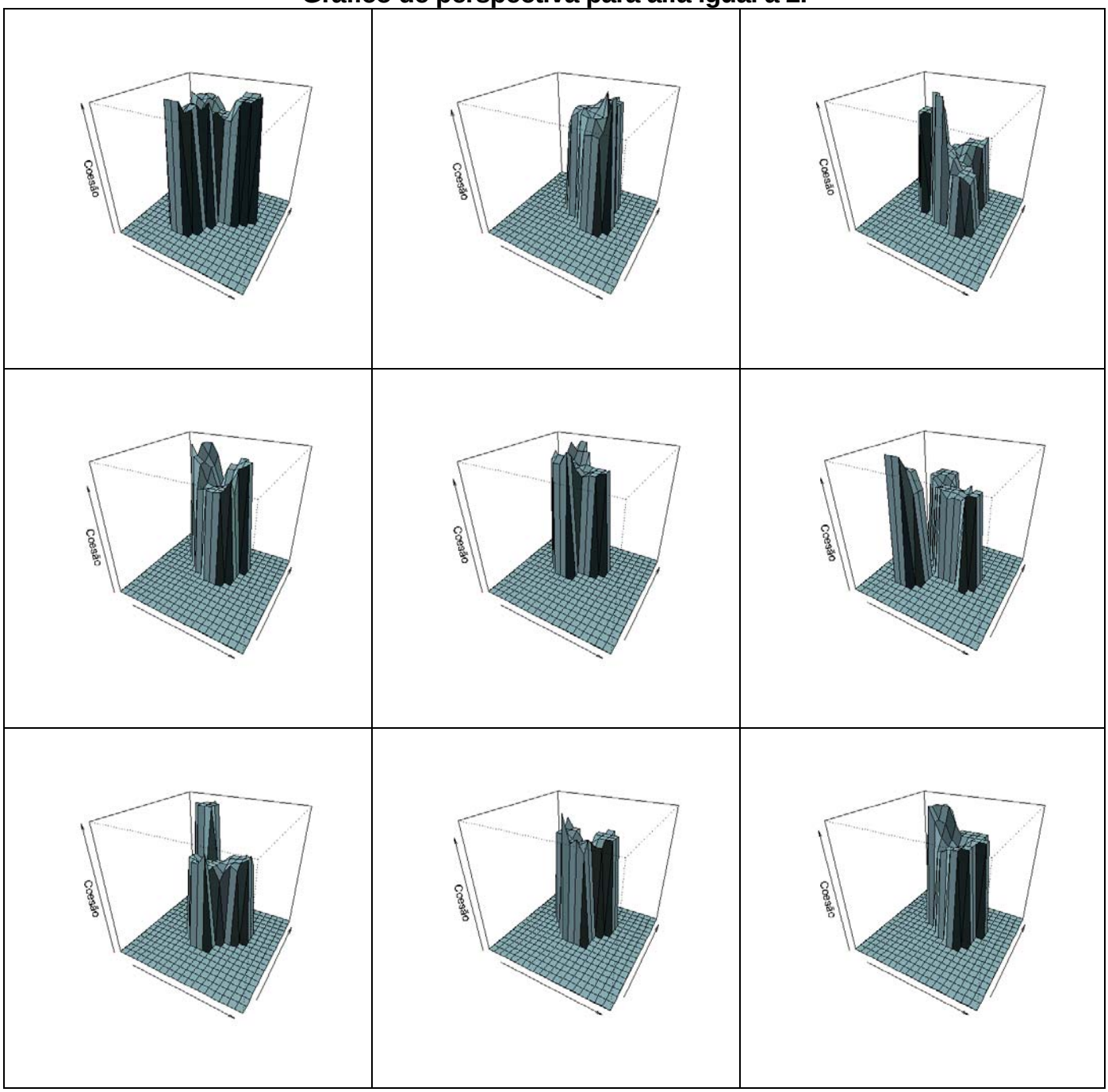


Figura 40 - Representação da Superfície de Coesão para 9 Réplicas em Cada Nível de α.
(Continua na Próxima Página)

Gráfico de contorno para alfa igual a 2.

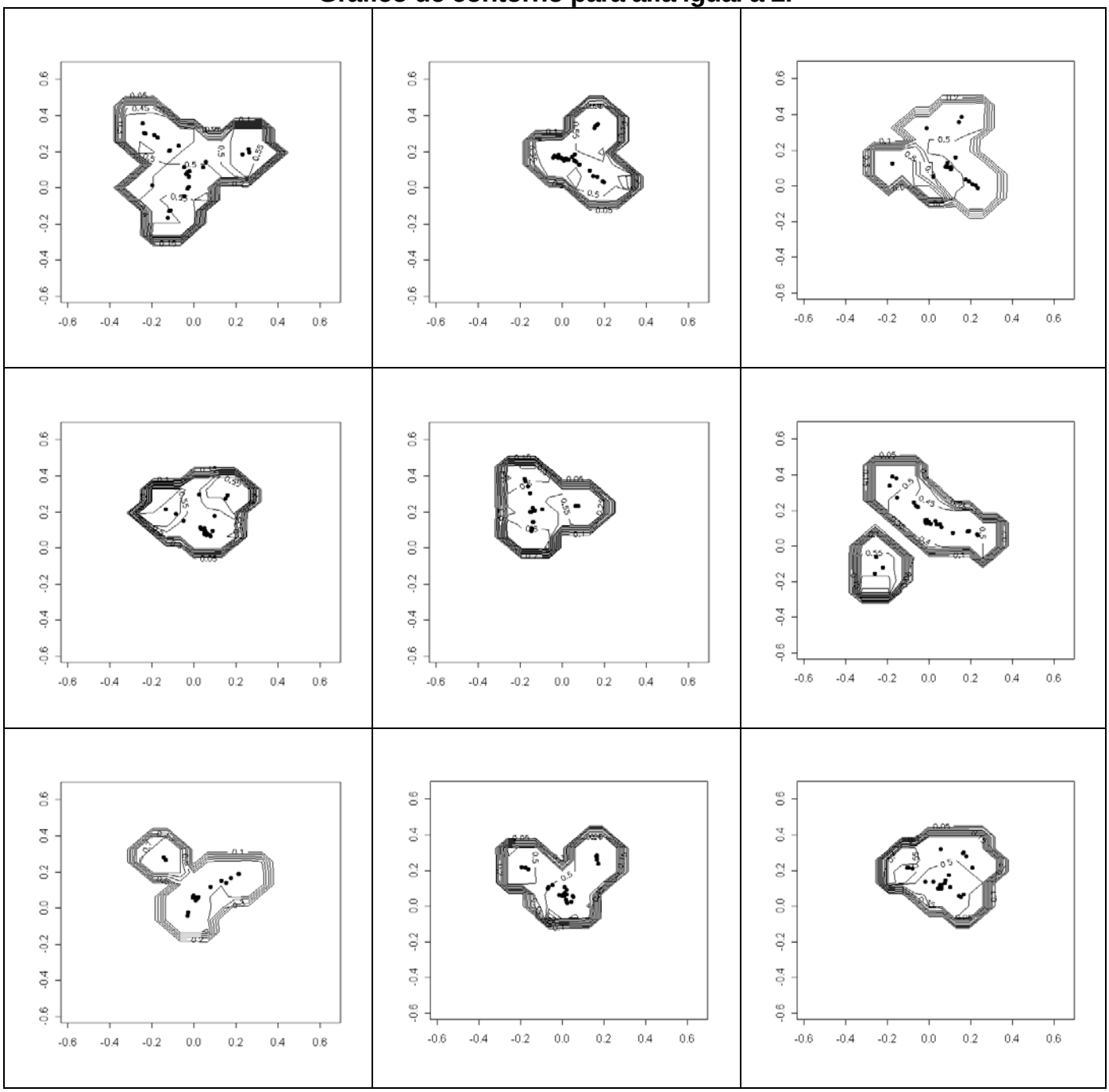


Figura 40 - Representação da Superfície de Coesão para 9 Réplicas em Cada Nível de α .
(Continua na Próxima Página)

Gráfico de perspectiva para alfa igual a 10.

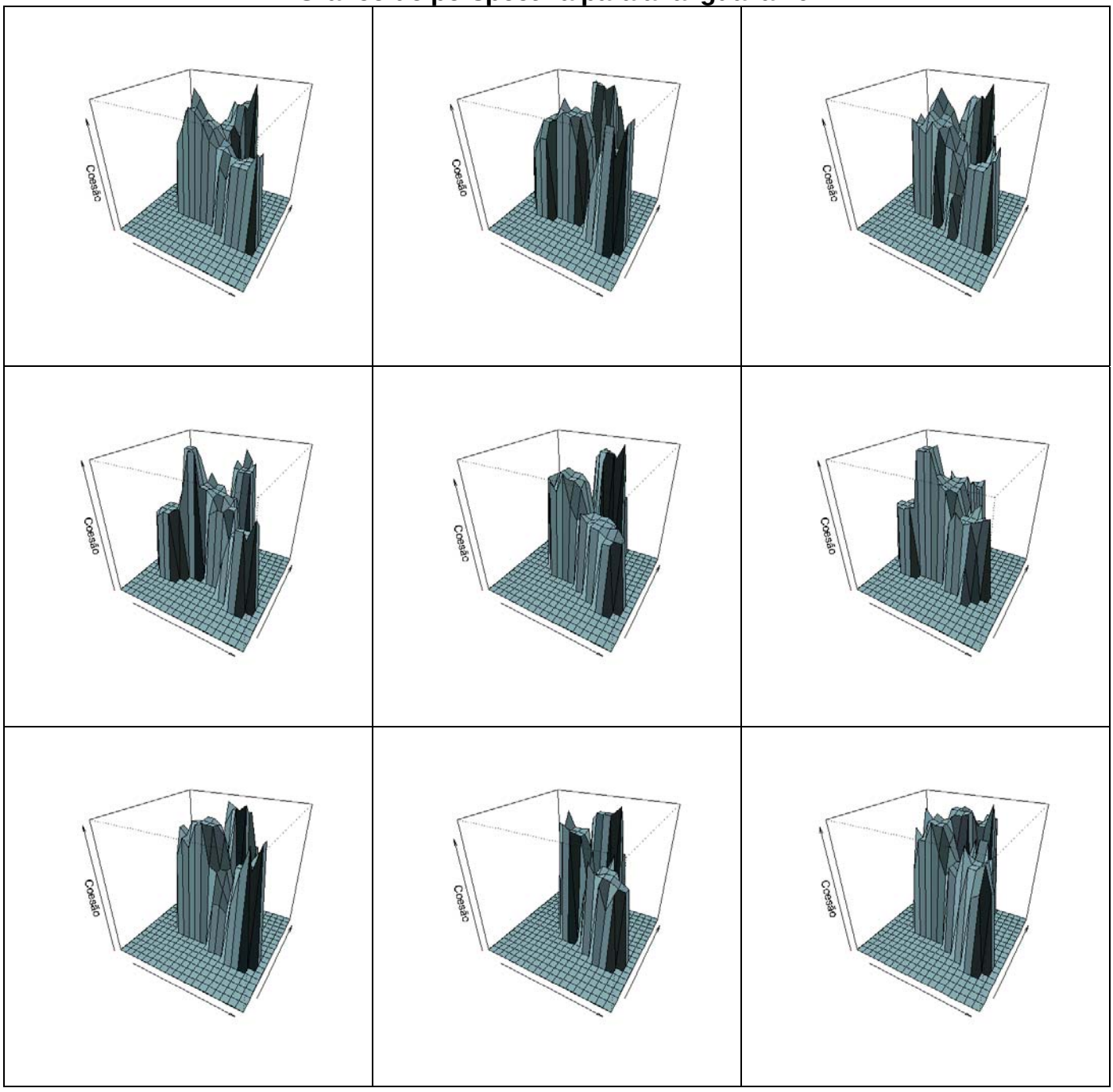


Figura 40 - Representação da Superfície de Coesão para 9 Réplicas em Cada Nível de α.
(Continua na Próxima Página)

Gráfico de contorno para alfa igual a 10.

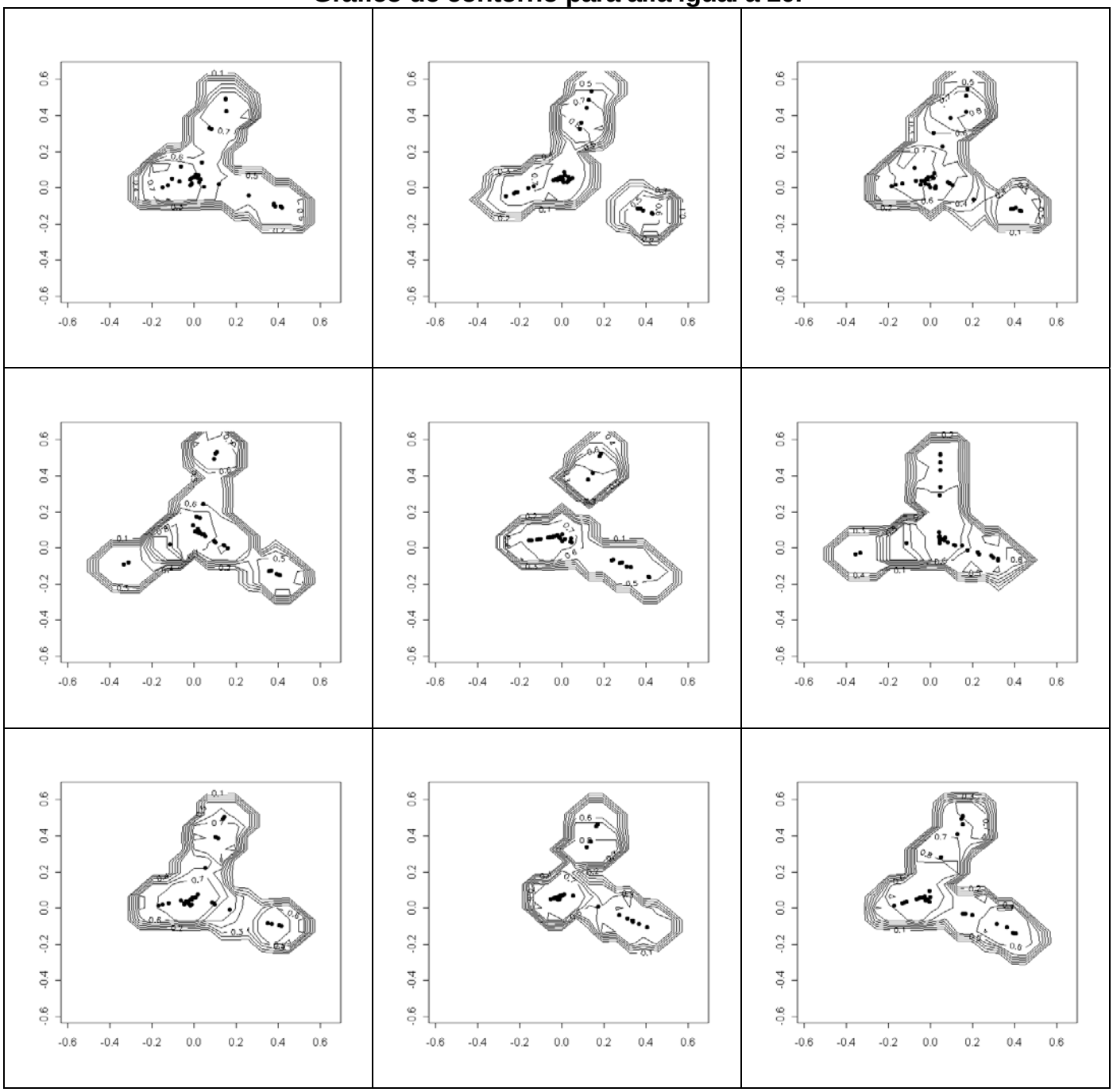


Figura 40 - Representação da Superfície de Coesão para 9 Réplicas em Cada Nível de α.
(Continua na Próxima Página)

Gráfico de perspectiva para alfa igual a 20.

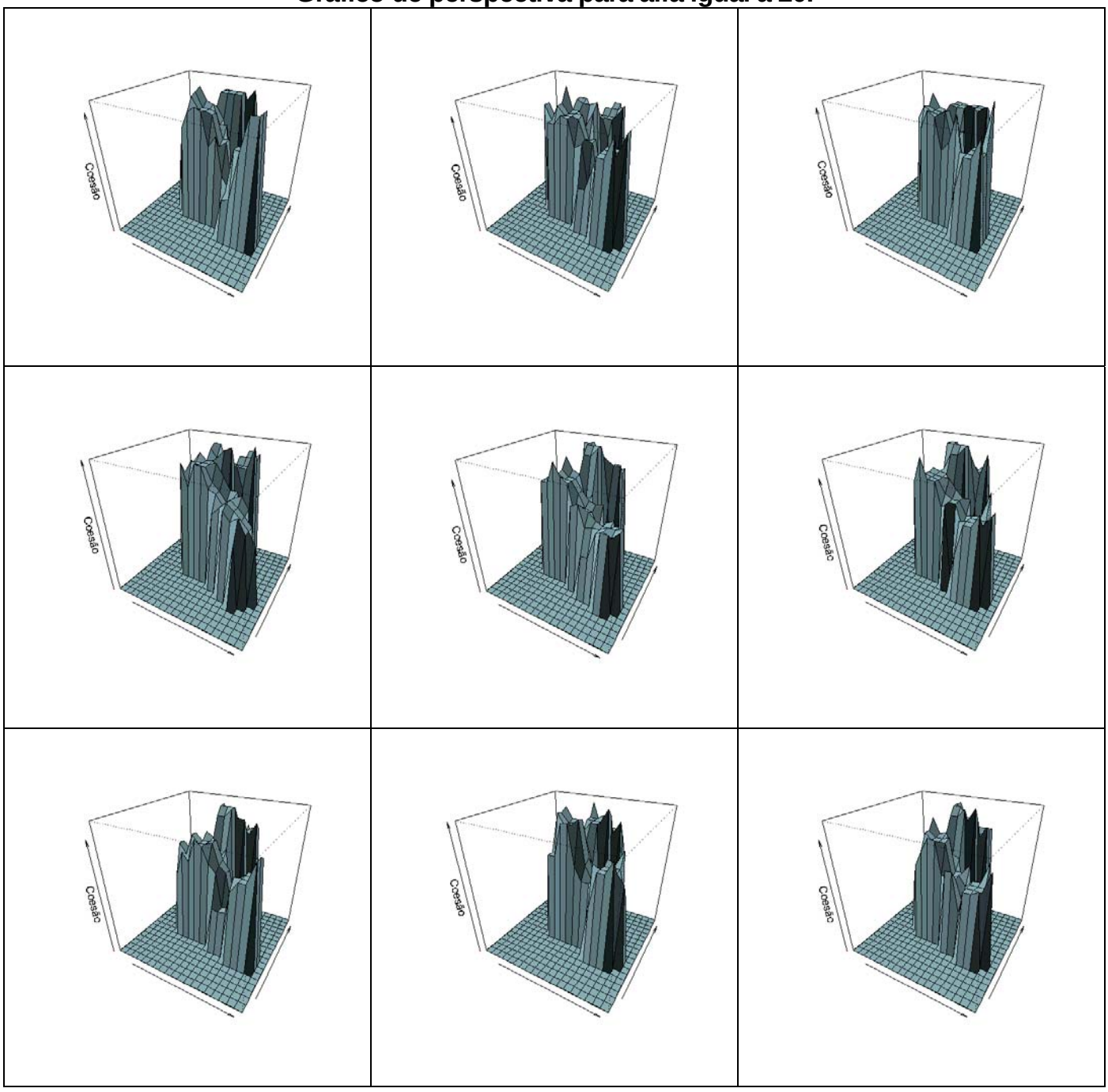


Figura 40 - Representação da Superfície de Coesão para 9 Réplicas em Cada Nível de α.
(Continua na Próxima Página)

Gráfico de contorno para alfa igual a 20.

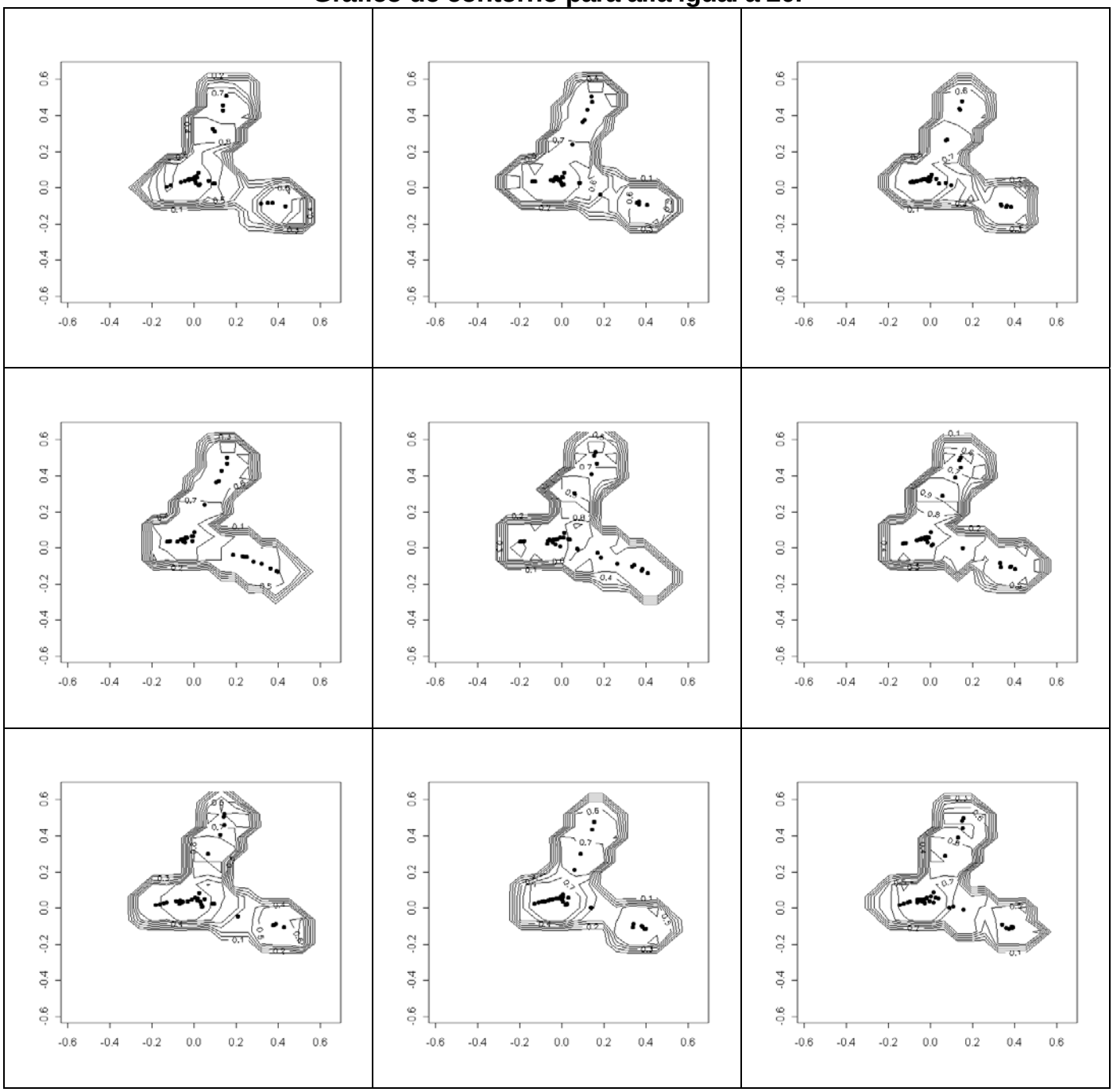


Figura 40 – Representação da Superfície de Coesão para 9 Réplicas em Cada Nível de α.

A análise comparativa e detalhada dos gráficos permite a seguinte conclusão:

15.6.1 Forma cônica ou cilíndrica para superfícies com base em amostras pequenas

Quando a amostra é pequena, a superfície de coesão tende a uma forma cônica ou cilíndrica, independentemente da forma da superfície de coesão da rede completa. À medida em que o número de arestas amostradas aumenta, a superfície de coesão calculada para a amostra torna-se mais parecida com a superfície de coesão da rede completa.

Em retrospectiva, isto é, após a realização do estudo de simulação, esta constatação pode parecer óbvia.Considerando-se que o núcleo firme de uma rede com poucas arestas deve ser anelar, e considerando-se a "flexibilidade geométrica" de uma rede anelar pequena, bem como as características de otimização do MDS clássico, este resultado de uma superfície cônica ou cilíndrica parece ser o esperado. Á medida que a rede cresce, e o número de arestas amostrado aumenta, a rede perde a flexibilidade e começa a conservar sua forma idiossincrática.

15.7. Síntese do Seção

Nesta seção, estudamos o efeito da amostragem sobre a representação construída com a aplicação do algoritmo RGR. Postula-se que, para ser útil, o algoritmo não poderia produzir representações completamente distintas de uma amostra para outra, sob pena de o analista passar a interpretar um mero "borrão" aleatório.

Adotamos o conjunto de dados do Exemplo (seção 8) como população e fizemos 500 réplicas para cada um de 6 níveis pré-determinados de α, parâmetro que regula as probabilidades de inclusão de arestas em amostras.

Observou-se que em amostras pequenas surge um viés nas coordenadas dos nós em direção ao centróide da configuração; e um viés da medida de coesão do nó em direção ao valor 0,5; e que a superfície de coesão da rede tende a uma simplificação cônica ou cilíndrica. Por outro lado, verificou-se que, à medida que a amostra aumenta de tamanho, o viés diminui e a forma da superfície de coesão se amolda ao formato "real".

Estes resultados dão segurança ao uso do algoritmo para grandes redes, caso para o qual o algoritmo RGR foi idealizado. A instabilidade da representação de redes grandes com base no algoritmo aplicado a amostras pequenas é uma realidade; no entanto, nunca houve a intenção de resolver o problema para amostras pequenas, que podem ser tratadas por Escalonamento Clássico.

Desta forma, concluímos que o algoritmo atende seus objetivos propostos. A redução do R^2 do Procrustes relatada na subseção 15.4 tampouco pode ser tomada como evidência desfavorável. O R^2 só é elevado para representações de amostras pequenas porque justamente estas são inespecíficas e deformadas em relação ao relevo real da rede.

Quinta Parte: Fechamento

16. CONCLUSÃO

16.1. Data Mining: Problemas e Soluções na Representação de Redes

Data Mining é um campo de estudo mal definido, uma combinação de Estatística, Tecnologia da Informação (principalmente programação e gestão de bancos de dados), Inteligência Artificial, técnicas de visualização e conhecimento sobre o campo semântico da aplicação.

No caso da aplicação de Filtro Colaborativo que motivou este trabalho, a técnica empregada foi a de Análise de Redes. A característica de mineração de dados decorreu do grande volume de transações a processar. Para fazê-lo, há duas abordagens complementares: a representação gráfica e avaliação matemática da rede.

Desenhar "bem" uma rede permite entender-lhe a estrutura (SKIENA, 1998). A representação gráfica permite entender o grande quadro.

A representação matemática permite calcular medidas de interesse e é suporte para a representação gráfica. Ambas permitem identificar padrões.

Para ambas as estratégias há soluções bem consolidadas para redes pequenas. Também para ambas, os procedimentos consolidados tornam-se disfuncionais à medida que aumenta a ordem e o tamanho da rede.

Neste trabalho, conceitos de matemática, estatística, computação, marketing e geomarketing foram utilizados para a produção de uma solução ao problema colocado, de representar uma rede de usuários da biblioteca como uma superfície de coesão sobre uma base tematicamente escalonada.

Os estudos para validação da representação gerada pelo algoritmo RGR produziram elementos para considerá-la consistente; o termo "consistente" é aqui empregado no sentido de que, quanto maior a amostra, a representação obtida com dados incompletos da rede tornam-se mais parecidos com a representação da rede integral.

Constatou-se também que amostras pequenas sofrem de viés: as coordenadas de usuários deslocam-s em direção ao centróide da configuração; e o valor da coesão devia-se em direção ao valor 0,5. O formato da superfície de coesão para amostras pequenas tende a um cone ou cilindro, independentemente do formato da superfície da rede completa. À medida que a amostra cresce, no entanto, a superfície amostrada torna-se mais parecida com a superfície que representa a população.

Como o algoritmo foi desenvolvido para a representação de grandes redes, estas propriedades assintóticas permitem considerar a interpretação do resultado do algoritmo como válida.

17. CONTRIBUIÇÕES DO TRABALHO

Foram as seguintes as principais contribuições do trabalho:

- proposição dos conceitos de base escalonada (seção 10) e de superfície (contínua) de coesão (seção 11) para a representação de redes;
- implementação de um algoritmo para o cálculo de Distâncias Temáticas (subseções 5.1 e 20.1.7);
- combinação das técnicas de escalonamento clássico e procrustes (seções 6 e 7, particularmente subseção 7.2.1; e subseção 20.1.18);
- definição da estratégia e implementação de um algoritmo para o escalonamento multidimensional de vértices de uma rede de ordem elevada (seção 13);
- utilização de uma abordagem multidisciplinar para o encaminhamento da solução, com ampla utilização de características próprias da aplicação (seção 12);
- verificação da interpretabilidade da solução proposta e comparação com alternativas de Escalonamento Clássico (seção 14).
- verificação de estabilidade da solução em termos de variação amostral (seção 15)

18. LIMITAÇÕES

O trabalho apresentado também apresenta várias limitações.

18.1. Triangulação

Por um erro de lógica da função triângulo, pode acontecer, e freqüentemente acontece, que vértices sejam desprezados na triangulação; neste caso, eles resultam sem coordenadas ao final da iteração no nível em processamento e não podem ser incluídos na iteração de descida para os níveis seguintes. Este *bug* está profundamente enraizado na programação do algoritmo e sua correção implica alterações generalizadas no código, o que era inviável para o cronograma deste trabalho.

Como solução provisória, e tendo em vista que a origem da necessidade de desenvolvimento do algoritmo é o **excesso** de dado, optamos simplesmente por remover os vértices perdidos durante o processamento, e todos os seus filhos.

No Exemplo 2 o problema ocorre. O vértice perdido no Nível 1 é o 5922 (veja Figura 20). Por propagação, 5 pontos (dentre os 71) da configuração final também são perdidos.

A solução final, embora incompleta, não fica qualitativamente prejudicada. A falha será corrigida nas próximas versões do programa.

18.2. Informática, Interface

O trabalho apresentado tem por objetivo desenvolver conceitos. Embora acabe profundamente envolvido com a programação da solução para prova dos conceitos, não é um trabalho em Ciência da Computação: a programação realizada não tem pretensão de ser eficiente ou elegante, mas objetiva apenas colocar os conceitos em prática.

Tampouco foi desenvolvida uma interface amigável, como é necessário em aplicações de *Data Mining.*

18.3. Técnica Descritiva

A técnica apresentada é descritiva.

18.4. Isolamento da Base de Dados

A solução apresentada não oferece alternativas para integrá-la de maneira mais orgânica à base de dados originais. Em conseqüência, a interpretação dos resultados é trabalhosa e artesanal.

18.5. Desconexão de Interface Gráfica

Tampouco foi desenvolvida uma maior integração do algoritmo com uma interface gráfica. Também a conexão dos resultados matemáticos com a representação gráfica foi feita de forma manual e artesanal.

18.6. Função Geradora de Probabilidade Sensível ao Grau do Nó

A função geradora de probabilidades para a amostragem de arestas incidentes sobre um nó de usuário utilizada no estudo da estabilidade das soluções do algoritmo RGR (veja seção 15.3) resultou sensível ao número de livros retirados pelo usuário, de maneira indesejável. Numa próxima versão seria interessante substituir a função.

19. PRÓXIMOS PASSOS

Todas as limitações acima podem ser entendidas como oportunidades para novos trabalhos e aperfeiçoamentos. Destacamos alguns a serem implementados na seqüência.

19.1. Aperfeiçoamentos Planejados

19.1.1 Triangulação – Remoção de *Bug* Conhecido e Aproveitamento de Redundância

A função de triangulação precisa ser revista para remoção do *bug* identificado; no momento ele está sendo apenas gerenciado. Nesta oportunidade talvez seja conveniente, em vez de corrigir o algoritmo, incorporar outro algoritmo de triangulação existente, mais rápido e robusto.

Está também em aberto o desenvolvimento de uma estratégia para o aproveitamento da redundância dos algoritmos de triangulação convencionais para obtenção iterativa de um escalonamento melhor ajustado.

19.1.2 Iteração no Cálculo da Distância Temática

Foi identificada a conveniência da inclusão dos vizinhos de segunda ordem e de ordem superior no cálculo da Distância Temática. Isto resultará em uma matriz *dp0p0* mais populada, e, portanto, em mais informação.

19.1.3 Aperfeiçoamento do Ajuste por Procrustes

O desenvolvimento de uma análise de procrustes não linear pode ser útil no aperfeiçoamento da junção das soluções parciais.

19.1.4 Ajuste Fino: Uma Janela Voadora

Uma etapa de "iteração geográfica" no algoritmo de descida pode servir para uma melhoria no resultado do ajuste entre as soluções parciais.

19.1.5 Stress Deve ser Indiferente a Distorções de $\delta_{ij} = 1$ para $d_{ij} > 1$

A implementação de um escalonamento por mínimos quadrados, com uma função de *stress* modificada para distâncias (calculadas com base na configuração) maiores do que 1, na fase de descida, pode levar a um melhor ajuste entre as soluções parciais.

19.2. Idéia Novas para Interpretação dos Resultados

Para a caracterização da solução proposta como uma solução de *Data Mining*, é necessário agregar mais "inteligência" ligada ao campo semântico da aplicação original (geração de recomendações para usuários da biblioteca). A seguir apresentamos algumas sugestões.

19.2.1 Delimitação Automática do Relevo

Automatização da criação de grupos de usuários com interesse temático afim; geração das listas de recomendação; caracterização dos grupos na representação gráfica.

19.2.2 Medidas de Interesse

Geração automatizada de relatórios sobre as características do grupo: tamanho, atividade, coesão, liderança, etc.

19.2.3 Identificação de "Conectores"

No caso dos livros, identificar os itens do acervo com características de conectores, na definição de GLADWELL (2000).

20. ANEXO: LISTAGEM DA IMPLEMENTAÇÃO EM R

20.1. Funções do Algoritmo RGR (Representação de Grandes Redes)

Apresentamos, a seguir, o código das funções implementadas para a construção do algoritmo.

20.1.1 Deduplique

Entrada: Matriz original de dados, com duas colunas contendo códigos de usuários e livros (**TO**).
Saída: Matriz da entrada, sem linhas duplicadas (**TD**).

```
f.deduplique<- function(m)

{
#Verifica se a função recebeu uma matriz na dimensão correta.
if(is.matrix(m))
  {if(dim(m)[2]!=2)
     {print("Esta função precisa receber uma matriz ncol=2!!")
      return()
     }
  }
else
  {print("Esta função precisa receber uma matriz!!")
  return()
  }
nm<-nrow(m)
cat("Numero de linhas da matriz original = ", nm, "\n")

#Coloca m em ordem, segundo as colunas 1 e 2.
m<-m[order(m[,1],m[,2]),]
manter<-T
for (i in 1:nm-1) if(sum(m[i,]==m[i+1,])==2) manter[i+1]<-F else manter[i+1]<- T

#Seleciona os registros a manter.
m<-m[manter,]
cat("Numero de linhas da matriz deduplicada = ", nrow(m), "\n")
return(m)
}
```

20.1.2 Remova Cílios

Entrada: **TD**.
Saída: Matriz sem nós de grau 1 (**TS**).

```
f.remova.cilios<- function(m)

{
#Verifica se a função recebeu uma matriz da dimensão correta.
if(is.matrix(m))
  {if(dim(m)[2]!=2)
    {print("Esta função precisa receber uma matriz ncol=2!!")
     return()
    }
  }
else
  {print("Esta função precisa receber uma matriz!!")
   return()
  }
#Inicializa k, ite, dimensao.
k<-1
ite<-1
dimensao<-c(1,1)
cat("Dimensão inicial = ", dim(m), "\n")

while (dimensao != dim(m) | (ite<=2)) #Obrigado a verificar as duas colunas.
  {
  dimensao<-dim(m)
  manter<-F

  #Só marca os usuários que se repetem mais de uma vez.
  manter.no<-unique(m[,k][duplicated(m[,k])])
  if (length(manter.no) != length(unique(m[,k])))
    {
    for (i in 1:length(manter.no)) manter<-manter+(manter.no[i] == m[,k])
       #Reconstroi m sem os cílios.
           m<-m[as.logical(manter),]
    }
  cat("Iteração:",ite,"  Dimensão = ", dim(m), "\n")
  if (k==1) k<-2 else (k<-1)
  ite<-ite+1
  }
return(m)
}
```

20.1.3 Vértices

Entrada: Coluna (usuário ou livro) da matriz **TS**.
Saída: Vetor com os distintos nós existentes na coluna.

```
f.vertices<-function(no)

{
#Verifica se a função recebeu um vetor.
if(!is.vector(no))
  {print("Esta função precisa receber um vetor!!")
   return()
  }
no<-unique(sort(no))
return(no)
}
```

20.1.4 Lista Vizinhos

Entrada: Códigos de usuários da TS; códigos de livros da TS; tamanho dos vetores de usuários e livros; vetor de usuários distintos; chave (ligada, desligada) para remoção de vizinhanças de um único nó.

Saída: Lista de Vizinhos 1 e 2; Grau 2; e Coesão de cada vértice.

```
f.lista.vizinhos<-function(m1,m2,n1,n2,no,ligado)

{

v1<-list()              #Lista de vizinhos 1 do nó
v2<-list()              #Lista de vizinhos 2 do nó
nv1<-vector()           #Tamanho de V1
nv2<-vector()           #Tamanho de V2
co<-vector()            #Para calcular coesão
livros<-vector()

i<-1
while (i <=n1)
 {
  cat("no i ", no[i],"\n")
  #Acha vizinhança de livros do nó i.
  v1[[i]]<-m2[m1==no[i]]

  v<-vector()
  for(j in 1:length(v1[[i]])) {
    #Acha vizinhança de usuários do nó i.
    v<-unique(c(v,m1[m2==v1[[i]][j]]))
     }
  cat("ante m2",  m2,"\n")
  v2[[i]]<-v[!v==no[i]] #Retira o próprio nó da lista de vizinhos.
  nv1[i]<-length(v1[[i]])
  nv2[i]<-length(v2[[i]])
  cat("nv2i ", nv2[i],"\n")
  if(nv2[i]<=1 & ligado==1){
   n1<-n1-1
   elemento<-no[i]

   livros<-m2[m1==elemento]
   no<-no[no!=elemento]

   m2<-m2[m1!=elemento]
   m1<-m1[m1!=elemento]
```

```
    cat("m1 ", m1,"\n")
    cat("m2", m2,"\n")
     i<-1
   }
   else{

   #Cria matriz com duas colunas(usuário e livro), do mesmo tamanho
   #da matriz de transações deduplicadas, que receberá 1 nas posições
   #onde um elemento aparecer tanto em m1 quanto nas vizinhança do nó, v1[[i]].
   existe<-matrix(0,length(m1),2)
   for (k in 1:nv2[i]) existe[,1][m1==v2[[i]][k]]<-1
   for (k in 1:nv1[i]) existe[,2][m2==v1[[i]][k]]<-1

   ex<-dim(existe[(existe[,1]==1 & existe[,2]==1),])
   #Acha a coesão de cada nó.
   co[i]<- ex[1]/((nv1[i])*(nv2[i]))
   i<-i+1
    }
  }

USUARIO<<-no
 cat("Usuário\n", USUARIO)
 vale<-length(no)
 LV<-list(V1=v1[1:vale],V2=v2[1:vale],Grau2=nv2[1:vale],Coesao=co[1:vale])
 return(LV)
}
```

20.1.5 Cria Transação

Entrada: Vizinhança em 2 passos de um conjunto de vértices; grau em 2 passos; ids dos nós.
Saída: Lista de arestas entre os vértices.

```
f.cria.transacao<-function(v2, grau2, no)

{
 transacao<-data.frame(matrix(0,0,2))
 n<-length(no)
 indice<-1:n
 usado<-c(0*1:n)
 k<-1
 for(i in 1:n){
   for(j in 1:grau2[i]){
     l<-indice[no==v2[[i]][j]]
     if(usado[l]==0){
       transacao[k,1]<-no[i]
       transacao[k,2]<-no[l]
       k<-k+1
     }
   }
   usado[i]<-1
  }

return(transacao)
}
```

20.1.6 Lista Vizinhos Simples

Entrada: Lista de distâncias entre vértices; vetor com ids distintos de vértices.
Saída: Vizinhos em 1 passo; grau do nó; coesão do nó.

```
f.lista.vizinhos.simples<-function(distpp,vertice)

{

 v1<-list()
 total<-0
 co<-vector()
 nv1<-vector()#Tamanho de v1.
 aux<-matrix()
 n<-length(vertice)
 for(i in 1:n){
   cat("\n",vertice[i])
   aux<-distpp[,1:2][distpp[,1]==vertice[i]|distpp[,2]==vertice[i],]
   v1[[i]]<-sort(aux[aux!=vertice[i]])

   nv1[i]<-length(v1[[i]])
   k<-v1[[i]]
   auxc<-distpp[is.element(distpp[,1],k)&is.element(distpp[,2],k),]
   total<-nrow(auxc)

   co[i]<-(2*total)/(nv1[i]*(nv1[i]-1))
  }
return(V1=v1,Grau=nv1,Coesao=co)

}
```

20.1.7 Distância Temática

Entrada: LVu$V1 (isto é, vizinhança de usuário em 1 passo – portanto livros); LVu$V2 (isto é , vizinhança de usuário em 2 passos, - portanto usuários).
Saída: Matriz com distâncias temáticas entre usuários.

```
f.distancia<-function(V1,V2,grau2,no)

{

#Matriz contendo usuários nas colunas 1 e 2 e distância na coluna 3.
dist<-data.frame(matrix(0,1,3))

k<-1  #Indica linha da matriz dist.
n<-length(no)
indice<-1:n
usado<-c(0*1:n)
#Calculando distância entre usuário e usuário
for (i in 1:n)
  {
  for (j in 1:grau2[i])
    {
     cat("no",no,"\n")
     cat("i e j",i,j,"\n")

     l<-indice[no==V2[[i]][j]]
     if(usado[l]==0){#Evita calcular a mesma distância mais de uma vez
       cat("indice l",l,"\n")
       ui<-union(V1[[i]],V2[[i]])
       ul<-union(V1[[l]],V2[[l]])
       if (length(intersect(ui,ul))!=0) #Não registra distancias iguais a 1
             {dist[k,1]<-no[i]
          dist[k,2]<-no[l]
          dist[k,3]<-1-(length(intersect(ui,ul))/length(union(ui,ul)))
              k<-k+1
       }
     }
    }
    usado[i]<-1
  }
return(dist)
}
```

20.1.8 Agrupa

Entrada: Vértice, Vizinhos 2, Grau2, Coesão.
Saída: Vetor com o nome dos grupos em cada nível e número de centróides.

```
f.agrupa<-function(no, v2, g2, co)

{

k<-0
cont<-0
n<-length(no)
indice<-1:n

#Zero como número.
v2[g2==0]<-0
co[g2==0]<-0

#Ordena por Coesao e Grau2, controlando um índice.
o<-order(co,g2)
lv<-cbind(indice,co,g2)[o,]
print(lv)

L<-c(0*1:n)    #Cria vetor de tamanho n para colocar nome do grupo.
usado<-L       #Indica se vértice já foi usado.

i<-n #Como está na ordem crescente de coesão e grau2, o começo é pelo fim.
g<-1    #Número do grupo (centróide).

while (any(usado==0))
  {
  if (usado[i]==0)
    {
    usado[i]<-1
    L[i]<-g
    cont<-1
    #Para cada vizinho de lv[i,1] verifica se o vizinho já foi usado ou não.
    for (j in 1:lv[i,3]){
```

```
        #Coloca o vértice no último grupo coeso.
        if(usado[no==v2[[lv[i,1]]][j]]==1) k<-L[no==v2[[lv[i,1]]][j]]
        if(usado[no==v2[[lv[i,1]]][j]]==0)#Indice baseado no índice que o nó tinha inicialmente na matriz.
          {
           #Verifica se o vizinho já não pertence a outro grupo.
           usado[no==v2[[lv[i,1]]][j]]<-1
           L[no==v2[[lv[i,1]]][j]]<-g
           cont<-cont+1
          }
        }
  }
  if(cont<=2){
    L[L==g]<-k
    g<-g-1
  }
  i<-i-1
  g<-g+1
  cat(i,"linha",no[i],"usado",usado,"\n")
  cont<-0
 }

g<-g-1
cat("Número de Centróides: ",g, L,"\n")
#Joga para o 1 grupo    .
if(length(L[L==-1])<3)L[L==-1]<-1
return(LO=L,G=g)
}
```

20.1.9 Distância Pai – Filho

Entrada: Vetor **l** com os grupos dos vértices, quantidade de grupos, matriz dist com as distâncias temáticas entre todos os vértices, vetor dos usuários (vértices).
Saída: Matriz com todas as distâncias dos vértices de um grupo até o centróide.

```
f.distancia.pf<-function(l,g,dista,no)

{

#dista<-recebe a distância do corpo do trabalho
#dist.vetor->contém as distâncias em forma vetorial na matriz de 3 colunas.
#dist.matriz->contém distâncias em forma matricial conforme determina o cmdscale.
#dist.pf->uma linha contendo as distâncias entre pais e filhos durante processamento.

distfpf<-matrix(0,0,0)                  #Matriz completa com resultados.
dist.pf<-data.frame(matrix(0,1,3))              #Linha da vez.
i<-1
kk<-p

while (i<=g)
  {
  #Seleciona apenas as distancias do grupo i e monta matriz de distancia para MDS.
  dist.matriz<-f.monta.matriz(dista,no[l==i])

  #Matriz com as distancias entre elementos de um mesmo grupo.
  print(dist.matriz)

  #Escalona o grupo i.
  #A ordem de X está como no[l==i] para ligar cada ponto a um vertice em no[l==i].
  X<-cmdscale(dist.matriz)

  n<-length(no[l==i])

  #Acha as distâncias euclidianas até o centróide.
  cat("n",n,"\n\n")
  for(j in 1:n)
    {
     dist.pf[j,1]<-kk
     dist.pf[j,2]<-no[l==i][j]
     dist.pf[j,3]<-sqrt((X[j,1])^2+(X[j,2])^2)
    }
  dist.pf<-dist.pf[1:n,]
```

```
  #Cria id do novo centróide.
  kk<-kk+1
  i<-i+1
  distfpf<-rbind(distfpf,dist.pf)

 }
p<<-kk
distfpf<-distfpf[order(distfpf[,1]),]
return(distfpf)
}
```

20.1.10 Monta Matriz

Entrada: Vetor com as distâncias entre as os pontos de um mesmo grupo; vetor contendo os ids dos membros do grupo.
Saída: Matriz subdiagonal de distâncias entre os membros do grupo.

```
f.monta.matriz<-function(dist.vetor,no)

{

#dist.vetor->Contém as distâncias em matriz de 3 colunas.
dist.vetor<-as.matrix(dist.vetor[is.element(dist.vetor[,1],no)&is.element(dist.vetor[,2],no),])
m<-nrow(dist.vetor)

#Subdiagonaliza os índices.
for (i in 1:m)
{
if (dist.vetor[i,1]<dist.vetor[i,2])
        {
        temp<-dist.vetor[i,1]
        dist.vetor[i,1]<-dist.vetor[i,2]
        dist.vetor[i,2]<-temp
        }
}

if(ncol(dist.vetor)==1) dist.vetor<-t(dist.vetor)
n<-length(no)

#Monta matriz subdiagonal.
dist.matriz<-matrix(1,n,n)
diag(dist.matriz)<-0
indice<-c(1:n)
for (i in 1:m)
{
dist.matriz[indice[no==dist.vetor[i,1]],indice[no==dist.vetor[i,2]]]<-dist.vetor[i,3]
}

dist.matriz<-as.dist(dist.matriz)
return(dist.matriz)
}
```

20.1.11 Distância entre Centróides

Entrada: Vetor **l** com grupos dos vértices, quantidade de grupos, matriz dist com as distâncias temáticas entre todos os vértices, vetor dos usuários(vértices).
Saída: Matriz com todas as distâncias entre os centróides de um grupo.

```
f.distancia.pp<-function(l,g,dista,no)

{

library(mva)
#dista<-recebe a distância do corpo do trabalho (dist é reservando na library mva).
#dist.pp->contém as distancias entre os pais.

dist.pp<-matrix(0,nrow(dista),3)
#Monta matriz com as distâncias entre os vértices de um mesmo grupo.
for (i in 1:length(no))
  {
   dist.pp[,1][dista[,1]==no[i]]<-l[no==no[i]]
   dist.pp[,2][dista[,2]==no[i]]<-l[no==no[i]]
   dist.pp[,3]<-dista[,3]
  }
print(dist.pp)
#Subdiagonaliza índices.
for (i in 1:nrow(dist.pp)){
   if (dist.pp[i,1]  < dist.pp[i,2])
     {
      aux<-dist.pp[i,1]
      dist.pp[i,1]<-dist.pp[i,2]
      dist.pp[i,2]<-aux
     }
}
#Retira as distâncias ii.
dist.pp<-dist.pp[dist.pp[,1]!=dist.pp[,2],]
dist.pp[,1]<-(dist.pp[,1] + pp-1)
dist.pp[,2]<-(dist.pp[,2] + pp-1)
print(dist.pp)

#Ordena dist.pp.
dist.pp<-dist.pp[order(dist.pp[,1],dist.pp[,2],dist.pp[,3]),]

#Deduplica pela menor distância.
manter<-T
for (i in 1:(nrow(dist.pp)-1))
```

```
  {
   if (sum(dist.pp[i,]==dist.pp[i+1,])==3) {
     manter[i+1]<-F
   }
   else
     {
     if (sum(dist.pp[i,1:2]==dist.pp[i+1,1:2])==2)
       {
       manter[i+1]<-F
       }
     else manter[i+1]<-T
     }
   }

dist.pp<-dist.pp[manter,]

return(dist.pp)
}
```

20.1.12 Desce

Entrada: Matriz com coordenadas escalonadas dos centróides do nível; distância entre centróides; distância entre centróides e filhos; distâncias entre os filhos.

Saída: Configuração dos filhos; vértices distintos na configuração.

```
f.desce<-function(X,dpp,dpf,dff)

{

vertfinal<-vector()
pontosfinais<-matrix(0,1,2)
#Começa descida pelo topo, ou seja, pelos pais
 distcent<-dpp
 while(dim(distcent)[1]!=0){
  #discent receberá o inicial dpp e será destituída de pontos até acabar
  triangle<-f.triangulo(distcent,matriz$n,X)
  distcent<-triangle$distancia
  familia<-f.reune(triangle,dpf,dff)
  familiac<-f.monta.matriz.simples(familia$o)
  vertfinal<-append(vertfinal,familiac$n)
  cat("\n\nvertfinal\n")
  cat(vertfinal)
  vertfinal<-vertfinal[1:(length(vertfinal)-length(triangle$pf))]

  pontos<-cmdscale(familiac$pontos)
  cat("\n\npontos\n\n")
  print(pontos)
  pontosp<-pontos[(length(pontos[,1])-length(triangle$pf)+1):length(pontos[,1]),]
  pontosf<-pontos[1:(length(pontos[,1])-length(triangle$pf)),]
  cat("\n\npontosf\n\n")
  print(pontosf)

#Ajusta coordenadas.
  pro<-f.procrustes(pontosp,triangle$trireal)
  pontosf<-f.ajustar(pontosf,pro$A,pro$ro,pro$B)

  pontosfinais<-rbind(pontosfinais,pontosf)
  cat("\n\n\n fiz uma vez\n\n\n")
  cat("\n\npontosfinais\n\n")
  print(pontosfinais)
 }
 pontosfinais<-pontosfinais[2:nrow(pontosfinais),]
```

```
 cat("\n\nvertfinal\n")

 ord<-order(vertfinal)
 cat(vertfinal,ord)
 cat("pontosfinais\n")
 print(pontosfinais)
 pontosfinais<-pontosfinais[ord,]
 cat("pontosfinais\n")
 print(pontosfinais)

 return(p=pontosfinais,v=sort(vertfinal))
}
```

20.1.13 Triângulo

Entrada: Matriz com as distâncias entre centróides (DIST), nós (vértices) do nível a ser escalonado, matriz de configuração (real) dos pontos escalonados.

Saída: Lista contendo as distâncias remanescentes em DIST, as distâncias entre os três pontos escolhidos, os três pontos escolhidos, e suas respectivas coordenadas escalonadas.

```
f.triangulo<-function(DIST, vertices, real)

{

dim<-1
distf<-matrix()
distf1<-vector()
aux1<-vector()
dist1<-dist2<-vector()
matriz<-matrix()
pai<-vector()
mreal<-matrix()
n<-length(DIST[,1])
cat(n)
DIST<-DIST[order(DIST[,3]),]
print(DIST)

#Controle sobre arestas existentes.
ok1<-0
ok2<-0
ok3<-0
i<-1

#Para três ou mais que cinco vértices em DIST.
 while((ok1==0&ok2==0) & i<=(nrow(DIST)-1)){

  distf1<-DIST[i,]
  cat("distf1",distf1)
  x<-distf1[1]
  y<-distf1[2]
  comp<-c(x,y)
  cat(x,y,"\n")
  DIST1<-as.matrix(DIST[(i+1):n,])
  if(ncol(DIST1)==1)DIST1<-t(DIST1)
  print(DIST)
  cat("\n")
```

```
 dist2<-as.matrix(DIST1[(DIST1[,1]==x|DIST1[,2]==x),])
 if(nrow(dist2)!=0){
  ok2<-1
  if(ncol(dist2)!=1)dist2<-dist2[1,]
 }
 dist1<-as.matrix(DIST1[(DIST1[,1]==y|DIST1[,2]==y),])
 if(nrow(dist1)!=0){
  ok1<-1
  if(ncol(dist1)!=1)dist1<-dist1[1,]
 }
 i<-i+1
}
if(ok1==0&ok2==0){
  linha<-length(DIST[,1])
  if(linha!=3){
    cat("linha",linha,"\n")
    distf<-matrix(1,6,3)
    mreal<-matrix(1,4,2)
    distf[1,]<-DIST[1,];distf[2,]<-DIST[2,];
    pai<-DIST[1,1:2]
    pai<-append(pai,DIST[2,1:2])
    if(nrow(DIST)==2)DIST<-matrix(0,0,0)
    else DIST<-DIST[3,]

    #Recupera na matriz escalonada as coordenadas dos pontos escolhidos.
    indice<-c(1:length(vertices))
    real<-real[indice[is.element(vertices,pai)],]
    kk<-2
    for(i in 1:2){
      for(j in 3:4){
         kk<-kk+1
         distf[kk,1:3]<-c(max(pai[i],pai[j]),min(pai[i],pai[j]),1)
      }
    }
    print(real)
      for(i in 1:4){
      mreal[i,1:2]<-real[i,1:2]
    }

    #Distâncias Euclidianas.
    distf<-distf[order(distf[,2],distf[,1]),]
    t<-1
    for(i in 1:3){
          for(j in (i+1):4){
```

```
      distf[t,3]<-sqrt((real[i,1]-real[j,1])^2+(real[i,2]-real[j,2])^2)
       t<-t+1
     }
  }
  matriz<-distf
   }
 else{
  distf<-matrix(1,15,3)
  mreal<-matrix(1,4,2)
  distf[1,]<-DIST[1,];distf[2,]<-DIST[2,];distf[3,]<-DIST[3,];
  pai<-DIST[1,1:2]
  pai<-append(append(pai,DIST[2,1:2]),DIST[3,1:2])
  DIST<-matrix(0,0,0)

  #Recupera na matriz escalonada as coordenadas dos pontos escolhidos.
  indice<-c(1:length(vertices))
  real<-real[indice[is.element(vertices,pai)],]
  kk<-3
  for(i in 1:2){
    for(j in 3:6){
      kk<-kk+1
      distf[kk,1:3]<-c(max(pai[i],pai[j]),min(pai[i],pai[j]),1)
    }
  }
  for(i in 3:4){
    for(j in 5:6){
      kk<-kk+1
      distf[kk,1:3]<-c(max(pai[i],pai[j]),min(pai[i],pai[j]),1)
    }
  }
  for(i in 1:6){
    mreal[i,1:2]<-real[i,1:2]
  }

  #Distâncias Euclidianas
  distf<-distf[order(distf[,2],distf[,1]),]
  t<-1
  for(i in 1:5){
       for(j in (i+1):6){
      distf[t,3]<-sqrt((real[i,1]-real[j,1])^2+(real[i,2]-real[j,2])^2)
       t<-t+1
     }
  }
  matriz<-distf
```

```
  }
}
else {
 if(ok1==1&ok2==1){
  #Acha menor distância entre os vértices que contenham x e y.
 if(dist1[3] < dist2[3] ){
   k<-dist1[1:2][dist1!=comp[1]&dist1!=comp[2]][1]
   distf2<-dist1
   cat(k,"\n")
   distf3<-as.matrix(DIST[(DIST[,1]==x|DIST[,2]==x) &  (DIST[,1]==k|DIST[,2]==k),] )
   if(nrow(distf3)==0)distf3<-c(max(x,k),min(x,k),1)
   DIST<-DIST[(DIST[,1]!=y&DIST[,2]!=y)&
      (DIST[,1]!=x&DIST[,2]!=x)&(DIST[,1]!=k&DIST[,2]!=k),]
  }
  else{
   k<-dist2[1:2][dist2!=comp[1]&dist2!=comp[2]][1]
   distf2<-dist2
   distf3<-as.matrix(DIST[(DIST[,1]==y|DIST[,2]==y) & (DIST[,1]==k|DIST[,2]==k),] )
   if(nrow(distf3)==0)distf3<-c(max(y,k),min(y,k),1)
   DIST<-DIST[(DIST[,1]!=x&DIST[,2]!=x)&
      (DIST[,1]!=y&DIST[,2]!=y)&(DIST[,1]!=k&DIST[,2]!=k),]
   }
  }
  else{
     if(ok1==1&ok2==0){
       k<-dist1[1:2][dist1!=comp[1]&dist1!=comp[2]][1]
       distf2<-dist1
       cat(k,"\n")
       distf3<-as.matrix(DIST[(DIST[,1]==x|DIST[,2]==x) & (DIST[,1]==k|DIST[,2]==k),] )
       if(nrow(distf3)==0)distf3<-c(max(x,k),min(x,k),1)
       DIST<-DIST[(DIST[,1]!=y&DIST[,2]!=y)&
         (DIST[,1]!=x&DIST[,2]!=x)&(DIST[,1]!=k&DIST[,2]!=k),]
      }
      else{
       if(ok1==0&ok2==1){
        k<-dist2[1:2][dist2!=comp[1]&dist2!=comp[2]][1]
        distf2<-dist2
        distf3<-as.matrix(DIST[(DIST[,1]==y|DIST[,2]==y) & (DIST[,1]==k|DIST[,2]==k),] )
        if(nrow(distf3)==0)distf3<-c(max(y,k),min(y,k),1)
        DIST<-as.matrix(DIST[(DIST[,1]!=x&DIST[,2]!=x)&
           (DIST[,1]!=y&DIST[,2]!=y)&(DIST[,1]!=k&DIST[,2]!=k),])
        }
      }
```

```
 pai<-c(x,y,k)
}

if(nrow(DIST)==1){
    aux1<-DIST
    pai<-unique(append(pai,aux1[1:2]))
    distf<-matrix(1,10,3)
    mreal<-matrix(1,5,2)
    matriz<-DIST[order(DIST[,1],DIST[,2]),]
       cat("pai",pai)

    #Recupera na matriz escalonada as coordenadas dos pontos escolhidos.
    indice<-c(1:length(vertices))
    real<-real[indice[is.element(vertices,pai)],]
    n<-length(pai)

    #Monta matriz das distancias entre os 4 pontos escolhidos.
    distf[1,1:2]<-distf1[1:2];distf[2,1:2]<-distf2[1:2];
              distf[3,1:2]<-distf3[1:2];distf[4,1:2]<-aux1[1:2];
    for(i in 5:7){
      for(j in 1:3){
       distf[i,1:3]<-c(max(pai[4],pai[j]),min(pai[5],pai[j]),1)
      }
    }
    for(i in 8:10){
      for(j in 1:3){
       distf[i,1:3]<-c(max(pai[5],pai[j]),min(pai[5],pai[j]),1)
      }
    }
    for(i in 1:5){
          mreal[i,1:2]<-real[i,1:2]
     }
     #Distâncias Euclidianas
         distf<-distf[order(distf[,2],distf[,1]),]
         t<-1
     for(i in 1:3){
          for(j in (i+1):5){
        distf[t,3]<-sqrt((real[i,1]-real[j,1])^2+(real[i,2]-real[j,2])^2)
        t<-t+1
           }
         }

        matriz<-distf
    DIST<-matrix(0,0,0);
```

```
    }
   else{
    distf<-matrix(1,3,3)
    mreal<-matrix(1,3,2)
    pai<-c(x,y,k)

    #Recupera na matriz escalonada as coordenadas dos pontos escolhidos.
    indice<-c(1:length(vertices))
    real<-real[indice[is.element(vertices,pai)],]

    #Monta matriz das distâncias entre os 3 pontos escolhidos.
    distf[1,1:2]<-distf1[1:2];distf[2,1:2]<-distf2[1:2];distf[3,1:2]<-distf3[1:2];

    for(i in 1:3){
      mreal[i,1:2]<-real[i,1:2]
          }

    #Distâncias Euclidianas.
    distf<-distf[order(distf[,2],distf[,1]),]
    t<-1
    for(i in 1:2){
          for(j in (i+1):3){
       distf[t,3]<-sqrt((real[i,1]-real[j,1])^2+(real[i,2]-real[j,2])^2)
           t<-t+1
      }
    }
    matriz<-distf
   }
 }
   cat("DISTfinal\n")

return(distancia=DIST, tri=matriz, pf=pai, trireal=mreal)
}
```

20.1.14 Reúne

Entrada: Distâncias remanescentes em DIST; distâncias entre 3 pais escolhidos; ids dos pais escolhidos; coordenadas dos pais escolhidos; distâncias entre pais e filhos; distâncias entre filhos.

Saída: Matriz contendo as distâncias relevantes reunidas.

```
f.reune<-function(distc,distpf,distff)

{

 o<-matrix()
 auxm<-matrix()
 print(distc)
 paifi<-vector()
 distfpf<-matrix(0,1,3)
 distfff<-matrix(0,1,3)
 mpais<-distc$pf
 cat("mpais",mpais,"\n")

 n<-length(mpais)
 if(n>3){
  if(n==4){
   #Seleciona distância dos centróides com os filhos
   distfpf<-distpf[(distpf[,1]==mpais[1]|distpf[,2]==mpais[1])
        |(distpf[,1]==mpais[2]|distpf[,2]==mpais[2])
        |(distpf[,1]==mpais[3]|distpf[,2]==mpais[3])
            |(distpf[,1]==mpais[4]|distpf[,2]==mpais[4]),]

   cat("distfpf\n")
   cat("n",n,"\n")
   }
  else{
     #Seleciona distância dos centróides com os filhos
     distfpf<-distpf[(distpf[,1]==mpais[1]|distpf[,2]==mpais[1])
        |(distpf[,1]==mpais[2]|distpf[,2]==mpais[2])
        |(distpf[,1]==mpais[3]|distpf[,2]==mpais[3])
            |(distpf[,1]==mpais[4]|distpf[,2]==mpais[4])
        |(distpf[,1]==mpais[5]|distpf[,2]==mpais[5]),]

   cat("distfpf\n")
   cat("n",n,"\n")
```

```
    }
 }
 else{

    if(n==3){
     #Seleciona distância dos centróides com os filhos
     distfpf<-distpf[(distpf[,1]==mpais[1]|distpf[,2]==mpais[1])
        |(distpf[,1]==mpais[2]|distpf[,2]==mpais[2])
        |(distpf[,1]==mpais[3]|distpf[,2]==mpais[3]),]

     cat("distfpf\n")
     cat("n",n,"\n")
    }
 }

 paifi<-unique(distfpf[,2])
 cat("paifi",paifi,"\n")
 #Encontra distâncias entre os filhos de um mesmo centróide
 auxm<-distff[is.element(distff[,1],paifi)&is.element(distff[,2],paifi),]
 cat("auxm","\n")
 print(auxm)
 distfff<-rbind(distfff,auxm)

 distfpf<-distfpf[order(distfpf[,1]),]
 distfff<-distfff[2:length(distfff[,1]),]
 distfff<-distfff[order(distfff[,1],distfff[,2]),]

 tri<-distc$tri
 if(n==3) o<-rbind(tri[1,],tri[2,],tri[3,],distfpf,distfff)
 else if(n==4) o<-rbind(tri[1,],tri[2,],tri[3,],tri[4,],tri[5,],tri[6,],
              distfpf,distfff)
     else if(n==5) o<-rbind(tri[1,],tri[2,],tri[3,],tri[4,],tri[5,],
                       tri[6,],tri[7,],tri[8,],tri[9,],
                          tri[10,],distfpf,distfff)

 print(o)
 return (o=o,t=n)
}
```

20.1.15 Monta Matriz Simples

Entrada: Matriz n x 3 com distâncias.
Saída: Matriz n x n subdiagonal, com distâncias.

```
f.monta.matriz.simples<-function(dist.vetor)

{

no<-sort(unique(c(dist.vetor[,1],dist.vetor[,2])))
n<-length(no)
cat(no)
dist.matriz<-matrix(1,n,n)
diag(dist.matriz)<-0
indice<-1:n
for(i in 1:n){
  dist.vetor[dist.vetor[,1]==no[i],1]<-indice[i]
  dist.vetor[dist.vetor[,2]==no[i],2]<-indice[i]
  }
dist.vetor<-arrumaposicao(dist.vetor)
print(dist.vetor)
for(i in 1:nrow(dist.vetor))
 dist.matriz[(dist.vetor[i,1]),(dist.vetor[i,2])]<-dist.vetor[i,3]

dist.matriz<-as.dist(dist.matriz)
return(pontos=dist.matriz,n=no)
}
```

20.1.16 Arruma Posição

Entrada: Matriz n x 3 de distâncias.
Saída: Mesma matriz, com índices subdiagonalizados.

```
arrumaposicao<-function(matriz)

{

 for (i in 1:nrow(matriz)){
   if (matriz[i,1]  < matriz[i,2])
     {
      aux<-matriz[i,1]
      matriz[i,1]<-matriz[i,2]
      matriz [i,2]<-aux
     }
 }

return(matriz)

}
```

20.1.17 Traço

Entrada: Matriz quadrada.
Saída: Traço da matriz.

```
tr<-function(x)

{
t<-sum(diag(x))
return(t)
}
```

20.1.18 Procrustes

Entrada: Configuração a ser deformada; configuração fixa.
Saída: Matriz de rotação; fator de dilação; translação rígida; coeficiente de procrustes.

```
f.procrustes<-function(xv,yv)

{

x<-xv
y<-yv

if (dim(x)[1]!=dim(y)[1]|dim(x)[2]!=dim(y)[2]|dim(x)[2]!=2)
        {
        print("Erro: as matrizes devem ter o mesmo número de linhas e duas colunas.")
        return()
        }

#Centrar x e y no (0,0).
xo<-t(as.matrix(c(mean(x[,1]),mean(x[,2]))))
x[,1]<-x[,1]-xo[1,1]
x[,2]<-x[,2]-xo[1,2]
yo<-t(as.matrix(c(mean(y[,1]),mean(y[,2]))))
y[,1]<-y[,1]-yo[1,1]
y[,2]<-y[,2]-yo[1,2]

#Rotação ótima
c<-t(x)%*%y%*%t(y)%*%x
ei<-eigen(c)
sqrtc<-ei$vectors%*%diag(sqrt(ei$values))%*%t(ei$vectors)
A<-sqrtc%*%solve(t(y)%*%x)

#Dilação
ro<-tr(sqrtc)/tr(t(x)%*%x)

#Translação rígida
B<-yv-ro*x%*%A

#Qualidade do ajuste
R2<- 1-  (  (tr(sqrtc)^2)  /  (tr(t(x)%*%x)*tr(t(y)%*%y))  )

return(A=A, ro=ro, B=B, R2=R2)
}
```

20.1.19 Ajustar

Entrada: Coordenadas a ajustar, matriz de rotação, fator de dilação, translação rígida.
Saída: Coordenadas ajustadas.

```
f.ajustar<-function(X, A, ro, B)

{

#Acha médias das colunas de B.
b1<-mean(B[,1])
b2<-mean(B[,2])

B<-matrix(0, nrow(X), 2)
B[,1]<-b1
B[,2]<-b2

X<-(ro*X)%*%A+B
return(X)
}
```

20.1.20 Suaviza Dados

Entrada: Matriz com coordenadas x e y da configuração; coesão de cada ponto; extremos do gráfico criado anteriormente, raio para suavização, número de partes em que dividir os eixos.

Saída: Coordenadas em x e coordenadas em y dos pontos de medida; e valor z das medidas em forma de matriz.

```
f.suaviza.grafico<-function(matriz,coesao,parametros,raio,lado)

{

#Cria matriz grade.
matriz<-cbind(matriz,coesao)
grade<-matrix(0,lado^2,3)

#Fixa distância entre pontos.
dx<-(parametros[2]-parametros[1])/lado
dy<-(parametros[4]-parametros[3])/lado

xcoord<-seq(parametros[1]+dx,parametros[2],dx)
ycoord<-seq(parametros[3]+dy,parametros[4],dy)

for (i in 0:(lado-1))
for (j in 1:lado)
        {
        grade[(i*lado)+j,1]<-xcoord[i+1]
        grade[(i*lado)+j,2]<-ycoord[j]
        }

nm<-nrow(matriz)

#Verifica para cada ponto de medida todos os vértices no seu raio de alcance.
for(i in 1:lado^2)
{
  print(i)
  co<-c()

  for(j in 1:nm)
        {
    #Teste se está dentro do círculo.
    dist<-sqrt((grade[i,1]-matriz[j,1])^2+(grade[i,2]-matriz[j,2])^2)
    if(dist<=(raio*dx)) co<-append(co,matriz[j,3])
```

```
      }

   #Grade[i] recebe coesão média dos elementos de seu raio de alcance.
   if(is.nan(mean(co))) grade[i,3]<-0
   else grade[i,3]<-mean(co)
}

dados<-matrix(grade[,3],lado,lado,byrow=TRUE)

return(x=xcoord, y=ycoord, z=dados)

}
```

20.2. Corpo do Programa para RGR do Exemplo 2

```
######################  Pré - Processamento  ########################

library(mva)

TO<-as.matrix(read.table("IMEtrans2.dat"))
save(TO, file="TO.r")

TD<-f.deduplique(TO)
save(TD, file="TD.r")
rm(TO)

TS<-f.remova.cilios(TD)
save(TS, file="TS.r")
colnames(TS)<-c("usuário","livro")
rm(TD)

USUARIO<-f.vertices(TS[,1])
LIVRO<-f.vertices(TS[,2])

LV<-f.lista.vizinhos(TS[,1],TS[,2],length(USUARIO),length(LIVRO),USUARIO,1)
save(LV, file="LV1.r")

f.lista.vizinhos(ts[,1],ts[,2],length(USUARIO1),length(LIVRO1),USUARIO1,1)

dp0p0<-f.distancia(LV$V1,LV$V2,LV$Grau2,USUARIO)
save(dp0p0, file="dp0p0.r")

######################  Início do "Sobe"  ########################

############################  Nível 0  ############################

VERTICE0<-USUARIO

L<-f.agrupa(VERTICE0,LV$V2,LV$Grau2,LV$Coesao,0)
L0<-L$L #Listagem na ordem de VERTICE, indicando o grupo que cada usuário pertence.
G<-L$G  #Número de grupos formados no nível 1 (número de centróides)
rm(L)

pp<-p<-VERTICE0[length(VERTICE0)]+1

dp1p0<-f.distancia.pf(L0,G,dp0p0,VERTICE0)
save(dp1p0, file="dp1p0.r")
```

```
dp1p1<-f.distancia.pp(L0,G,dp0p0,VERTICE0)
save(dp1p1, file="dp1p1.r")

#####################  Início do "Desce"  ########################

matriz<-f.monta.matriz.simples(dp1p1)
X1<-cmdscale(matriz$pontos)
vert1<-matriz$n

p1<-f.desce(X1,dp1p1,dp1p0,dp0p0,vert1)
X0<-p1$p
vert0<-p1$v

#####################  Gerenciamento do Bug  ####################
dp1p0s<-dp1p0
vertex<-setdiff(VERTICE0,vert0)
for(i in 1:length(vertex))
  {
  dp1p0s<-dp1p0s[(dp1p0s[,1]!=vertex[i]&dp1p0s[,2]!=vertex[i]),]
  }

dp0p0s<-dp0p0
vertex<-setdiff(VERTICE0,vert0)
for(i in 1:length(vertex))
  {
  dp0p0s<-dp0p0s[(dp0p0s[,1]!=vertex[i]&dp0p0s[,2]!=vertex[i]),]
  }
save.image(".RData")

#########################      FIM      ##########################

#####################################################################


#################### Gráfico de Pais e Filhos ####################
plot(X0,
ann=FALSE,
)
points(X1, pch=16)

n<-nrow(dp1p0s)
for (i in 1:n)
        {
        aresta<-matrix(0,2,2)
```

```
        aresta[1,1]<-X1[(vert1==dp1p0s[i,1]),1]
        aresta[1,2]<-X1[(vert1==dp1p0s[i,1]),2]
        aresta[2,1]<-X0[(vert0==dp1p0s[i,2]),1]
        aresta[2,2]<-X0[(vert0==dp1p0s[i,2]),2]
        lines(aresta)
        }

############### Gráfico da Rede Com Base Neste Algoritmo ################

plot(
X0,
ann=FALSE,   #sem títulos
bty="n",     #sem caixa
xaxt="n",    #sem eixo x
yaxt="n",    #sem eixo y
xlim=1.2*range(X0[,1]),
ylim=1.2*range(X0[,2]),
pch=16,
)

n<-nrow(dp0p0s)
for (i in 1:n)
        {
        aresta<-matrix(0,2,2)
        aresta[1,1]<-X0[(vert0==dp0p0s[i,1]),1]
        aresta[1,2]<-X0[(vert0==dp0p0s[i,1]),2]
        aresta[2,1]<-X0[(vert0==dp0p0s[i,2]),1]
        aresta[2,2]<-X0[(vert0==dp0p0s[i,2]),2]
        lines(aresta)
        }

############################## Grafico de X0 com legendas###############

plot(
X0,
type="n",
ann=FALSE,
xlim=1.2*range(X0[,1]),
ylim=1.2*range(X0[,2])
)

text(X0[,1], X0[,2], vert0, font=2, cex=.5)
```

```
################### Gráfico em Perspectiva ####################

        plot(X0,
        ann=FALSE,xlim=1.2*range(X0[,1]),ylim=1.2*range(X0[,2]), type="n")
        user<-par("usr")
        escolhe<-is.element(USUARIO, setdiff(USUARIO, vert0))
        coesao<-LV$Coesao[!escolhe]

        X0grid<-f.suaviza.grafico(X0, coesao, user, 2, 20)

        contour(X0grid$x,X0grid$y, X0grid$z)
        points(X0, pch=16)

        win.graph()
        persp(
                X0grid$x, X0grid$y, X0grid$z,
                ann=FALSE,
                theta = 30,
                phi = 30,
                col = "lightblue",
                shade= 0.3,
                xlab="",
                ylab="",
                zlab="Coesão",
                box=TRUE
                )
```

20.3. Funções para Estudo da Estabilidade do Algoritmo de RGR

20.3.1 Algoritmo RGR (função f.alg.chico)

```
###############ALGORITMO RGR  ###################################
#Entrada: Matriz de transacoes iniciais (TO),
#         Original (1: TO original ou 0: TO de amostragem).
#Saída:
# original==1: TS, USUARIO, LIVRO, V1, Coesao, X0, vert0
# original==0: USUARIO, LIVRO, Coesao, X0, vert0
################################################################

f.alg.chico<-function(TO,original)
{
cat("Funcao ALG.CHICO","\n")

library(mva)
###################### Pré - Processamento ######################
TD<-f.deduplique(TO)
TS<-f.remova.cilios(TD)
colnames(TS)<-c("usuário","livro")

USUARIO<-f.vertices(TS[,1])
USU<-USUARIO
LIVRO<-f.vertices(TS[,2])

LV<-
f.lista.vizinhos(TS[,1],TS[,2],length(USUARIO),length(LIVRO),USUARIO,1)
USUARIO<-LV$Usuario

cat("Numero de usuario = ",length(USUARIO),"\n")
cat("Numero de livros =  ",length(LIVRO),"\n")
if (length(USUARIO)<15)
  {
  cat("Numero de usuario menor do que 10 nao concluimos o algoritmo!!
\n")
  return()
  }

dp0p0<-f.distancia(LV$V1,LV$V2,LV$Grau2,USUARIO)

VERTICE0<-USUARIO
if (original==1)
  {
  ###################### Início do "Sobe" ######################
  L<-f.agrupa(VERTICE0,LV$V2,LV$Grau2,LV$Coesao)
  L0<-L$L #Listagem na ordem de VERTICE, indicando o grupo a que cada
          usuário pertence.
  G<-L$G  #Número de grupos formados no nível 1 (número de centróides)
  rm(L)

  dp1p0<-
f.distancia.pf(L0,G,dp0p0,VERTICE0,VERTICE0[length(VERTICE0)]+1)
```

```
  dp1p1<-
f.distancia.pp(L0,G,dp0p0,VERTICE0,VERTICE0[length(VERTICE0)]+1)

  ######################## Início do "Desce" ########################
  matriz<-f.monta.matriz.simples(dp1p1)
  X1<-cmdscale(matriz$pontos)
  vert1<-matriz$n
  p1<-f.desce(X1,dp1p1,dp1p0,dp0p0,vert1)
  X0<-p1$p
  vert0<-p1$v
  }
else
  {
  matriz<-f.monta.matriz.simples(dp0p0)
  X0<-cmdscale(matriz$pontos)
  vert0<-matriz$n
  }

######################## Gerenciamento do Bug ####################
########## Das distancias para construcao dos gráficos  #########

if (original==1)
  {
  dp1p0s<-dp1p0
  vertex<-setdiff(VERTICE0,vert0)
  for(i in 1:length(vertex))
    dp1p0s<-dp1p0s[(dp1p0s[,1]!=vertex[i]&dp1p0s[,2]!=vertex[i]),]
  }

dp0p0s<-dp0p0
vertex<-setdiff(VERTICE0,vert0)
for(i in 1:length(vertex))
  dp0p0s<-dp0p0s[(dp0p0s[,1]!=vertex[i]&dp0p0s[,2]!=vertex[i]),]


#Para parte inferencial - tirar usuarios excluidos em LV e em X0
vertex<-setdiff(USU,vert0)
for(i in 1:length(vertex))
  TS<-TS[TS[,1]!=vertex[i],]

LIVRO<-f.vertices(TS[,2])

#Salvando algumas matrizes importantes
if (original==1)
  {
  save(TD, file="TD.r")
  save(TS, file="TS.r")
  save(LV, file="LV1.r")
  save(dp0p0, file="dp0p0.r")
  }

if (original==1)
```

```
return(TS=TS,USUARIO=USUARIO,LIVRO=LIVRO,V1=LV$V1,Coesao=LV$Coesao,X0=X0
,vert0=vert0)
else
  return(USUARIO=USUARIO,LIVRO=LIVRO,Coesao=LV$Coesao,X0=X0,vert0=vert0)
}
```

20.3.2 Algoritmo para Amostragem (f.alg.amostra)

```
####################ALGORITMO PARA AMOSTRAGEM  ##########################
#Entrada:   Matriz de transacoes simplificadas original (SAIDA$TS),
#           Vertice com os usuarios (SAIDA$USUARIO),
#           Vertice com os livros (SAIDA$LIVRO),
#           Lista de vizinhos livros dos usuarios (SAIDA$V1),
#           Coordenadas dos usuarios da TS original (SAIDA$X0),
#           Vertices das coordenadas consideradas (SAIDA$vert0),
#           Percentil considerado para construir vizinhanca de livros
p=(0,1]),
#           Numero de amostras para escalonamento (repl).
#
#Saída:     Lista de coordenadas de cada replica (X0a),
#           Lista de vertices de cada replica (vert0a),
#           Coesao dos usuarios que entraram em cada replica (Coesaoa),
#          Usuarios em cada replica (pode ser maior do que o vert0a)
(USUARIOa),
#         Valor de cada R2 do procruste entre a coord. da amostra e da
original.
#########################################################################

f.alg.amostra<-function(TS,USUARIO,LIVRO,V1,X0,vert0,repl,alpha)
{
cat("Funcao ALG.AMOSTRA","\n")

#Constroi a lista de vizinhos em um passo (usuarios) dos livros
LVb<-f.lista.vizinhos.livros(TS[,2],TS[,1],LIVRO)
print(LVb)

#Distancia entre livros e usuarios
DISTUL<-f.distul(X0,vert0,LVb$V1,LVb$Grau1,LIVRO)
Xb<-DISTUL$Xb
DISTUL<-DISTUL$Dist

#Tamanho da amostra baseada no numero de livros que cada usuario retirou
m<-vector()
for (i in 1:length(USUARIO)) m[i]<-length(V1[[i]])
vertex<-setdiff(USUARIO,vert0)
m<-cbind(m,USUARIO)
for(i in 1:length(vertex)) m<-m[m[,2]!=vertex[i],]
m<-as.vector(m[,1])

print("Codigo dos vertices na matriz de transacoes originais (TO) =
vert0")
print(vert0)
print("Numero de livros que cada usuario retirou em TO = m")
print(m)

X<-list()
vert<-list()
co<-list()
no<-list()
```

```
r2<-vector() #recebe o R2 de cada procruste
cont<-1
while (cont<=repl)
  {
  cat("Contador=",cont,"\n")
  TOa<-f.amostra(DISTUL,vert0,LIVRO,m,alpha)
  SAIDA<-f.alg.chico(TOa,0)

  if (length(SAIDA)!=0)
    {
    #Deixar X0 da TO original com os mesmos usuarios da X0 amostrada
    vertex<-setdiff(vert0,SAIDA$vert0)
    x<-cbind(X0,vert0)
    for(i in 1:length(vertex))
      x<-x[x[,3]!=vertex[i],]
    x<-x[,1:2]

    pro<-f.procrustes(SAIDA$X0,x)
    x<-f.ajustar(SAIDA$X0,pro$A,pro$ro,pro$B)  #coordenadas  da  amostra
deformada pela X0

    X[[cont]]<-x
    vert[[cont]]<-SAIDA$vert0
    co[[cont]]<-SAIDA$Coesao
    no[[cont]]<-SAIDA$USUARIO
    r2[cont]<-pro$R2

    cont<-cont+1
    }
  }
return(X0a=X,vert0a=vert,Coesaoa=co,USUARIOa=no,R2=r2,Xb=Xb)
}
```

20.3.3 Agoritmo para Inferência (f.inferência)

```
####################ALGORITMO PARA INFERENCIA  ##########################
#Entrada:
#     Coordenadas dos usuarios da transacao original (SAIDA$X0),
#     Codigo dos vertice referentes a essas coordenadas (SAIDA$vert0),
#     Lista com as coordenadas dos usuarios amostrados (AMOSTRA$X0a),
#     Lista   com   os   codigos   dos   usuarios   dessas   coordenadas
(AMOSTRA$vert0a),
#     Lista   das   coesoes   de   cada   usuario   em   cada   amostra
(AMOSTRA$Coesaoa),
#     Lista dos codigos dos usuarios referentes as coesoes(pode ser ou
nao igual a AMOSTRA$vert0a) (AMOSTRA$USUARIOa),
#     Numero que replicas (repl).
#
#Saída:
#     Lista contendo todas as coesoes calculadas na amostra de cada
usuario,
#     Matriz contendo a soma das coordenadas (amostra) do mesmo usuario,
#     Numero de vezes que cada vertice saiu em todas as amostras.
########################################################################

f.inferencia<-function(X0,vert0,X0a,vert0a,coa,noa,repl)
{
cat("Funcao INFERENCIA","\n")

allcoa<-list() #Lista do tamanho de vert0 contendo todas coesoes de cada
usuario
for (i in 1:length(vert0)) allcoa[[i]]<-vector()

sX0a<-matrix(0,length(vert0),2) #Matriz contendo a soma das coordenadas
do mesmo usuario
n0a<-rep(0,length(vert0)) #Contem o numero de vezes que cada vertice
saiu nas replicas (repl)

id<-1:length(vert0)
for (i in 1:repl)
  {
  cat("Repl=",i,"\n")
  for (j in 1:length(vert0a[[i]]))
    {
    k<-id[vert0a[[i]][j]==vert0]
    sX0a[k,]<-sX0a[k,]+X0a[[i]][j,]
    n0a[k]<-n0a[k]+1
    }
  for (j in 1:length(noa[[i]])) #noa é um vetor maior ou igual ao vert0a
    {
    k<-id[noa[[i]][j]==vert0]
    allcoa[[k]][length(allcoa[[k]])+1]<-coa[[i]][j]
    }

  }
return(Coesao=allcoa,sX0a=sX0a,n0a=n0a)
```

```
}
```

20.3.4 Lista Vizinhos dos Livros em um Passo

```
#############################################################################
#
#Lista Vizinhos dos Livros em um Passo
#Entrada:   Códigos de livros da TS (TS[,2]),
#           Códigos de usuarios da TS (TS[,1]),
#           Vetor de livros distintos (LIVRO)
#
#Saida:     Lista de Vizinhos 1 (usuarios) dos livros
#############################################################################


f.lista.vizinhos.livros<-function(m1,m2,no)
{
cat("Funcao LISTA.VIZINHOS.LIVROS","\n")
n1<-length(no)
v1<-vector()      #Lista de vizinhos 1 do nó
nv1<-vector()   #Tamanho de V1

no<-sort(unique(m1))

for (i in 1:n1)
  {
   v1[i]<-list(m2[m1==no[i]])
   nv1[i]<-length(v1[[i]])
  }
LV<-list(V1=v1,Grau1=nv1)
return(LV)
}
```

20.3.5 Calcula Distância entre Usuários e Livros

```
#############################################################################
#Distancia entre usuarios e livros
#Entrada: Coordenadas dos usuarios (X0),
#       Codigo dos usuarios (vert0),
#       Usuarios vizinhos de cada livro (LVb$V1),
#       Grau1 da lista de livros (LVb$Grau1),
#       Codigo dos livros (LIVRO),
#Saída: Lista de vizinhanca de livros para cada usuario
#----------------------------------------------------------------------
f.distul <- function(X,no,v1,nv1,nob)
{
cat("Funcao VIZINHANCA","\n")
#Calculo das coordenadas dos livros com relacao aos usuarios (X0)
n<-length(nob) #Tamanho da lista de LIVROS
Xb<-matrix(0,n,2)

for (i in 1:n)
  {
  cat("i",i,"livro=",nob[i],"grau=",nv1[i],"\n")

  x<-matrix(0,nv1[i],2)
  for (j in 1:nv1[i])
    x[j,]<-X[no==v1[[i]][j],]

  Xb[i,1]<-mean(x[,1])
  Xb[i,2]<-mean(x[,2])
  cat("Xb=",Xb[i,],"\n")
  }

n<-length(no) #Tamanho da lista de USUARIOS

#Matriz contendo a distancia entre cada usuario e livro
#Cada linha representa um usuario e cada coluna representa um livro,
#ordenado de acordo com os codigos dos usuarios na lista no e dos livros
na lista nob.
distul<-matrix(0,length(no),length(nob))
for (i in 1:n)
  {
  distul[i,]<-as.vector(sqrt((X[i,1]-Xb[,1])^2+(X[i,2]-Xb[,2])^2))
  cat("i:",i," distul=",distul[i,],"\n")
  }
#Para que nao haja probabilidade Na, já que prob é 1/distul na funcao
amostra
distul[distul<0.0000001]<-0.0000001

return(Dist=distul,Xb=Xb)
}
```

20.3.6 Amostra da TO

```
#############################################################################
#Amostra da TO
#Entrada: Distancia entre usuarios e livros (DISTUL)
#       Codigo dos usuarios (vert0),
#       Codigo dos usuarios (LIVRO),
#       Numero de arestas para serem selecionadas de cada usuario (m),
#       Parametro de controle da distancia (alpha)
#Saída: Matriz TO amostrada
#############################################################################

f.amostra <- function(dist,no,nob,m,alpha)
{
cat("Funcao AMOSTRA","\n")
n<-length(no)
#Para que exista a matriz TOa antes de entrar no for
#prob=(1/d^alpha)/soma(1/d^alpha) com d sendo a distancia de 1 usuario
com 1 livro
pr<-(1/dist[1,]^alpha)/sum(1/dist[1,]^alpha)
print(pr)
pr[pr==NA]<-0.0000001
amostra<-sample(nob,m[1],replace=T,pr)
TOa<-matrix(c(rep(no[1],length(amostra)),amostra),length(amostra),2)
for (i in 2:n)
  {
  amostra<-
sample(nob,m[i],replace=F,(1/dist[i,]^alpha)/sum(1/dist[i,]^alpha))
  toa<-matrix(c(rep(no[i],length(amostra)),amostra),length(amostra),2)
  TOa<-rbind(TOa,toa)
  }

TOa<-as.matrix(TOa)
return(TOa)
}
```

20.3.7 Constrói Gráficos

```
############################ Grafico de X0 com legendas###############
plot.leng<-function(X0,vert0)
{
plot(X0,
type="n",
ann=FALSE,
xlim=c(-.6,.6),
ylim=c(-.6,.6)
)
text(X0[,1], X0[,2], vert0, font=2, cex=.5)
}

############# Gráfico em Perspectiva e de Curva de Nivel #############
plot.pc<-function(X0,vert0,USUARIO,Coesao)
{
plot(X0,
ann=FALSE,
xlim=c(-.6,.6),
ylim=c(-.6,.6),
type="n"
)
user<-par("usr")
escolhe<-is.element(USUARIO, setdiff(USUARIO, vert0))
coesao<-Coesao[!escolhe]
X0grid<-f.suaviza.grafico(X0, coesao, user, 2, 20)

#Grafico de Curva de Nivel
contour(X0grid$x, X0grid$y, X0grid$z)

points(X0, pch=16)
win.graph()

#Grafico em Perspectiva
persp(X0grid$x, X0grid$y, X0grid$z,
ann=FALSE,
theta = 30,
phi = 30,
col = "lightblue",
shade= 0.3,
xlab="",
ylab="",
zlab="Coesão",
box=TRUE
)
}
```

20.3.8 Mostra Distribuição de Distância

```
#############################################################################
#Mostra alguns resultados da distancia
#Entrada:
#Saída:
#############################################################################

f.histdist<-function(TS,LIVRO,V1,X0,vert0,alpha,e)
{
#Constroi a lista de vizinhos em um passo (usuarios) dos livros
LVb<-f.lista.vizinhos.livros(TS[,2],TS[,1],LIVRO)
print(LVb)

#Distancia entre livros e usuarios
DISTUL<-f.distul(X0,vert0,LVb$V1,LVb$Grau1,LIVRO)
Xb<-DISTUL$Xb
DISTUL<-DISTUL$Dist

par(mfrow=c(2,3))
for (j in 1:length(e))
  hist(DISTUL[vert0==e[j],],xlab="",main=rbind("Usuário ",e[j]))
}
```

20.3.9 Mostra Distribuição de Probabilidades

```
#############################################################################
#Mostra alguns resultados de probabilidade
#Entrada:
#Saída:
#############################################################################
f.histprob<-function(TS,LIVRO,V1,X0,vert0,alpha,e)
{
#Constroi a lista de vizinhos em um passo (usuarios) dos livros
LVb<-f.lista.vizinhos.livros(TS[,2],TS[,1],LIVRO)
print(LVb)

#Distancia entre livros e usuarios
DISTUL<-f.distul(X0,vert0,LVb$V1,LVb$Grau1,LIVRO)
Xb<-DISTUL$Xb
DISTUL<-DISTUL$Dist

par(mfrow=c(2,3))
for (i in 1:length(alpha))
  {
  pr<-(1/DISTUL[vert0==e,]^alpha[i])/sum(1/DISTUL[vert0==e,]^alpha[i])
  hist(pr,xlab="",main=rbind("alpha ",alpha[i]))
  }

}
```

20.4. Corpo do Programa para Estudo da Estabilidade do Algoritmo RGR

```
#############################################################################
#    CORPO DO PROGRAMA – ESTUDO DA ESTABILIADE DO ALGORITO
#_____________________________________________________________________________
#############################################################################

TO<-as.matrix(read.table("IMEtrans2.dat"))
save(TO, file="TO.r")

SAIDA<-f.alg.chico(TO,1)
#Como o eixo y dessa saida está invertido com a dissertacao, irei fazer -1
SAIDA$X0[,2]<--SAIDA$X0[,2]

#Construcao dos graficos de legenda, perspectiva e contorno
#para as transações originais
plot.leng(SAIDA$X0,SAIDA$vert0)
plot.pc(SAIDA$X0,SAIDA$vert0,SAIDA$USUARIO,SAIDA$Coesao)

#Verificando a variacao da distancia e probabilidade alterando alpha
alpha<-c(0.1,0.5,1,2,10,20)
e<-c(1347,1168,4020,3154,1189,968)
f.histdist(SAIDA$TS,SAIDA$LIVRO,SAIDA$V1,SAIDA$X0,SAIDA$vert0,alpha,e)
j<-1
f.histprob(SAIDA$TS,SAIDA$LIVRO,SAIDA$V1,SAIDA$X0,SAIDA$vert0,alpha,e[j])

repl<-500
alpha<-1 #utilizamos alpha=(0.1,0.5,1,2,10,20)
f.alg.amostra(SAIDA$TS,SAIDA$USUARIO,SAIDA$LIVRO,SAIDA$V1,SAIDA$X0,SAIDA$ve
rt0,repl,alpha)
save(AMOSTRA, file="AMOSTRA.r")


######################### Construção de Gráficos ##########################
#Grafico dos livros e usuarios
plot(SAIDA$X0, ann=FALSE, xlim=c(-.6,.6),ylim=c(-.6,.6),pch=16)
points(AMOSTRA$Xb)
leg<-c("Usuário","Livro")
legend(c(.3,.6),c(.4,.6),leg,pch=c(16,1),cex=1.1,bty="n")


#Grafico dos livros e usuarios com algumas arestas
plot(SAIDA$X0, ann=FALSE, xlim=c(-.6,.6),ylim=c(-.6,.6),pch=16)
text(AMOSTRA$Xb[,1],AMOSTRA$Xb[,2], SAIDA$LIVRO, font=2, cex=.5)
```

```
#Livro: 14241 ligado aos Usuarios: 2872 e 3815
#Livro: 14386 ligado aos Usuarios: 1189, 1691, 1722, 1767, 1789
#Livro: 14282 ligado aos Usuarios: 517 e 1104
lines(c(SAIDA$X0[SAIDA$vert0==1789,1],AMOSTRA$Xb[SAIDA$LIVRO==14386,1]),
      c(SAIDA$X0[SAIDA$vert0==1789,2],AMOSTRA$Xb[SAIDA$LIVRO==14386,2]))
points(AMOSTRA$Xb[SAIDA$LIVRO==14386,1],AMOSTRA$Xb[SAIDA$LIVRO==14386,2],pc
h=3)


#Grafico das legendas dos livros
plot.leng(AMOSTRA$Xb,SAIDA$LIVRO)

#Histograma de R2
par(mfrow=c(2,3))
AMOSTRA<-AMOSTRADec #Amostra com alpha igual a .1
hist(AMOSTRA$R2,xlim=c(0.6,1),ylim=c(0,300),xlab="Estatística
R2",main=rbind("alpha ",alpha[1]))
AMOSTRA<-AMOSTRAMeio #Amostra com alpha igual a .5
hist(AMOSTRA$R2,xlim=c(0.6,1),ylim=c(0,300),xlab="Estatística
R2",main=rbind("alpha ",alpha[2]))
AMOSTRA<-AMOSTRA1 #Amostra com alpha igual a 1
hist(AMOSTRA$R2,xlim=c(0.6,1),ylim=c(0,300),xlab="Estatística
R2",main=rbind("alpha ",alpha[3]))
AMOSTRA<-AMOSTRA2 #Amostra com alpha igual a 2
hist(AMOSTRA$R2,xlim=c(0.2,1),ylim=c(0,300),xlab="Estatística
R2",main=rbind("alpha ",alpha[4]))
AMOSTRA<-AMOSTRA10 #Amostra com alpha igual a 10
hist(AMOSTRA$R2,xlim=c(0.2,1),ylim=c(0,300),xlab="Estatística
R2",main=rbind("alpha ",alpha[5]))
AMOSTRA<-AMOSTRA20 #Amostra com alpha igual a 20
hist(AMOSTRA$R2,xlim=c(0.2,1),ylim=c(0,300),xlab="Estatística
R2",main=rbind("alpha ",alpha[6]))


####################Grafico do Passeio#######################################
#Grafico do Passeio do vertice nas amostras em que foi sorteado
e<-c(1347,1168,4020,3154,1189,968)
j<-1
plot(SAIDA$X0, ann=FALSE, xlim=c(-1,1),ylim=c(-1,1),pch=1)
points(SAIDA$X0[SAIDA$vert0==e[j],1],SAIDA$X0[SAIDA$vert0==e[j],2],pch=16)
for (i in 1:repl)
  if (any(AMOSTRA$vert0a[[i]]==e[j])==T)
    {
    x<-AMOSTRA$X0a[[i]][AMOSTRA$vert0a[[i]]==e[j],1]
```

```
    y<-AMOSTRA$X0a[[i]][AMOSTRA$vert0a[[i]]==e[j],2]
    points(x,y,pch=3)
    }

#Grafico da Coesao versus o eixo x
plot(cbind(SAIDA$X0[SAIDA$vert0==e[j],1],SAIDA$Coesao[SAIDA$USUARIO==e[j]])
,
     ann=FALSE, xlim=c(-1,1),ylim=c(0,1),pch=3)
abline(SAIDA$Coesao[SAIDA$USUARIO==e[j]],0)

for (i in 1:repl)
  if (any(AMOSTRA$vert0a[[i]]==e[j])==T)
    {
    x<-AMOSTRA$X0a[[i]][AMOSTRA$vert0a[[i]]==e[j],1]
    y<-AMOSTRA$Coesaoa[[i]][AMOSTRA$USUARIOa[[i]]==e[j]]
    points(x,y,pch=3)
    }

#Grafico da Coesao versus o eixo y
plot(cbind(SAIDA$X0[SAIDA$vert0==e[j],2],SAIDA$Coesao[SAIDA$USUARIO==e[j]])
,
     ann=FALSE, xlim=c(-1,1),ylim=c(0,1),pch=3)
abline(SAIDA$Coesao[SAIDA$USUARIO==e[j]],0)

for (i in 1:repl)
  if (any(AMOSTRA$vert0a[[i]]==e[j])==T)
    {
    x<-AMOSTRA$X0a[[i]][AMOSTRA$vert0a[[i]]==e[j],2]
    y<-AMOSTRA$Coesaoa[[i]][AMOSTRA$USUARIOa[[i]]==e[j]]
    points(x,y,pch=3)
    }


############Grafico de contorno e perspectiva para 9 replicas############
e<-sample(1:500,9,F)
cat("Replicas escolhidas= ",e,"\n")

AMOSTRA<-AMOSTRA20
j<-e[i]
plot.pc(AMOSTRA$X0a[[j]],AMOSTRA$vert0a[[j]],AMOSTRA$USUARIOa[[j]],AMOSTRA$
Coesaoa[[j]])
```

21. BIBLIOGRAFIA

AHO, Alfred V; HOPCROFT, John E; e ULLMAN, Jeffrey D. **The design and Analysis of Computer Algorithms**. Reading: Addison-Wesley Publishing Company, 1974

ARANHA, Francisco. *"E-Service em Bibliotecas: Geração de Valor para Pesquisadores por Meio de Cooperação Indireta"*. **RAE – Revista de Administração de Empresas,** São Paulo: FGV/EAESP, v. 40, n. 4, Out/Dez 2000, pp. 84-93.

ARANHA, Francisco. **Perfil de Usuários da Biblioteca Karl A. Boedecker:** Geração de Valor para Pesquisadores por Meio de Cooperação Indireta. São Paulo: FGV/EAESP/NPP, Relatório de Pesquisa no. 14, 2001a, 64 pp.

ARANHA, Francisco. **Análise de Redes em Procedimentos de Cooperação Indireta: Utilização no Sistema de Recomendações da Biblioteca Karl A. Boedecker**. São Paulo: FGV/EAESP/NPP, Relatório de Pesquisa no. 27, 2001b, 77 pp.

BALAKRISHNAN, V. K. **Theory and Problems in Graph Theory**. New York: McGraw-Hill, 1997, 293 pp.

BARABASI, Albert-Laszlo. **Linked: The New Science of Networks**. Cambridge (MA): Perseus Publishing, 2002, 280pp.

BATTISTA, Giuseppe di; EADES, Peter; TAMASSIA, Roberto; e TOLLIS, Ioannis G. **Graph Drawing: Algorithms for the Visualization of Graphs.** Upper Saddle River: Prentice Hall, 1999, 397pp.

BEATTY, Sharon E. e SMITH, Scott M. *"External Search Effort: An Investigation Across Several Product Categories"*. **Journal of Consumer Research,** Gainesville, v. 14, p. 83-95, June 1987.

BERRY, Michael e LINOFF, Gordon. **Data Mining Techniques For Marketing, Sales and Customer Support.** New York: John Wiley and Sons, 1997, 454 pp.

BERRY, Michael e LINOFF, Gordon. **Mastering Data Mining: The Art and Science of Customer Relationship Management**. New York: John Wiley and Sons, 2000, 494 pp.

BOLLOBÁS, Béla. **Random Graphs**. London; Orlando: Academic Press, 1985, 447 pp.

BOLLOBÁS, Béla. **Modern Graph Theory**. New York: Springer, 1998, 394 pp.

BUSSAB, Wilton O; MIAZAKI, Édina S. e ANDRADE, Dalton F. **Introdução à Análise de Agrupamentos**. São Paulo: IME/USP, 1990, 105 pp.

CABENA, Peter; HADJINIAN, Pablo; STADLER, Rolf; VERHEES, Jaap; e ZANASI, Alessandro. **Discovering Data Mining**, Upper Saddle River: Prentice Hall, 1997, 195 pp.

CARSON, Kerry D; CARSON, Paula P; e PHILIPS, Joyce S. **The ABCs of Collaborative Change:** The Manager's Guide to Library Renewal. Chicago: ALA Editions, 1997, 272 pp.

CLUNG, Fan e GRAHAM, Ron. **Erdos on Graphs: His Legacy and Unsolved Problems**. Wellesley, Mass: A. K. Peters, 1998.

COX, Trevor F. e COX, Michael A. **Multidimensional Scaling**, 2nd ed. Boca Raton: Chapman & Hall/CRC, 2001, 308 pp.

GLADWELL, Malcom. **The Tipping Point**. London: Abacus, 2000, 279 pp.

GOOD, Nathaniel, SCHAFER, J. Ben e **OUTROS**. **Combining Collaborative Filtering with Personal Agents for Better Recommendations**. Minneapolis: GroupLens Research Project, Department of Computer Science and Engineering, University of Minnesota. http://www.cs.umn.edu/research/grouplens/aaai-99.pdf (link válido em 23/07/01), 7 pp.

GRAHAM, R. L. e NESETRIL, J. **The Mathematics of Paul Erdos**. Berlin: Springer, 1997.

GRANOVETTER, Mark S. *The Stregth of Weak Ties.* **American Journal of Sociology**, Volume 78, Issue 6, May 1973, pp. 1360-1380.

HAIR, Joseph; ANDERSON, Rolph; TATHAM, Ronald; e BLACK, William. **Multivariate Data Analysis**, 4th ed., Englewood Cliffs: Prentice Hall, 1995, 745pp.

HANN, Leslie. *"High-Tech Sleuths"*. **Best's Review**, Nov. 1998, pp. 83-85.

JENSEN, David. *"Prospective Assessment of AI Technologies for Fraud Detection"*. **Working Papers of the AAAI-99 Workshop on Artificial Intelligence Approaches to Fraud Detection and Risk Management**.

JOHNSON, Richard A. e WICHERN, Dean W. **Applied Multivariate Statistical Analysis**, 5th ed. Englewood Cliffs: Prentice Hall, 2002.

KAUTZ, Henry; SELMAN, Bart; e SHAH, Mehul. *"Combining Social Networks and Collaborative Filtering"*, **Communications of the ACM**, March 1997a, vol. 40, n. 3, pp. 63-65.

KAUTZ, Henry; SELMAN, Bart; e SHAH, Mehul. *"The Hidden Web"*, **AI Magazine**, Summer 1997b, pp. 27-36. Disponível em http://www.research.att.com/~kautz/referralweb/doc/aimag.pdf, *link* válido em 27/12/99 às 11h10).

KNOKE, David e KUKLINKSI, James H. **Network Analysis**, Newbury Park: Sage Publications, Sage University Paper 28, 1982, 96 pp.

LARGE, Peter. **The Micro Revolution Revisited.** New Jersey: Rowman & Allanheld, 1984.

MIDGLEY, David F. *"Patterns of Interpersonal Information Seeking for the Purchase of a Symbolic Product"*. **Journal of Marketing Research**, Chicago, v. 20, p. 74-83, Feb. 1983.

MILLER, Kathy. *"Librarians Getting a Handle on Knowledge"*. **Information World Review**, June 1998, pp. 25-26.

NEWMAN, Joseph W. e STAELIN, Richard. *"Information Sources of Durable Goods"*. **Journal of Advertising Research**, New York, v. 13, p. 19-29, Apr. 1972

PAYTON, David. *"Discovering Collaborators by Analyzing Trails Trough an Information Space"*, **Artificial Intelligence and Link Analysis Papers from the 1998 Fall Symposium**, October 23-25, Orlando Florida.

PEPPERS, Don e ROGERS, Martha. *"Is Your Company Ready for One-to-One Marketing?"*. **HBR,** January-February 1999, pp. 151-163.

RAMO, Joshua C. *"The Fast-Moving Internet Economy Has A Jungle of Competitors... And Here Is the King: Jeffery Preston Bezos, 1999 Person of the Year"*, **TIME Magazine**, 27/12/1999, pp. 42-59.

ROGERS, Everett M. **Diffusion od Innovations**, 4th ed. New York: The Free Press, 1995, 518 pp.

SANTOS, Érico R. **Implantação de Tecnologia de Data Warehouse em Bibliotecas com Uso de Tecnologia Adequada.** São Paulo: FGV/EAESP, 2000, 53 pp.

SCHWARTZ, Michael e WOOD, David. *"Discovering Shared Interests Using Graph Analysis"*. **Communications of the ACM**, August 1993, vol. 36, n. 8, pp. 78-89.

SKIENA, Steven S. **The Algorithm Design Manual**. New York: Springer-Verlag, 1998, 486 pp.

SMYTH, Barry e COTTER, Paul. *"A Personalized Television Listings Service"*. **Communications of the ACM**. New York: Association for Computing Machinery, v. 43, n. 8, Aug 2000, pp. 107-111.

SWANSON, Don e SMALHEISER, David. *"Link Analysis of Medline Titles as an Aid to Scientific Discovery"*. http://kiwi.uchicago.edu/ libtrends.html, *link* válido em 04.01.2000.

TRUDEAU, Richard J. **Introduction to Graph Theory**. New York: Dover, 1993, 203 pp.

WASSERMAN, Stanley e FAUST, Katherine. **Social Network Analysis: Methods and Applications.** Cambridge: Cambridge University Press, 1994, 825 pp.

WASSERMAN, Stanley e GALASKIEWCZ, Joseph (eds.). **Advances in Social Network Analysis,** Thousand Oaks: Sage Publications, 1994, 299 pp.

WATTS, Duncan J. **Small Worlds: The Dynamics of Networks between Order and Randomness**. Princeton: Princeton University Press, 1999, 262 pp.

WILSON, Robin J. **Introduction to Graph Theory**. Harlow: Prentice Hall, 1996, 171.

WURMAN, Richard Saul. **Ansiedade de Informação.** São Paulo: Cultura, 1991, 380 pp.

A

acessibilidade de documentos na Internet, 23
AHO, HOPCROFT e ULLMAN, 20
Amazon.com, 26
análise de agrupamentos, 15
análise de cestas, 15
análise de redes, 15, 18, 22, 23
ARANHA, 14, 15, 16
Arrowsmith, 12

B

BALAKRISHNAN, 21
BARABASI, 21, 26
BEATTY e SMITH, 9
BERRY e LINOFF, 15
Biblioteca Karl A. Boedecker, 14, 146
bibliotecas, 14
BOLLOBÁS, 21
BUSSAB, MIAZAKI e ANDRADE, 15

C

CABENA e OUTROS, 12
CARSON e OUTROS, 14
CLUNG e GRAHAM, 21
coesão, 18, 22
colaboradores potenciais, 12
collaborative filtering, 9, 148
cooperação indireta, 23

D

data mart de transações, 14
detecção de fraudes, 12
difusão de inovações, 23
distância temática, 16

E

escalabilidade de algoritmos, 17, 20
escalonamento multidimensional, 18
espaço temático, 18, 19

F

filtro colaborativo, 9, 11, 12, 15, 18
filtro de conteúdo, 10
filtro informativo, 9, 10, 11, 12
formação de subgrupos especializados, 15

G

GLADWELL, 24
GOOD, SCHAFER e OUTROS, 9
GRAHAM e NESETRIL, 21
GRANOVETTER, 24, 25

H

HAIR, TATHAM e OUTROS, 15
HANN, 13

I

identificação de assuntos significativos, 15
information filtering, 9
information retrieval, 9

J

JENSEN, 13
JOHNSON e WICHERN, 15

K

KAUTZ, SELMAN e SHAH, 12
KNOKE e KUKLINSKY, 22

L

LARGE, 8
LAWRENCE e GILES, 26
livraria virtual, 17, 26

M

matriz de co-ocorrências, 16
MIDGLEY, 8
MILLER, 14
Movie Lens, 27

N

NEWMAN e STAELIN, 8

P

padrão de navegação, 12
PAYTON, 12
PEPPERS e ROGERS, 14
período de latência, 12
Personalized Television, 27
procura de empregos, 23

R

RAMO, 26
recuperação de informação, 9
Redes, 21
relevância de um item, 11
relevância localizada, 9
ROGERS, 23, 24

S

SANTOS, 14
SCHWARTZ e WOOD, 12, 15, 22
sistema de recomendações, 14, 15, 16, 17, 27
sistemas de recomendações, 26
SMYTH e COTTER, 10, 27
subgrupos especializados, 15, 16, 18
SWANSON e SMALHEISER, 12

T

teoria dos grafos, 18, 21, 23
TRUDEAU, 21

V

vizinhança, 18, 22

W

WASSERMAN e FAUST, 22
WASSERMAN e GALASKEWICZ, 23
WATTS, 22
WILSON, 21
WURMAN, 8