Um estudo sobre métodos de *kernel* para classificação e agrupamento de dados

Francisco Alberto de Andrade Queiroz

Um estudo sobre métodos de *kernel* para classificação e agrupamento de dados

Francisco Alberto de Andrade Queiroz

Orientador: *Prof Dr Antônio de Pádua Braga*

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da UFMG - PPGEE UFMG, como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica

UFMG - Belo Horizonte
Agosto/2009

Agradecimentos

Agradeço a Deus por dar-me as capacidades para viver e para aprender. Aos meus pais, Sérgio e Cecília, e ao meu irmão Sérgio, meu agradecimento especial pela compreensão, carinho e suporte.

Sou grato ao Professor Antônio Braga pelas inúmeras vezes em que a mim confiou desafios, tarefas e a amizade. Inúmeras foram as vezes em que a sua ajuda paciente e entusiasmada me mostraram caminhos novos. Agradeço também ao Professor Eduardo Mazoni pela co-orientação no projeto CEMIG-Demanda e pela convivência bem humorada e atenciosa. Não posso deixar de agradecer aos amigos, colegas e professores do Laboratório de Inteligência Computacional (LITC) e do Centro de Desenvolvimento e Pesquisa em Engenharia Elétrica (CPDEE): André, Augusto, Cidiney, Cristiano, Daniel, Euler, Frederico, Guilherme, Illya, Levi, Lucas, Luciane, Manoel, Marcelo, Mendonça, Mozelli, Naísses, Rafael, Talles, Thiago, Wilian e tantos outros com quem a convivência foi um grande presente. Também aos amigos e colegas do Laboratório de Materiais e Pilhas a Combustível (LaMPaC) com quem a convivência breve foi intensa e insubstituível.

Agradeço também aos cursos de graduação e mestrado em Engenharia Elétrica que proporcionaram grande transformação durante todo o tempo em que permaneci na UFMG. À Companhia Energética de Minas Gerais (CEMIG), em especial à equipe envolvida no Projeto de Previsão de Demanda de Energia Elétrica, o meu muito obrigado por permitir ao mesmo tempo aprender, pesquisar e participar de um projeto desafiador.

Aos meus amigos do Promove que torceram por mim e com quem dividi várias emoções deste trabalho.

A todas as pessoas que, direta ou indiretamente, contribuíram para a realização desta dissertação.

"Fazer previsões é difícil, especialmente sobre o futuro."

Niels Bohr

À minha família.

Resumo

O projeto de máquinas de aprendizagem envolve a modelagem de um conjunto de amostras tendo como base o desempenho do mapeamento dos pares entrada-saída. O grupo de amostras reservado ao treinamento fornece informações para a determinação dos parâmetros do modelo. E o grupo de validação e/ou de teste avalia o desempenho do classificador quanto à sua capacidade de generalização. Entretanto, o classificador obtido ao final desse processo na maioria dos casos não incorpora as relações de similaridade entre as amostras e as classes. Essa metodologia, portanto, resulta em uma modelagem incompleta das informações contidas nos dados.

Neste trabalho, procura-se lidar ao mesmo tempo com o problema básico da análise de dados e com o projeto de máquinas de núcleo (ou de *kernel*): determinar o número de grupos em um conjunto de amostras e os parâmetros da função de núcleo escolhida. Para tanto, utiliza-se como métrica o Alinhamento Empírico para determinar a similaridade entre a matriz de núcleo e a de proximidade resultante do C-Médias Nebuloso (FCM).

Mostra-se que a métrica escolhida pode ser maximizada em função dos parâmetros do FCM e da função do núcleo. O alinhamento é maior quanto maior é a coerência entre a informação estrutural embutida dos dados nas duas matrizes. No entanto, a determinação dos parâmetros não é possível por métodos diretos de ajuste. Sendo assim, o Algoritmo Genético e a Otimização por Enxames de Partículas são os métodos evolutivos escolhidos para encontrar aproximações dos parâmetros que, ao resolverem o problema de otimização mono-objetivo formulado, maximizem a métrica escolhida. Os parâmetros obtidos são utilizados em Máquinas de Vetor de Suporte por Mínimos Quadrados (LS-SVMs) segundo a metodologia aqui proposta para o projeto de classificadores. Utilizando os métodos Minus e de autovetor para ordenação das amostras nas matrizes referidas, é possível observar a similaridade entre indivíduos de cada um dos grupos e obter outras informações que auxiliam na caracterização desses últimos.

Por meio de experimentos com bases de dados de teste e de referência, obtêm-se resultados que corroboram a escolha da métrica e dos métodos utilizados nas referidas bases para agrupamento e classificação binária. Além disso, no âmbito do problema citado inicialmente, as observações fornecidas suscitam maior conhecimento sobre as relações e os métodos empregados, permitindo a utilização apropriada da informação estrutural dos dados.

Palavras-chave: Agrupamento, Algoritmo Genético, C-Médias Nebuloso, Classificação, Função de Base Radial, matriz de Afinidade, matriz de Núcleo, Máquina de Vetor de Suporte por Mínimos Quadrados, Ordenação, Otimização por Enxames de Partículas, Reordenação.

Abstract

The learning machines project involves modelling a set of samples based on the mapping performance of the input-output pairs. The group of samples submitted to training provides information for determining the parameters of the model. And the validation and/or test group evaluates the performance of the classifier on its generalization ability. However, the classifier obtained at the end of this process in most cases does not embody the relationship of similarity between samples and classes. This approach therefore results in an incomplete modelling of the information provided by the data.

In this work, we deal simultaneously with the basic problem of data analysis and of the project of kernel learning machines: the number of groups in a set of samples and the chosen parameters of the core function. For both, the metric used is the Empirical Alignment to determine similarity between the kernel and the proximity matrix of Fuzzy C-Means (FCM).

It is shown that the metric chosen can be maximized depending on the parameters of the FCM and of the core function. The greater the consistency between the structural information embedded in the two data matrices, the higher is the alignment. However, the determination of parameters is not possible by direct adjustment methods. Thus, by solving the problem of mono-objective optimization formulated, the Genetic Algorithm and the Particle Swarm Optimization are the evolutionary methods chosen to find approximations of the parameters which maximize the chosen metric. The parameters obtained are used in Least Square Support Vector Machines (LS-SVMs) according to the methodology proposed here for designing classifiers. Using the eigenvector and Minus methods for ordering the samples in these matrices, it is possible to observe the similarity between individuals of each group and additional information to help characterize the latter.

Through experiments using test and reference databases, the results obtained here corroborate the metric and the methods used in these databases for binary classification and clustering. Moreover, under the initially afore-

mentioned problem, the provided observations raise greater awareness about the relationships and the methods employed, allowing for better use of the structural information of the data.

Keywords: Affinity matrix, Classification, Clustering, Fuzzy C-Means, Genetic Algorithm, Kernel matrix, Reordering, Sorting, Least Square Support Vector Machine, Particle Swarm Optimization, Radial Basis Function.

Sumário

Lista de Abreviaturas

Abreviatura	**Significado**
AG	Algoritmo Genético
BD	Melhor Decisão (Best Decision)
FCM	C-Médias Nebuloso(Fuzzy C-Means ou Fuzzy ISODATA)
FPM	Matriz de proximidade
FT	Frequência de Transformação
LS-SVM	Máquina de Vetor de Suporte por Mínimos Quadrados (Least Square Support Vector Machine)
IDH	Índice de Desenvolvimento Humano
IMRS	Índice Mineiro de Responsabilidade Social
KKT	Sistema de Karush-Kuhn-Tucker
MLP	Rede Neural Artificial do tipo perceptron multi-camadas (multilayer perceptron)
NP	Não deterministicamente polinomial
PIB	Produto Interno Bruto
PSO	Otimização por Enxames de Partículas (Particle Swarm Optimization)
QAP	Problema de Associação Quadrática
QP	Problema Quadrático (Quadratic Problem)
RBF	Função de Base Radial (Radial Basis Function)
SEPLAG	Secretaria de Planejamento e Gestão
SVM	Máquina de Vetor de Suporte (Support Vector Machine)
STS	Lado a Lado (Side-to-Side)
VC	Vapnik-Chervonenkis (dimensão de)

Lista de Símbolos

Símbolo	**Significado**
σ	Raio da função Gaussiana
γ ou C	Parâmetro de folga da margem
μ	Centro da função Gaussiana
$\varphi(\cdot)$	Função de transformação ou mapeamento de dimensões
w	Peso ou vetor de pesos
acc	Acurácia
N	Número de amostras
b	Termo de polarização
c	Número de grupos ou partições
n	Número de características ou atributos das bases de dados
A	Alinhamento
K	Matriz de *kernel*
U	Matriz de partição
m	Coeficiente ou fator de *fuzziness*
P	Matriz de proximidade
S	Matriz de afinidade
λ_k	k-ésimo autovalor de uma matriz quadrada
nVP	Número de verdadeiros positivos
nVN	Número de verdadeiros negativos
nFP	Número de falsos positivos
nFN	Número de falsos negativos
N_{CV}	Número de amostras utilizados no processo de sintonia por validação cruzada
N_{test}	Número de amostras no conjunto de teste
n_{num}	Número de atributos numéricos
n_{cat}	Número de atributos categóricos
n_{gen}	Número máximo permitido de gerações
n_{ind}	Número de indivíduos por população
$variavel_{\mu}$	Valor médio da variável
$variavel_{Md}$	Valor mediano da variável
$\mathbf{x}^T$	Transposto do vetor $\mathbf{x}$, representado pelo índice T
$\langle \mathbf{x} \rangle_F$	Norma de Frobenius (ou norma de Hilbert-Schmidt) do vetor $\mathbf{x}$

Lista de Figuras

Lista de Tabelas

CAPÍTULO

1

Introdução

A classificação é um dos desafios mais frequentemente presentes nas tarefas de tomada de decisão em que se baseia a atividade humana [49]. Apesar de parecer um problema corriqueiro, demanda grande esforço e conhecimento para estabelecer categorias determinando a subdivisão de conjuntos de amostras. Principalmente quando a solução deste problema se baseia na informação disponível em um conjunto reduzido (mas significativo) de amostras multidimensionais (caracterizados por muitos atributos). Tais condições são comuns em casos reais abordados como problemas de classificação em ciência, na indústria e em negócios [18]. Esforço adicional é necessário se o objetivo é obter um modelo de consenso para diferentes classificadores aplicados a um mesmo conjunto de amostras. Vigor semelhante é requerido para a obtenção de partições de amostras e de estruturas de consenso para diferentes agrupamentos [39]. A formulação detalhada destes problemas bem como a discussão das peculiaridades de cada um podem ser encontradas em [29] e em [39].

Adaptando as definições apresentadas em [29], os problemas tratados aqui envolvem dados $x = \{a_1, \ldots, a_d, y\}$ compostos por d atributos (ou características). Deste modo, a dimensionalidade do espaço de representação original da coleção de n dados do conjunto $X = \Gamma_u = \{x_i, \ldots, x_n\}$ é também d. A matriz de padrões $n \times d$ tem associada uma coluna de rótulos Y que determina uma classe $y \in [-1, +1]$ para cada amostra. A classe é definida por uma função geradora de amostras desconhecida a que está associada uma densidade de probabilidade no espaço de características. Os métodos de agrupamento tentam reunir os padrões de tal forma que os grupos obtidos reflitam os diferentes processos de geração de amostras representados no conjunto de padrões. E os

métodos de classificação tentam separar as classes de tal forma a ter grupos coerentes formados por padrões de mesmo rótulo. Nesse contexto, a medida de distância (ou proximidade) é uma métrica (ou quase-métrica) no espaço de características para quantificar a similaridade dos padrões.

A classificação se difere claramente do agrupamento pela definição de cada problema (Jain et al. [29]). Denomina-se agrupamento (*clustering*) a classificação não supervisionada de padrões (observações, itens, pontos num espaço multidimensional ou vetores de atributos ou de medidas) em grupos (*clusters*). Para esses, o problema consiste em agrupar em subconjuntos significativos uma dada coleção de padrões não rotulados. Os rótulos estão associados também aos subconjuntos, mas são obtidos somente dos dados. Por sua vez, a análise discriminante é a classificação supervisionada de um conjunto de padrões. Neste caso, a coleção de padrões rotulados (pré-classificados) serve de base para a classificação de uma ou mais amostras ainda sem rotulação. Ou seja, a partir do conjunto de treinamento, um modelo aprende a descrição das classes para então rotular um novo padrão.

O objetivo do uso de métodos de agrupamento é a obtenção de uma "abstração dos dados": uma representação simples e compacta do conjunto de amostras (Jain et al. [29]). Tanto as máquinas quanto os humanos se beneficiam desta representação seja no processamento eficiente, seja na compreensão da estrutura nos dados. Ainda segundo Jain et al., o agrupamento é um processo subjetivo cuja solução possível reflete o conhecimento que se tem sobre os dados. O resultado deve atender a uma aplicação definida previamente. E justamente por isso, agrupar não é uma tarefa simples e não possui um algoritmo de uso geral.

As máquinas têm desempenho menor ou igual ao dos humanos quando se trata da análise de amostras com uma ou duas dimensões. Entretanto, os problemas reais frequentemente envolvem muitas dimensões, situação esta em que as máquinas conseguem acessar mais eficientemente a estrutura embutida nos conjuntos de dados. E para revelar a estrutura característica dos diferentes conjuntos de dados, há grande número de métodos e estratégias como apresentado em [29] e ao longo deste trabalho. As metodologias de agrupamento recebem diversas nomenclaturas, terminologias e suposições nas diversas áreas em que encontram aplicação. Em todas elas, os agrupamentos possibilitam a exploração das inter-relações das amostras através da representação da estrutura dos dados conforme o método escolhido. A observação destas representações quando possível, é avaliada internamente, externamente ou de forma relativa em relação aos métodos utilizados e ao conhecimento *a priori*. A estrutura verdadeira dos dados se torna cada vez mais a acessível quanto mais informações o especialista obtiver.

Seguindo a intuição comum aos humanos [8], muitos métodos se baseiam na similaridade para realizar a partição dos dados em grupos. Busca-se a melhor partição com a finalidade de que a similaridade seja maior dentro e não entre os grupos de amostras. Ou ainda, que a partição final resulte em grupos mutuamente isolados mas com satisfatória "coesão" interna [34]. Há muitas métricas que expressam a similaridade entre dois pontos (ou vetores de entrada) [8], sendo que a maioria é sensível à distribuição espacial dos dados e à faixa de valores encontrados nos vetores de entrada. Por esse motivo, frequentemente os dados devem passar por algum tipo de pré-processamento genericamente denominado de normalização [16] ou padronização [34]. Essa etapa inicial tem como objetivo apenas alterar a faixa de valores em cada atributo, ficando a cargo de etapas posteriores, quando necessário, a alteração das distribuições. Dentre os tipos mais comuns de normalização estão:

- A divisão dos atributos pelos respectivos valores máximos (ou mínimos ou ainda máximos absolutos) verificados de modo a ter novos valores apenas entre 0 e 1. É referida também como normalização para o interior do hipercubo unitário [30];
- A divisão dos atributos por suas respectivas normas como, por exemplo, as normas Euclidiana, Máxima, de Manhattan, infinita, etc. [16];
- A subtração do valor médio de cada atributo seguida da divisão dos atributos pelos desvios padrão respectivos [34].

Além da padronização, há a possibilidade de avaliar a semelhança entre as amostras utilizando métricas apropriadas para o cálculo da distância de modo coerente com a diferença de magnitude e de distribuições entre os valores dos atributos, ou ainda, o tipo desses últimos. Igual atenção deve ser dada à presença de valores discrepantes (*outliers*) e correlação entre as variáveis. Por esses argumentos, é sugerido que se faça uma análise inicial tão abrangente quanto possível do conjunto de amostras para identificar as providências a serem tomadas e possíveis condições de falha dos métodos [34]. Pode-se tentar identificar, por exemplo, a possibilidade de redução do número de dimensões no espaço de entrada ou ainda a sensibilidade em relação aos valores iniciais dos métodos [22].

O extenso número de aplicações dos algoritmos de agrupamento (ou *clustering*) dá-se não apenas para organizar e categorizar dados, mas também com a finalidade de compressão de dados e construção de modelos [30]. São grandes áreas de aplicação dos métodos de agrupamento: a análise de padrões, a tomada de decisão, a aprendizagem de máquina entre outras. Todavia, não há uma solução exata para o problema da escolha do número c de *clusters*.

Alguns critérios para auxiliar a definição de c são detalhados em [34]. Ainda segundo essa referência, nos métodos não hierárquicos, que têm como objetivo obter diretamente uma partição do conjunto de dados desde que previamente informado o número de divisões, grupos novos podem ser criados a cada iteração a partir de amostras que não necessariamente constituíam o mesmo grupo em partições anteriores para um mesmo c. E pelo fato de serem métodos iterativos, os métodos não hierárquicos permitem a análise de grandes conjuntos de amostras.

Para tentar cumprir a sua tarefa, um método de agrupamento tem como grande desafio definir a representação apropriada dos padrões e avaliar se é vantajoso selecionar (identificar um subconjunto dos originais) ou extrair (computar novos a partir dos originais) atributos. Para tanto é preciso estar atento ao custo computacional destes procedimentos, que é função do número de dados e de dimensões. Outras estratégias são possíveis como, por exemplo, utilizar apenas as amostras completamente caracterizadas. Mas o resultado obtido sempre reflete cada uma das etapas de processamento. Assim sendo, apenas a partir do estabelecimento do que se deseja atingir deve-se decidir entre diversas opções e métodos para que o resultado atenda às expectativas.

Em Jain et al. [29], a taxonomia das técnicas de agrupamento divide inicialmente os grupos em hierárquicos e particionais. Dentro do segundo grupo estão contidos os métodos baseados em erro quadrático, em gráfico teórico, em misturas e segundo o modo de busca. A construção de uma árvore dos métodos pode se basear nas seguintes características:

- A estrutura e a operação dos algoritmos: aglomerativos ou divisivos;
- O uso sequencial ou simultâneo das características dos dados;
- A separação rígida ou nebulosa das classes;
- A natureza determinística ou estocástica dos métodos;
- O uso incremental ou não dos dados conforme o seu número total.

Os métodos citados neste trabalho têm recebido grande atenção em áreas em que a computação é vista como ferramenta indispensável para tratar grande volume de amostras e informações. Algumas das aplicações que mais se destacam atualmente estão no setor de bioinformática [42, 46, 11], para internet [39, 33], em estudos socioeconômicos [45, 23, 34] e em análise de imagens [47]. Tanto quanto possível, a notação adotada neste trabalho segue a utilizada pelas principais referências pesquisadas.

1.1 O problema do aprendizado

A composição de um modelo que aprenda a partir de exemplos, pode ser definida simplificadamente pela existência de três componentes básicas [43]:

1. Um gerador de vetores **x** aleatórios, gerados de forma independente de uma distribuição fixa desconhecida;
2. Um supervisor que retorna um vetor de saída **y** para cada vetor de entrada **x** de acordo com uma função de distribuição condicional também fixa mas desconhecida;
3. Uma máquina de aprendizagem capaz de implementar um grupo de funções.

Ainda segundo Vapnik [43], o problema do aprendizado consiste em escolher de um grupo de funções a que prediz a resposta do supervisor da melhor forma possível. Ou seja, encontrar a função que minimize a probabilidade de erros de classificação quando os pares de entrada-saída são fornecidos mas a medida de probabilidade é desconhecida. Posto desta forma, este problema de aprendizado é um caso particular do problema geral de minimização do risco funcional com base em dados empíricos.

1.2 Utilização da estrutura dos dados no projeto de classificadores

Análises não supervisionadas de agrupamento são apropriadas para a exploração da estrutura inerente dos dados. Tais análises se destinariam a particionar um conjunto de amostras não rotuladas em grupos que sejam significativos. Sobre estes apontamentos de Jain et al. em [29], Cai et al. [7] argumenta que esse agrupamento não supervisionado não pode ser diretamente aplicado à classificação porque os rótulos das classes não são utilizados na partição. Além disso, apesar de métodos de agrupamento poderem ser usados para classificação, não há garantia de que os grupos receberão o rótulo de apenas um classe uma vez que esses últimos podem ser constituídos por amostras de mais de uma classe. Assim, há uma sugestão de que as abordagens de agrupamento não supervisionado e de classificação supervisionada são mais susceptíveis a se complementarem pela sua integração [7].

Os métodos de treinamento de classificadores em que métodos de agrupamento são utilizados normalmente enfatizam o desempenho do classificador segundo várias métricas e descartam as informações dos dados com relação

à estruturas entre as amostras. O presente trabalho procura enfatizar a informação estrutural valorizando os métodos de agrupamento, aos quais são dados com frequência papeis auxiliares. Para tanto, os parâmetros dos classificadores são induzidos pelas relações entre as amostras. E por consequência, os valores induzidos dependem do método de *clustering* e da função de transformação aplicados aos dados. É preciso, portanto, estar atento à robustez do método e da função quanto a dados discrepantes e a ruído [7]. Daí a importância de submeter os métodos a diferentes bases de referência e comparar os seus desempenhos com os classificadores mais bem sucedidos reportados na literatura.

1.3 Organização dos tópicos abordados

O texto está dividido em cinco capítulos. O problema geral de análise e classificação de amostras é exposto no capítulo inicial. O segundo capítulo trata das definições principais e dos métodos relacionados. A metodologia utilizada nos experimentos é assunto da terceiro capítulo, onde são detalhadas as alterações nos métodos para atender às condições que se apresentaram. No quarto capítulo, os experimentos e seus resultados são expostos e analisados detalhadamente. O último capítulo traz as considerações e as conclusões gerais, além de apontar sugestões para trabalhos futuros. O texto termina com a listagem das referências consultadas.

1.4 Conclusões do capítulo

A breve introdução da área de agrupamento e classificação de dados permite um contato inicial com o vasto número de aplicações existentes e por desenvolver. Por meio da introdução de alguns métodos e suas definições ao longo dos próximos capítulos, ficará melhor delineado o campo em que se concentra o presente trabalho.

CAPÍTULO

2

Definições Principais

2.1 Introdução

Várias são as abordagens existentes e os seus métodos associados para revelar a estrutura de um conjunto de dados. Neste capítulo, a revisão dos métodos apresenta algumas das muitas abordagens possíveis para a construção de classificadores que incorporam informações sobre as estruturas do dados no espaço de entrada.

Deve-se entender por estrutura dos dados a noção registrada em [7], segundo a qual há localizações relativas das amostras em um espaço de alta dimensão. Estas localizações agregam transparência aos resultados de classificação melhorando a sua interpretação.

Conforme será exposto, a notação e os termos adotados se assemelham entre os métodos. A diferença da formulação dessas metodologias se dá de acordo com as características das bases de dados tais como:

- A área de aplicação dos dados;
- O tipo de codificação (categórica ou numérica - real ou inteira);
- O número total de amostras;
- A distribuição espacial dos grupos e classes;
- O armazenamento (local ou distribuído);
- A estrutura de interesse (na maioria dos casos, o interesse é pela similaridade entre amostras).

Portanto, para entender as peculiaridades e as estratégias de cada método, é necessário apresentar as definições de cada um. Antes porém, como referência para o entendimento dos objetivos dos métodos e desse trabalho, é necessário apresentar o problema motivador.

2.2 Caracterização do problema

Muito frequentemente um grupo de especialistas não dispõem de informações suficientes sobre um conjunto de dados. Dentre as características faltantes estão as que possibilitam especificar apropriadamente a modelagem dos dados. No entanto, determinar a estrutura existente entre as amostras não é condição necessária para o desenvolvimento de uma máquina de aprendizado de melhor desempenho. Na maioria dos casos, o treinamento dos modelos se baseia na eficiência do mapeamento entrada-saída, e, eventualmente, na complexidade da máquina. Sem contemplar o conhecimento sobre a estrutura dos dados, fica incompleta a modelagem destes últimos.

No caso em questão, deseja-se especificar os parâmetros de um classificador de margem larga (ou seja, de máxima distância de cada classe ao hiperplano de separação; ver seção 2.4 na página 13) agregando à sua constituição informações sobre a estrutura dos dados modelados. É preciso para tanto lançar mão de métodos de agrupamento de dados, alinhamento de matrizes e otimização de funções. Por isso, deve-se estabelecer os termos e as condições em que se dá a busca desse objetivo.

2.2.1 Definição de kernel

Um núcleo ou *kernel* é definido como sendo uma função K que mapeia os pontos no espaço de entrada de dimensão n_0 para pontos correspondentes em um novo espaço de dimensão n_1 a que se refere como espaço oculto ou de características. A operação de mapeamento ou transformação consiste no produto interno dos pontos segundo uma função não linear implícita. Serão dados maiores detalhes ao longo deste capítulo complementando as definições a seguir.

2.3 Definições sobre as matrizes

2.3.1 Kernels de Mercer

No âmbito do aprendizado supervisionado, os *kernels* de Mercer [10] são caracterizados principalmente pelo fato de poderem mapear os dados de entrada para o espaço de características sem que o cômputo do mapeamento

seja necessário. Isso é possível porque o mapeamento é obtido pelo cálculo do produto interno ainda no espaço de entrada. A matriz de *kernel* (ou de Gram) $K = [k(x_i, x_j)]$ é tida como *kernel* de Mercer se é simétrica e positiva semi-definida. Sendo assim, a matriz de *kernel* é geralmente considerada com sendo uma matriz que captura a similaridade entre todos os pares de pontos de uma base de dados.

Dentre as funções disponíveis para a implementação do *kernel* existem as polinomiais, a linear, as sigmoidais e a gaussiana. As matrizes geradas por essa última também são chamadas de *kernel Radial Basis Function* (RBF), cuja equação que as determina é a seguinte:

$$k(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}} \tag{2.1}$$

em que σ é o raio da função gaussiana. A avaliação de cada par de pontos segundo a relação 2.1 indica a semelhança entre x_i e x_j. Tomando x_j como centro da gaussiana, por meio da razão entre a distância euclidiana e o raio de abrangência da função tem-se uma estimativa de similaridade entre as amostras.

2.3.2 Fuzzy C-Means (FCM) e Matrizes de Proximidade (FPM)

O *Fuzzy C-Means* (FCM) foi proposto por Bezdek [4]. Também conhecido por *Fuzzy ISODATA* [30, 19], o método em questão possui uma heurística para tentar identificar estruturas em conjuntos de dados: cada amostra pertence a cada um dos grupos segundo um grau de participação. Isso se dá com base na abordagem nebulosa em que são estabelecidos limites suaves entre grupos ao invés de fronteiras com transição abrupta. A saída do método não é uma partição, e sim, um agrupamento em que cada *cluter* nebuloso é um conjunto de todos os padrões. Tal abordagem dá ao método algumas vantagens. A primeira é a possibilidade de uma mesma amostra pertencer com algum grau a todos os grupos ao mesmo tempo, sendo cada relação de participação dada por valores contínuos (entre 0 e 1), e não, discretos como em outros métodos (0 ou 1 equivalentes, respectivamente a não pertence e pertence ao grupo). A segunda vantagem é a possível identificação de pontos localizados nas margens de separação do grupos quando tais amostras pertencem com graus semelhantes a dois ou mais *clusters*. Por fim, a abordagem confere certa vantagem ao método FCM quando aplicado a dados com interseção espacial das distribuições de grupos. Conforme observa [26], o subconjunto de amostras na fronteira entre as classes em geral não possui um tamanho mínimo único, mas é de grande valia para o estabelecimento da superfície de separação.

O FCM, um dos mais significativos métodos *off-line* de agrupamento se-

gundo [30], é formulado como um problema de otimização restrita. A busca pelo centro de cada grupo é feita de tal forma a minimizar a função de custo de dissimilaridade. A determinação prévia do número de centros dá-se pela união de abordagens baseadas em tentativa e erro e quando possível, na avaliação do comportamento de alguma métrica influenciada pelo número de grupos (c). Normalmente o número de protótipos é independente do número de classes. Por esse motivo, se os rótulos das amostras não forem tomados em consideração, grupos podem ser constituídos por dados de mais de uma classe. Evidentemente, a definição do número de grupos envolve um custo de tempo que é função do número de valores testados e do tempo demandado pelo FCM para cada tentativa. Por isso, procura-se determinar valores baixos para c a fim de se ter um compromisso adequado entre o tempo demandado, o número de partições a serem modeladas e a complexidade de outros algoritmos dependentes desta escolha.

A matriz de partição (ou de graus de participação) $U = [u_{ik}]$ do método FCM satisfaz a restrição 2.2,

$$\sum_{i=1}^{c} u_{ik} = 1, \forall k = 1, \ldots, N. \tag{2.2}$$

tal que a função objetivo a ser minimizada é determinada pela equação:

$$J(U, c_1, \ldots, c_c) = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} \|x_k - v_i\|^2 \tag{2.3}$$

em que c é o número de partições (ou *clusters*) no massa de dados de tamanho N, v_i é o protótipo (centro candidato, semente ou centroide ponderado) do grupo i e m é o coeficiente de *fuzziness* (ou expoente de ponderação tal que $m \in (1, \infty]$). Este último parâmetro controla a característica nebulosa dos grupos: para $m \to 1$, a divisão dos conjuntos se torna mais abrupta; para $m \to \infty$, os conjuntos se tornam mais nebulosos, com cada ponto pertencendo a todos os *clusters*. Para se aproximar do valor mínimo da função de custo, o método parte de centros gerados aleatoriamente e itera sobre as condições 2.4:

$$c_i = \frac{\sum_{k=1}^{N} u_{ik}^{m} x_j}{\sum_{j=1}^{N} u_{ij}^{m}} \quad \text{e} \quad u_{ij} = \frac{1}{\sum_{k=1}^{c} (\frac{\|v_i - x_j\|}{\|v_k - x_j\|})^{2/(m-1)}} \tag{2.4}$$

É preciso destacar que por ser um método heurístico, não há garantia de sua convergência para a solução ótima. Ou seja, pode haver convergência para um valor mínimo local do erro quadrático, utilizado como critério da evolução dos resultados. De modo geral, o desempenho do método depende da distribuição espacial dos dados, da normalização destes, da métrica de

distância utilizada, da geração dos centros iniciais e do número de centros previamente informado. Grande parte dos códigos implementados como o FCM do *MATLAB* utilizam como estratégia de início a geração aleatória da matriz U atendendo às restrições 2.2. A cada iteração, o método realiza o cálculo dos c centros minimizando a função de custo conforme a sequência:

1. Criação da matriz U com valores aleatórios entre 0 e 1 atendendo à restrição 2.2;
2. Cálculo dos centros c_i segundo 2.4;
3. Avaliação da função de custo 2.3. Parar se o valor da função for menor que o mínimo tolerado ou não houver melhora em relação à iteração anterior. Caso contrário, continuar;
4. Calcular uma nova matriz U conforme u_{ij} de 2.4 e retornar ao passo 2.

A matriz de partição $U = [u_{ik}]$ é a base para a obtenção da matriz de proximidade $P = [p_{kl}]$ de acordo com a conhecida relação registrada em 2.1 conforme apresentada pela função 2.5 a seguir:

$$p_{kl} = \sum_{i=1}^{c} min(u_{ik}, u_{il}) \tag{2.5}$$

A matriz de proximidade P, assim como as matrizes de *kernel*, contém as relações entre os padrões de acordo com as relações de participação nebulosa representadas na matriz de partição U.

2.3.3 Matrizes de Afinidade

Para um conjunto de dados Γ_u constituído por N amostras x, os elementos s_{ij} da matriz de Afinidade (ou Similaridade [8]) $S = [s_{ik}]$ contêm a medida ou estimação da afinidade dos pares de padrões (x_i, x_j), uma vez que a afinidade é definida com sendo a semelhança baseada na relação ou conexão eventual [47]. Por haver afinidades reflexivas, S é simétrica, o que implica em $s_{ij} = s_{ji}$. Em geral, a matriz de Afinidade pode ser representada como uma matriz simétrica diagonal em bloco como a seguinte:

$$S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1k} \\ S_{21} & S_{22} & \cdots & S_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ S_{k1} & S_{k2} & \cdots & S_{kk} \end{bmatrix} \tag{2.6}$$

em que $S_{ij} = S_{ji}$ são sub-matrizes de S e k é o número total de sub-grupos de S.

Para conjuntos de dados ordenados, as matrizes de proximidade e de *kernel* podem ser representadas na forma anterior 2.6, em que a sub-matriz S_{ii} representa a afinidade intra grupos e a S_{ij} representa a afinidade entre grupos para $i \neq j$.

2.4 Métodos de aprendizado

A utilização de métodos que utilizam as matrizes de núcleo (*kernel*) tem sido enfatizada em grande medida pelas Máquinas de Vetor de Suporte (SVMs). Nestas, a sintonia dos parâmetros da função de mapeamento tem um dos papéis mais importantes no desempenho do classificador, uma vez que ele faz uso da informação sobre os dados contida na matriz. No entanto, a determinação dos parâmetros de uma função de *kernel* não é conseguida por métodos de cálculo direto dos mesmos. A maioria das formas correntes de obtenção dos parâmetros envolve buscas exaustivas no espaço de características segundo algum método de otimização. Como a avaliação das soluções candidatas pauta-se em grande medida pelo desempenho do classificador final dadas algumas restrições, muito frequentemente tanto a busca quanto o resultado desse processo não permitem o entendimento das relações contidas no *kernel* com relação ao espaço de entrada. Há, portanto, a perda dessa informação importante contida nos dados e que poderia ser encontrada no classificador final se fosse imposto algum conjunto de restrições durante a busca. Antes, contudo, é preciso investigar se a imposição adicional de tais condições vai de encontro ao desempenho do classificador.

Informação semelhante quanto à estrutura dos dados também está contida na matriz de proximidade (FPM) resultante do método FCM. De fato, a partição dos dados pode ser obtida utilizando informação contida no *kernel* encontrando a ordem das amostras que revela o relacionamento entre os grupos. Uma representação similar da matriz de afinidade pode ser obtida por FPMs, o que pode sugerir que *kernel* e FPM encorporam informação semelhante sobre os dados se os parâmetros de ambas forem escolhidos apropriadamente de forma a manter relações semelhantes entre os dados. Partindo de tal observação, é possível inferir que o aprendizado supervisionado e não-supervisionado podem ter uma relação estreita através da troca de informações entre *kernels* e FPMs. Ou seja, as matrizes resultantes representam a mesma relação entre as amostras, aqui chamada de informação estrutural dos dados. Mas para aproximar as matrizes, há um problema que consiste na determinação dos parâmetros de ambos de forma a maximizar a semelhança entre os resultados. Para melhor entendimento deste problema, a sua definição será feita mais adiante neste capítulo.

A separação dos vetores das classes pelo hiperplano ótimo (ou hiperplano de margem máxima) ocorre se as classes são separadas sem erro e a distância do vetor mais próximo de cada classe até o hiperplano é máxima. Os classificadores em que a resposta final é um hiperplano que atende a condição descrita são considerados de margem larga. Tanto a SVM quanto a Máquina de Vetor de Suporte por Mínimos Quadrados (LS-SVM) são redes de vetores de suporte em que os vetores de entrada são mapeados por uma função do espaço de entrada não linear para um novo espaço de alta dimensão em que é construído um hiperplano ótimo capaz de separar as classes. A função que realiza este mapeamento entre dimensões neste caso atende à condição de Mercer [43]. E os vetores de suporte são as amostras selecionadas para definir o hiperplano de separação.

As duas máquinas de aprendizagem apresentadas a seguir têm recebido grande atenção dos pesquisadores e possuem aplicações em que alcançam desempenho relevante em problemas de classificação (binária e multi-classes) e regressão de funções.

2.4.1 Support Vector Machine (SVM)

A Máquina de Vetor de Suporte (SVM) foi desenvolvida por Boser et al. [5]. Seguindo a teoria de aprendizado estatístico, a sua formulação resulta em um problema convexo que pode ser resolvido por programação quadrática [43]. Procura-se minimizar o Erro Estrutural (ou de generalização) e o Erro Empírico (ou de treinamento) pelo Princípio de Minimização do Risco Estrutural [38, 9]. Segundo [23], vários são os métodos que são utilizados para obter a resolução deste problema, que é global e única. Os limites do seu erro de generalização estão expressos em termos da dimensão Vapnik-Chervonenkis (VC) com base também na teoria de aprendizado estatístico: a taxa de erro de generalização é limitada pela soma da taxa de erro de treinamento e por um termo que depende de VC [27].

A classificação não linear, a estimação de função e a estimação de densidade são problemas para os quais é indicada a metodologia desta máquina de aprendizado. O classificador SVM é interpretado como sendo de margem larga. Vários tipos de funções de *kernel* são possíveis nas SVMs desde que seja satisfeita a condição de Mercer [43]. Dentre as suas desvantagens se destacam a seleção dos hiper-parâmetros e a matriz envolvida no problema QP, cujo tamanho é proporcional ao número de pontos de treinamento.

Para um conjunto de treinamento $\{(x_i, y_i)\}_{i=1}^{N}$, com $x_i \in \Re^n$ dados de entrada e $y_i \in \{-1, +1\}$ rótulos de saída correspondentes para as classes, o classificador SVM satisfaz as seguintes condições:

$$\begin{cases} \omega^T \varphi(x_i) + b \geq +1, & \text{se} \quad y_i = +1 \\ \omega^T \varphi(x_i) + b \leq -1, & \text{se} \quad y_i = -1 \end{cases} \tag{2.7}$$

em que: $\varphi(\cdot) : \Re^n \rightarrow \Re^{n_h}$ é a função não linear que mapeia o espaço de entrada para um espaço de características de alta dimensão; ω é o vetor de pesos associados a cada vetor de suporte; b é o parâmetro de polarização. O problema de otimização por sua vez é definido por:

$$min_{\omega,b,\xi} \jmath(\omega, \xi) = \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{N}\xi_i \tag{2.8}$$

sujeito a

$$\begin{cases} y_i \left[\omega^T \varphi(x_i) + b\right] \geq 1 - \xi_i, & i = 1, \ldots, N \\ \xi_i \geq 0, & i = 1, \ldots, N \end{cases} \tag{2.9}$$

em que ξ_i são as variáveis de folga necessárias para permitir erros de classificação no conjunto de igualdades (por exemplo, devido a sobreposição de distribuições). A constante $C > 0; C \in \Re$ (ou γ em outras definições) é considerada o parâmetro de sintonia do algoritmo e controla o compromisso entre a complexidade da máquina e o número de pontos não-separáveis [27].

2.4.2 Least Squares Support Vector Machine (LS-SVM)

A Least Squares Support Vector Machine foi proposta por Suykens e Vandewalle [41]. O que difere esta máquina de aprendizado transdutivo (estimação direta dos rótulos de amostras novas e de comportamento diferente do aprendido no treinamento) da SVM é a modificação da formulação do problema de otimização para utilizar restrições de igualdade, o que permite a resolução eficiente de sua função de custo no espaço dual pelo algoritmo do Gradiente Conjugado [23] e por outros métodos [38]: um sistema de equações lineares que deve atender às condições de Karush-Kuhn-Tucker [40, 24, 25]. Portanto, as LS-SVMs têm menor custo computacional no treinamento e têm desempenho similar sob certas condições em comparação com as SVMs [48]. Segundo [23], a formulação das LS-SVMs corresponde implicitamente a uma regressão rígida com classes binárias ± 1.

Na LS-SVM, todos os vetores são considerados vetores de suporte já que os valores dos multiplicadores de Lagrange são proporcionais ao erro [24]. Parte significativa desses mesmos multiplicadores é nula na SVM, o que representa uma grande diferença em relação ao outro método. A esparsidade perdida na LS-SVM pela escolha da norma-2 é uma desvantagem que pode ser contornada por procedimento de poda dos vetores menos significativos

[23]. A validação cruzada realizada i-vezes combinada a diferentes tipos de busca permite a determinação dos hiper-parâmetros da LS-SVM.

A LS-SVM tem o seu problema de otimização definido da seguinte forma:

$$min_{\omega,b,\epsilon} J(\omega,\xi) = \frac{1}{2}\omega^T\omega + \gamma\frac{1}{2}\sum_{i=1}^{N}\epsilon_i \tag{2.10}$$

sujeito às condições de igualdade

$$y_i\left[\omega^T\varphi(x_i)+b\right] = 1-\epsilon_i, i = 1,\ldots,N. \tag{2.11}$$

A solução é obtida após a construção do Lagrangiano:

$$L(\omega,b,\epsilon,\alpha) = L(\omega,b,\epsilon) - \sum_{i=1}^{N}\alpha_i\{y_i\left[\omega^T\varphi(x_i)+b\right]\} \tag{2.12}$$

em que $\alpha_i \in \Re$ são os multiplicadores de Lagrange que podem ser positivos ou negativos na formulação da LS-SVM. Das condições de otimalidade, é obtido o sistema de Karush-Kuhn-Tucker (KKT):

$$\begin{cases} \frac{\partial L}{\partial \omega} = 0 \rightarrow \omega = \sum_{i=1}^{N}\alpha_i y_i \varphi(x_i) \\ \frac{\partial L}{\partial b} = 0 \rightarrow \omega = \sum_{i=1}^{N}\alpha_i y_i \\ \frac{\partial L}{\partial \epsilon_i} = 0 \rightarrow \alpha = \gamma\epsilon_i, i = 1,\ldots,N. \\ \frac{\partial L}{\partial \alpha_i} = 0 \rightarrow y_i\left[\omega^T\varphi(x_i)+b\right] - 1 + \epsilon_i = 0, i = 1,\ldots,N. \end{cases} \tag{2.13}$$

Na maioria dos casos, SVMs e LS-SVMs com *kernel* RBF têm desempenho ao menos tão bom quando é utilizada a função de transformação linear [23]. Ou seja, o índice de acerto desses classificadores com tais funções de *Kernel* é maior que 80%. E isso significa que frequentemente pode-se obter conhecimento se a fronteira ótima de decisão é ou não fortemente não linear. Como as LS-SVMs RBF em [23] e as SVMs RBF em [8] obtiveram o melhor desempenho dentre as funções testadas, a função RBF 2.1 foi a escolha para a função de mapeamento nos experimentos desta dissertação. A adoção do núcleo RBF também em [7], por possibilitar uma métrica robusta à distribuição dos dados, corrobora o seu emprego aqui. Sendo assim, a função de *Kernel* $K(x,x_i) = \varphi(x)^T\varphi(x_i)$ escolhida *Radial Basis Function* (RBF) $K(x,x_i) = e^{\frac{\|-x-x_i\|^2}{2\sigma^2}}$.

O classificador LS-SVM é dado pela expressão:

$$y(x) = sign\left[\sum_{i=1}^{N}\alpha_i y_i K(x,x_i) + b\right] \tag{2.14}$$

em que a função $sign$ é uma função degrau de saídas -1 e $+1$.

2.4.3 Considerações sobre a Radial Basis Function (RBF)

As Funções de Base Radial, sob o ponto de vista da regressão utilizando *Kernel* com base na noção de estimativa de densidade, relacionam o uso das Redes Neurais Artificiais RBF às Máquinas de Vetor de Suporte. Dentro do contexto da regularização para a transformação do problema de mapeamento de entrada-saída tornando-o bem-formulado, percebe-se as interconexões entre as formulações desses modelos para a separação de padrões. No entanto, as SVMs estão estruturadas de tal forma que o hiperplano tenha superfície com margem máxima de separação entre as classes. Além disso, ao ser uma implementação com base no princípio de minimização do risco estrutural, a SVM tem bom desempenho em problemas de classificação ainda que ela não incorpore conhecimento do domínio do problema. Ainda segundo [27], apenas as SVMs possuem este atributo.

Assim como as Redes de Funções de Bases Radiais, as LS-SVMs são utilizadas como classificadores em problemas uni e multi classes. Em especial, tais modelos têm em comum a estratégia fundamentada no Teorema de Cover com relação à separabilidade de padrões [27, 15]. São ambas aproximadores universais locais cuja primeira camada agrupa os dados em *clusters* transformando os conjuntos de padrões de entrada não linearmente separáveis em um conjunto de saídas linearmente separáveis. A segunda camada é a de saída e tem a função de classificar os padrões transformados pela camada de entrada. Contudo embora a arquitetura das Redes RBF seja semelhante às redes neurais artificiais do tipo perceptron multi-camadas (MLP - *Multilayer Perceptron*) [16] por conter nodos, a função de ativação (ou função de transformação da camada) comumente utilizada é a mesma das LS-SVMs com *kernel* do tipo RBF: a função gaussiana

$$f(\upsilon, \sigma) = e^{\left(\frac{-\upsilon^2}{2\sigma^2}\right)} \tag{2.15}$$

em que

$$\upsilon = \|x - \mu\| \tag{2.16}$$

representa a distância euclidiana entre o vetor $\mathbf{x}$ de entrada e o vetor μ do centro da função radial de largura σ. Tal como nas principais funções de base radial utilizadas com maior frequência, o *kernel* aqui escolhido para as LS-SVMs possui o centro μ e o raio σ como seus únicos parâmetros. Outra característica importante da LS-SVM escolhida é também não apresentar capacidade semelhante à MLP quanto à generalização em regiões para as quais não há dados de treinamento [16]. Logo, os classificadores obtidos por uma LS-SVM não possuem o problema dos "falsos padrões" [16]: a classificação,

em uma das classes existentes, de amostras significativamente diferentes das presentes na etapa de treinamento.

Quanto à complexidade da etapa de treinamento, as LS-SVMs tipicamente têm complexidade de memória e de cálculos maiores que $O(N^2)$ [40]. Para a utilização de LS-SVMs neste trabalho, foram utilizados os códigos de [40, 25], cujo núcleo está escrito em linguagem C. A escolha por esse conjunto de códigos deu-se pelo seu reconhecido bom desempenho e sua integração direta com os demais códigos em *MATLAB*. Cabe registrar que não foram utilizadas: a formulação de LS-SVM fixo para grandes bases de dados; a estrutura de sintonia automática por validação cruzada; o procedimento de poda da estrutura final, reduzindo a complexidade por retirada dos valores suporte menos importantes para o desempenho da máquina.

Como não há a necessidade da definição de número de nodos ou de camadas, a sintonia dos parâmetros e a determinação dos vetores suporte concentram os esforços de cálculo. Entretanto, não há mapeamento exato do vetor de entrada para a saída uma vez que apenas uma função radial ajustada na etapa de treinamento transforma as amostras para a camada de saída. Evita-se desse modo problemas como o *overfitting*, enquanto que a complexidade fica a cargo do número de vetores suporte. Logo, a velocidade da construção e ajuste de novos classificadores é dependente do número amostras de treinamento, o que dificulta a sintonia dessas máquinas para grandes bases de dados.

Assim como na etapa de treinamento de redes do tipo RBF, podem ser utilizados diferentes métodos em cada uma das fases de determinação dos *clusters*, para determinar o número de grupos, bem como o centro e o raio da função. Não raramente o problema de classificação envolve conjuntos de amostras que possuem mais de três dimensões de entrada (ou características). Isso impossibilita a identificação visual dos agrupamentos existentes nas amostras, seja no conjunto total seja nas partes utilizadas para o treinamento, para a validação e para o teste.

Métodos como o *K-means-clustering* (ou K-médias [16]) e o *Fuzzy C-means* (C-médias Nebuloso) entre outros podem auxiliar a determinar os grupos permitindo o ajuste dos parâmetros μ e σ. No entanto, é preciso obter, previamente, o número de grupos em que o total de amostras deve ser dividido. Os agrupamentos resultantes dos métodos citados podem fornecer indicadores de qualidade da divisão do conjunto inicial, tais como similaridade dos integrantes de cada subconjunto. Porém, em casos em que se deseja utilizar matrizes de *kernel* ou de similaridade para construir um classificador, os métodos de *clustering* não informam diretamente os parâmetros da função de mapeamento. Sendo assim, para ajustar o classificador, o método proposto

em [36] procura os melhores parâmetros na semelhança da matriz de proximidade (fruto da transformação do resultado de um método de agrupamento) com a matriz de *kernel*.

2.5 Métodos de Ordenação de Dados em Matrizes

Os algoritmos de agrupamento podem ser descritos de acordo com o objetivo da ordenação das matrizes. Há vários algoritmos de agrupamento relacionados na literatura e cujo objetivo é ordenar linhas e colunas de matrizes tais como as de *kernel* de modo a obter matrizes de afinidade. Uma vez formadas essas últimas, a informação dos grupos contida nas matrizes pode ser diretamente extraída por *kernels*.

A aplicação do método de ordenação por autovetor, por exemplo, permite à matriz de afinidade que representa o *kernel* revelar a informação já presente neste. Para tanto, é necessário permutar linhas e colunas segundo um critério que direcione as operações na matriz e, consequentemente, a sequência de amostras. Conforme será demonstrado, o *kernel* ajustado apropriadamente contem a informação sobre a estrutura dos dados. E essa informações é equivalente à obtida por outros métodos de agrupamento.

São apresentados nas subseções a seguir alguns dos vários métodos disponíveis para a ordenação de valores em matrizes. Procurou-se agrupar as técnicas com base no princípio que rege a heurística para a obtenção da permutação ótima de linhas e colunas.

2.5.1 Sorting Points Into Neighborhoods (SPIN)

Ordenação de Pontos na Vizinhança é a tradução do método não supervisionado de busca heurística iterativa proposto por [42]. Foi inspirado no problema de revelar e apresentar variáveis e trajetórias contínuas. Para uma matriz de distância entre N amostras de um conjunto inicialmente desordenado, o modelo visual desenvolvido apresenta de forma gráfica (por pseudocores) o agrupamento de pontos próximos. Para conseguir através do método em questão a ordenação desejada, a matriz de entrada precisa ser submetida a dois algoritmos diferentes, *Side-to-side (STS)* e *Neighborhood*, que permutam os grupos de pontos segundo as propriedades que lhes dão os respectivos nomes. Tem-se ao final do processo uma nova matriz transformada a partir da inicial que sofreu a permutação que revela a relação entre as amostras do conjunto.

Assim como nos demais métodos a seguir, no SPIN as distâncias grandes são concentradas nas bordas das matrizes, enquanto que na diagonal

concentram-se as distâncias pequenas. Há, portanto, a formação de matrizes bloco na diagonal.

A busca pela permutação ótima das linhas e colunas é formulada como um problema de otimização (Problema de Associação Quadrática - QAP) de uma função de custo a ser minimizada. Por ser de difícil tratamento computacional (*NP-Hard*), a aproximação da solução se dá por força bruta ou por heurísticas. Nestes dois algoritmos apresentados em [42], a ordenação iterativa dos pontos dá-se pela busca heurística de uma ordenação linear. Enquanto o STS desloca as distâncias grandes para as regiões fora da diagonal (topo direito e base esquerda), o *Neighbourhood* concentra as distâncias pequenas na diagonal da matriz. Ainda segundo seus autores, o primeiro algoritmo é mais rápido que o segundo, porém esse último fornece resultados melhores para dados cuja estrutura contém objetos compostos.

2.5.2 Bond Energy Algorithm (BEA)

O Algoritmo de Energia de Vinculação, revisitado em [2], é empregado para a fragmentação vertical de bases distribuídas de dados. Os agrupamentos obtidos pelo método em [46], por exemplo, são baseados na relação entre elementos circundantes dos dados analisados em um grupo particular. Para isso, o algoritmo utiliza como métrica a Afinidade de Atributo não trivial: uma matriz de pesos que armazenam a força da relação entre todos os elementos do conjunto. Segundo [2], as relações transitivas são exemplo das relações de difícil identificação que podem ser reveladas pelo método utilizando um terceiro elemento.

O agrupamento se dá em duas etapas utilizando os valores de afinidade de atributos na matriz de afinidade de atributos. A primeira, que consiste no algoritmo de ordenação, aloca em regiões próximas os elementos mais relacionados entre si, separando também os não relacionados. A segundo etapa identifica e realiza cortes nos dados ordenados, sendo o algoritmo de que cria os grupos.

2.5.3 Métodos que utilizam autovetores

Alguns trabalhos recentes utilizam os autovetores da matriz de afinidade para obter a ordenação desejada [47, 45]. O grande atrativo desses métodos é a simplicidade característica da Decomposição Espectral da matriz em autovalores e autovetores por algoritmos de diagonalização de matrizes [37]. Embora a estabilidade desses últimos seja bem entendida, essa não é a condição atual da decomposição obtida no contexto de segmentação [47]. Ainda segundo esta referência, os autovetores das matrizes permutadas são as permutações dos

autovetores da matriz original. O primeiro e mais significativo autovetor ordenado é mencionado como sendo a escolha que fornece a melhor permutação das amostras para aproximar as que se assemelham e permitir a obtenção da matriz diagonal em bloco.

As definições em [11] serviram de base para o modo como um dos autovetores foi empregado nesta dissertação. Dada uma matriz de *kernel* ou de proximidade, são obtidos os seus autovalores e autovetores associados. O autovetor mais significativo, referente ao primeiro e maior autovalor, tem tamanho N, devendo ser ordenado em ordem crescente de valores guardando a ordem antiga dos valores no vetor. Desse modo, a permutação ótima das amostras é justamente a nova posição dos valores do autovetor em sua nova ordenação.

2.5.4 Método Minus

Utilizando a denominada Escala de Conformidade para encontrar a Melhor Decisão (*Best Decision - BD*), [44] define o problema e apresenta a técnica Minus para reordenar linhas e colunas em uma matriz. Não há necessidade de que esta última seja quadrada, mas os valores nela contidos devem ser inteiros.

Como métrica escolhida, a Conformidade é calculada pela transformação que usa a frequência de ocorrência (também chamada de Frequência de Transformação - FT) ao invés do valor do atributo. Dessa forma, em cada linha (ou coluna), é calculada a soma de todos os valores de frequências dos atributos, o que faz de cada uma das somas o peso da Conformidade em cada linha correspondente. Assim, quanto maior é a Conformidade de uma linha, mais representativos são os valores dos atributos dessa linha para a matriz, e, por esse princípio, a linha com a maior Conformidade torna-se a Melhor Decisão.

O Algoritmo 1 descreve o pseudo código da técnica Minus tomando como exemplo uma matriz P de tamanho N em que cada elemento p_{ij} possui um valor discreto entre 1 e L.

2.6 A métrica de alinhamento entre matrizes

Como forma de quantificar a similaridade entre as matrizes de *kernel* e de proximidade, adotou-se neste trabalho o Alinhamento Empírico (A) apresentado detalhadamente em [11]. A grandeza A utilizada é definida pela função a seguir:

$$A(K,P) = \frac{\langle K,P\rangle_F}{\sqrt{\langle K,K\rangle_F \langle P,P\rangle_F}} \tag{2.17}$$

Algoritmo 1: Método Minus

Data: Matriz P de tamanho N em que cada elemento p_{ij} possui um valor discreto entre 1 e L.
Result: Ordem de retirada das colunas.

```
begin
    while ainda houver colunas na matriz P ≠ ∅ do
        Passo 1: Calcular as frequências FT(t, j) para cada valor de
        atributo t = 1, 2, ..., L_j nas linhas j, onde j = 1, ..., N
        Passo 2: Para cada coluna i = 1, 2, ..., N, encontrar as somas (ou
        pesos) W(i) = ∑ FT(t, j)
        Passo 3: Encontrar R = minW(i) e guardar i
        Passo 4: Eliminar a coluna i da matriz
    end
    Passo 5: Reordenar as colunas da matriz na ordem das eliminações
    efetuadas no Passo 4
end
```

em que K e P são, respectivamente, as matrizes de *kernel* e de proximidade. A operação $\langle , \rangle_F$ é o produto interno de Frobenius entre duas matrizes (originalmente normalizadas segundo [11]) definido pela equação:

$$\langle K, P \rangle_F = \sum_{i=1}^{N} \sum_{j=1}^{N} K(i,j) P(i,j). \tag{2.18}$$

Tomando-se como meta a investigação de métodos para maximizar a semelhança entre as matrizes geradas pela função Gaussiana Multivariada 2.1 (também chamadas de RBF *kernels*) e as matrizes resultantes de FCM, os parâmetros a serem ajustados são o σ e o c. Espera-se como resultados do ajuste simultâneo desses parâmetros atingir o objetivo inicial e também a obtenção de grupos (*clusters*) que revelem o número de funções geradoras dos dados.

Diferentemente de [11], A não é calculado em relação aos rótulos das amostras, mas em relação ao agrupamentos. Portanto, a classificação prévia das amostras não influencia a otimização de A.

Resultados experimentais em [36] apontam a possibilidade de se obter um valor para c que coincide com o número de grupos das funções geradoras. Há ainda no mesmo trabalho a sugestão de que, para uma dada matriz de *kernel*, o número apropriado de grupos pode ser induzido pela maximização da similaridade entre a matriz de *kernel* e a FPM. Tais resultados mostram também que para os mesmos problemas, a aplicação em SVMs do valor do parâmetro σ que maximiza a similaridade resulta em classificadores binários com desempenho satisfatório.

Por sua qualidade e a sua capacidade de fornecer U além de criar as partições, o FCM foi escolhido para fornecer as entradas necessárias à avaliação de A. No entanto, cabe registrar que o FCM é utilizado no Classificador Re-

lacional Fuzzy citado em [7], onde os autores atribuem a falta de robustez do classificador à falha do FCM ao lidar com conjuntos de dados de distribuição não esférica ou que também contenham pontos muito diferentes dos demais (*outliers*). Nesta condições, os centros encontrados para os grupos não correspondem aos centros reais. Para contornar essas características, é utilizada uma variação do FCM com *kernel* e novas funções para a definição dos centros, proporcionando a robustez desejada. A estratégia utilizada neste trabalho utiliza a função objetivo original do FCM, o que a difere da estratégia *kernel* FCM de [7].

2.7 O alinhamento de matrizes como problema de otimização

Como exposto anteriormente, dependendo dos seus parâmetros, *kernels* e FPMs podem conter informações análogas sobre as relações que os padrões e os grupos possuem no espaço de entrada. Para o caso dos *kernels* do tipo RBF, a informação estrutural pode ser revelada por meio da determinação apropriada do raio σ, enquanto que nas FPMs, a habilidade de descrever as relações entre os padrões depende do número c de grupos previamente escolhido. Ainda assim, esses parâmetros são configurados previamente pelo usuário de modo cego. No projeto de uma SVM, o valor de σ pode ser finamente ajustado de acordo com o desempenho integral da máquina de aprendizado, enquanto análises posteriores de grupos podem também fornecer um palpite para o ajuste correto do valor de c.

Quando a determinação de (σ, c) resulta em matrizes de afinidade que são coerentes com a estrutural real dos dados, o *kernel* e o FPM correspondentes são alinhados um com o outro de acordo com a métrica A escolhida. Tais argumentos sugerem que A é função de K e P, ou seja, $A(K, P)$ e que possui uma região de máximo valor próximo ao par ordenado (σ^*, c^*). Isso caracteriza um problema de otimização segundo a definição 2.19. O objetivo é então obter os valores dos parâmetros supracitados que maximizem o alinhamento e que, como esperado, sejam também os que melhor descrevem o conjunto de dados tanto para o *kernel* quanto para a matriz de proximidade conforme a relação a seguir:

$$max_{(\sigma,c)} A(K, P) \tag{2.19}$$

O procedimento de validação cruzada (*cross-validation*) [27] foi a ferramenta estatística utilizada similarmente a [23] na metodologia para que os classificadores tenham bom desempenho de generalização. Dos valores candidatos

para o modelo de cada base, foram escolhidos os valores médios de σ e de c, além do valor mais frequente de γ (aqui também definido como C) dentre os que proporcionaram melhor desempenho à LS-SVM.

2.8 Métodos evolucionários de otimização

Esta seção trata de forma breve os principais aspectos dos algoritmos de otimização evolucionária utilizados no presente trabalho: Algoritmo Genético (AG) e *Particle Swarm Optimization* (PSO). Como exposto anteriormente, a escolha de uso desses métodos se justifica pela capacidade atender o objetivo do problema em questão na forma em que a função objetivo está definida. É preciso destacar que em [11, 23, 7], a determinação dos parâmetros se dá por teste em um conjunto de valores candidatos em função do desempenho dos classificadores. Diferente dos trabalhos de referência [36, 23], não foram computados os valores de A segundo o método de grade no espaço de busca. Apesar de o tamanho das bases envolvidas nos experimentos demandar grande esforço computacional para cada avaliação da função de custo, acredita-se ser possível obter melhor aproximação da solução ótima utilizando os métodos com evolução de populações.

Como se pretende destacar, o PSO possui grande similaridade com o AG. Ambos os métodos, com capacidade de avaliar também problemas descontínuos e discretos (combinatoriais), têm sido empregados com sucesso em muitas pesquisas e aplicações. Os resultados obtidos são bastante satisfatórios, sendo que no caso do PSO, eles são atingidos utilizando menos iterações em comparação com outros métodos. Um característica que torna o PSO atrativo é o fato de o método possuir poucos parâmetros para ajuste e estrutura mais simples quando comparado ao AG. Com pequenas variações quando necessárias, os dois métodos são encontrados grande número áreas de aplicação. Dentre as principais aplicações estão a otimização de funções, treinamento de redes neurais artificiais e sistemas nebulosos de controle [24, 16, 20, 30].

2.8.1 Algoritmo Genético (AG)

O Algoritmo Genético (AG) é um método de busca heurística com base em evolução de populações valendo-se de regras probabilísticas [24]. Cada indivíduo da população é um cromossomo constituído por genes que guardam o valor codificado para cada uma das variáveis de busca. Dessa forma, a cada iteração, uma geração de indivíduos é submetida à função de custo. E por ser cada indivíduo uma solução candidata para o problema de otimização, o espaço de busca é explorado de forma distribuída pela população.

A evolução das soluções se baseia no fato de que a cada indivíduo ser atribuído um valor que o caracteriza quanto à sua aptidão para atender o problema. Sendo assim, é possível comparar as avaliações dos indivíduos em cada iteração. Os indivíduos mais bem avaliados são selecionados para constituir uma população em que são aplicados os operadores de cruzamento e mutação, dando origem a uma nova geração de soluções candidatas. Utilizando essas populações, o método consegue realizar uma busca global, evitando a convergência em pontos em que a função de custo assume valor ótimo local.

Por utilizar a avaliação de desempenho dos indivíduos e não as derivadas das funções objetivo, o AG encontra aplicação em diversas áreas em que a formulação do problema de otimização não dispõe ou permite o cálculo mesmo que aproximado das derivadas da função de custo. Por sua natureza estocástica e pela sua sensibilidade à seleção dos seus parâmetros, o AG não garante a obtenção da solução ótima e nem da aproximação desta. Porém, como detalhado em [24, 16], a formulação do método permite uma série de ajustes de parâmetros e adaptações (como por exemplo, alteração de operadores e de tipos de codificação das variáveis) que o capacitam a encontrar soluções quase ótimas. Ainda que o algoritmo demande grande esforço computacional (de cálculo e de memória), é aceitável o tempo necessário para que a população convirja e sejam obtidas boas aproximações da solução ideal.

2.8.2 Particle Swarm Optimization (PSO)

O método de Otimização por Enxames de Partículas (PSO) é uma técnica estocástica também baseada em população. Foi apresentada em [31], tendo sido inspirada em comportamento social de aglomerações de pássaros e de peixes. Cada partícula constitui uma solução candidata possuindo um conjunto de valores para as variáveis independentes da função objetivo e um valor para a sua aptidão para resolver o problema de otimização. O que caracteriza uma partícula são as suas posições (pos_i), a sua velocidade ($veloc_{part}$) e a sua "proximidade do alimento" (aptidão segundo a função de custo). Em outras palavras, cada a partícula é caracterizada respectivamente por suas soluções, pela alteração da solução numa iteração e pela sua aptidão. Em todas as iterações, cada uma das partículas é influenciada pela melhor solução até então e pela melhor solução atual do bando. As equações a seguir demonstram as regras de atualização da velocidade e de posição de uma partícula.

$$veloc_{part} = rand_0 \times ine \times veloc_{part} + cor \times rand_1 \left(pos_{global^*} - pos_i\right) + cor \times rand_2 \left(pos_{local^*} - pos_i\right) \tag{2.20}$$

$$pos_{i+1} = pos_i + veloc_{part} \tag{2.21}$$

A posição nova de uma partícula pos_{i+1} é o resultado da atualização da posição anterior pos_i pela velocidade nova $veloc_{part}$. Por sua vez, essa componente é atualizada a cada iteração a partir da soma ponderada do seu valor anterior e das diferenças entre a posição da partícula e a melhor posição do bando na iteração e de todas as iterações até então. Cada uma dessas componentes é multiplicada, respectivamente, por: um valor aleatório $rand_0$ e a constante de inércia ine; um valor aleatório $rand_1$ e o fator de correção cor; um valor aleatório $rand_2$ e o fator de correção cor;

A população do enxame é iniciada aleatoriamente de forma a criar um conjunto de soluções que vão tender à solução ótima ao longo das gerações. Contudo, não há nesse método operadores como cruzamento e mutação. A cada passo, a velocidade é alterada para cada partícula em direção aos pontos vizinhos de melhor desempenho. A aceleração da partícula recebe um peso por um termo aleatório ($rand_i$), que separa valores também aleatórios gerados para aceleração em direção às localizações vizinhas e de melhor desempenho. Garante-se assim a diversidade de resultados e de posições dos indivíduos em cada iteração, com a exploração aceitável do espaço de busca sem comprometer o número de iterações ou a convergência do método na região da solução ótima.

2.9 Conclusões do capítulo

Tendo sido estabelecidas as bases para o trabalho com os dados, o problema em foco fica delineado e o seu entendimento, possível. Além disso, ficam mais claros os caminhos possíveis para abordagens permitindo a formação das bases para alterações e novas estratégias. Ainda neste capítulo, foi apresentada parte da metodologia proposta e aplicada neste trabalho para a obtenção de classificadores de margem larga e que mantêm em sua constituição a estrutura dos dados de treinamento no espaço de entrada. Maior detalhamento dos procedimentos adotados será feito no próximo capítulo.

Cabe registrar que a metodologia proposta segue em grande medida os trabalhos iniciados por Antônio de Pádua Braga e Witold Pedrycz para investigar os meios para agregar aos classificadores de margem máxima os resultados obtidos por métodos de agrupamento. Tais trabalhos não puderam ser referenciados uma vez que ainda estão em desenvolvimento e não foram publicados.

CAPÍTULO

3

Metodologia dos Experimentos

O primeiro experimento, cujos procedimentos estão ilustrados na Figura 3.1, contou com três bases com valores originais e padronizados. Em situações separadas, cada forma da base foi submetida à validação cruzada de dez amostragens para determinar a constituição de dez conjuntos de treinamento com dois terços do total de amostras. A cada um desses foram aplicados os algoritmos de otimização para determinar os valores dos parâmetros número de grupos e raio do *kernel* gaussiano. Posteriormente, as matrizes de *kernel* e de proximidade foram constituídas utilizando todo o conjunto de amostras e a média dos dez valores encontrados para cada parâmetro. Por fim, cada matriz foi ordenada pelo método do maior autovetor e pelo método Minus.

No segundo experimento, cada uma das nove bases de amostras de casos reais foi utilizada na sua forma original e padronizada, como apresentado na Figura 3.2. Em uma primeira etapa de cada execução dos códigos, foram definidas dez divisões dos conjuntos. Cada divisão deu origem a um conjunto de amostras para treinamento, validação e teste nas proporções indicadas na Tabela 3.5, na página 37. Os conjuntos de treinamento foram aplicados separadamente aos algoritmos de otimização com a mesma função de custo para obter os parâmetros σ e c. Na etapa seguinte, as amostras de treinamento definiram o parâmetro γ da LS-SVM dentre os valores candidatos seguindo o procedimento adotado em [23]. O valor escolhido foi o que mais vezes proporcionou o melhor desempenho do classificador para os dez conjuntos de treinamento. Na penúltima etapa, os dez conjuntos de validação e de teste correspondentes determinaram o desempenho dos classificadores. Os valores médios de σ e c serviram de base para a montagem das matrizes de *kernel* e de

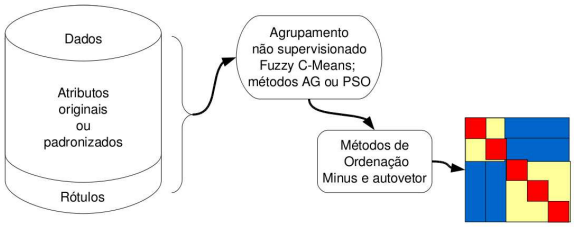

Figura 3.1: Procedimentos realizados no Experimento 1. Utilização de dados rotulados para ajustar os parâmetros σ e c. A partir desses últimos, foram geradas as matrizes de *kernel* e de proximidade que posteriormente foram ordenadas para a observação da estrutura dos dados.

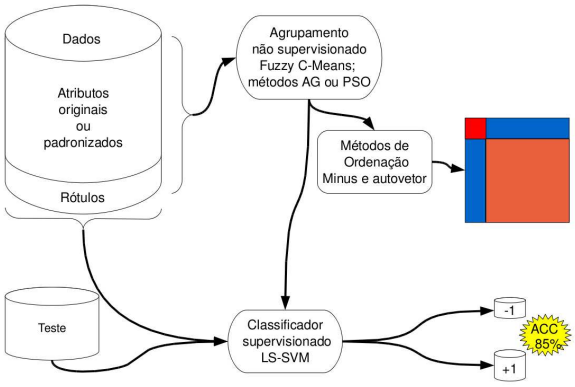

Figura 3.2: Procedimentos realizados no Experimento 2. Utilização de dados rotulados para ajustar os parâmetros σ e c. Esses últimos foram utilizados não apenas no classificador LS-SVM, mas também para gerar as matrizes que posteriormente foram ordenadas para a observação da estrutura dos dados.

proximidade utilizando todas as amostras. Finalmente, os métodos Minus e do maior autovetor ordenaram as matrizes. Uma vez que o trabalho de referência [23] apresentou os valores de σ mas não os de γ para todos as bases de dados, os mesmos valores determinados para γ pela metodologia aqui proposta foram utilizados com os valores de σ de [23] e os encontrados neste trabalho. Apenas desta forma foi possível computar e comparar os desempenhos dos classificadores para cada base.

O último experimento seguiu os mesmos procedimentos do primeiro, porém com dados reais de uma base criada pela união de índices socioeconômicos. Ao final do procedimento ilustrado pela Figura 3.3, foi acrescida uma etapa de geração de mapas geográficos para os municípios de Minas Gerais agrupados conforme o valor c encontrado na etapa intermediária.

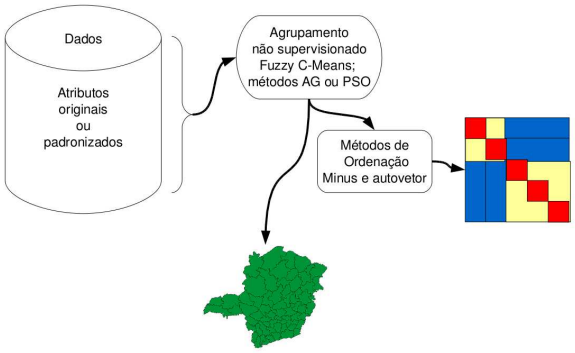

Figura 3.3: Procedimentos realizados no Experimento 3. Utilização de dados rotulados para ajustar os parâmetros σ e c. Esses últimos foram utilizados para gerar as matrizes de *kernel* e de proximidade que posteriormente foram ordenadas para a observação da estrutura dos dados e comparadas aos mapas geográficos.

Pelas razões apresentadas nos capítulos anteriores, várias decisões foram tomadas para adaptar os métodos e os procedimentos às condições que se apresentaram em função das características das bases. As seções a seguir detalham em que condições se deram os experimentos.

3.1 Condições comuns aos três experimentos

Para que a técnica Minus pudesse ser utilizada em matrizes que contêm valores contínuos, cada uma delas foi submetida a um pré-processamento para então ser ordenada. A etapa de transformação da matriz consistiu inicialmente no estabelecimento de nove níveis quartis a partir de todos os valores existentes nas linhas e nas colunas. Em seguida, cada uma das posições da matriz recebeu o valor do nível quartil mais próximo. Só então a matriz passou ao processo de ordenação. Como as matrizes de *kernel* e de proximidade são quadradas e simétricas em relação à diagonal, apenas a ordenação das linhas é necessária e foi realizada tendo aplicado simultaneamente os resultados nas linhas e colunas das matrizes.

A escolha de nove níveis deu-se pelos experimentos prévios que apontaram ser este um número razoável para representar a diversidade de valores existentes nas matrizes destacando-os ao estabelecer uma distância equivalente entre os patamares consecutivos. Além disso, na ordenação, o nivelamento quartil prévio permite à Frequência de Transformação ter uma base constante de caixas no histograma para o seu cálculo.

O custo do erro de classificação foi considerado como sendo o mesmo para as duas classes (positiva e negativa) em todos os casos testados. Entretanto,

esse não é o caso para as bases de dados **gcr**, **hea** e **pid** (ver Tabela 3.3 na página 36) dentre várias outras bases reais [49]. Alguns trabalhos utilizam o erro quadrático médio como métrica para avaliação do desempenho das máquina de aprendizado. Mas o resultado dessa medida não expressa claramente a quantidade de amostras classificadas corretamente, dificultando análises como a da matriz de confusão. Portanto, assim como realizado em [23], o desempenho dos classificadores neste trabalho é calculado pela acurácia (acc), também referida como exatidão, que é definida pela expressão a seguir considerando amostras classificadas como sendo da classe positiva ou da classe negativa:

$$acc = \frac{nVP + nVN}{nVP + nFP + nVN + nFN} \tag{3.1}$$

em que nVP é o número de verdadeiros positivos, nVN é o número de verdadeiros negativos, nFP é o número de falsos positivos e nFN é o número de falsos negativos. Como é possível notar, quanto mais próximo de 1 for o valor de acurácia obtido, melhor será o desempenho do classificador já que maior será o número de amostras rotuladas de modo correto.

É importante destacar que em nenhuma das etapas houve o balanceamento dos conjuntos quanto ao número de amostras de cada classe. Em muitos trabalhos, no projeto de classificadores, esse é um pré-requisito importante e que influencia o desempenho final da máquina de aprendizado [49].

A variação dos resultados obtidos em cada execução do método FCM, por ser iterativo e obter grupos com partição diferente a cada rodada, influencia o processo de otimização. Tendo em mente essa característica, os parâmetros pré-ajustados no FCM permitiram a obtenção de resultados tão semelhantes quanto possível e que aqui foram considerados iguais.

Os experimentos realizados utilizaram os valores padrão de entrada para o método FCM no *MATLAB* sem permitir a impressão das informações ao longo das iterações:

- 2,0 para exponente da matriz U;
- 100 para máximo número de iterações;
- 1e-5 para mínima melhora do resultado.

Diferentemente da metodologia de [23], apenas uma validação cruzada foi realizada com cada grupo de dados. Para cada combinação de método de otimização e opção de pré-processamento dos dados submetidos, as amostras de cada base foram dividas dez vezes em três grupos:

- Treinamento, com aproximadamente dois terços das amostras;

- Validação, com um terço das amostras;
- Teste, com oitenta amostras.

Dessa forma, procurou-se manter parte das escolhas de [23] para possibilitar a comparação dos resultados de validação com os estabelecidos em trabalhos de referência.

Em razão da diferença da proporção de amostras entre os grupos de treinamento e de teste em comparação a [23], os resultados para tais partes dos dados não fornecem as mesmas relações estabelecidas no trabalho de referência. Contudo, acredita-se que essa partição diferenciada dos dados estabelece uma forma mais condizente com os propósitos comumente adotados. Isso porque o conjunto de treinamento é o único a influenciar a construção dos classificadores. O grupo de validação estabelece o desempenho desses últimos. Já o conjunto de teste fornece medidas que podem servir de comparação entre os classificadores de bases diferentes, uma vez que possui sempre o mesmo número de amostras. Essa escolha mantem no grupo de treinamento da maioria das bases o maior número de amostras, exceto para os dados da base **snr**.

Os métodos utilizados neste trabalho para a sintonia dos parâmetros c e σ, embora necessitem de um número muito maior de avaliações de pares candidatos no espaço de busca, permitem que valores não necessariamente contidos no espaço de busca inicial possam ser alcançados. No método de busca por refinamento em [23], tal efeito não é possível. Por esses motivos, acredita-se que os parâmetros encontrados pelos métodos adotados no presente trabalho permitem alcançar aproximações melhores dos valores ideais para a metodologia proposta. Contudo, uma vez que a metodologia adotada em [23] não foi reproduzida neste trabalho, não foi possível confirmar tal expectativa sobre a superioridade do método aqui adotado.

Os códigos para criação e treinamento de LS-SVM [40] permitem a utilização de dados em sua forma original ou pré-processada por normalização (padronização de cada coluna de características por sua média e seu desvio padrão). Como objetivou-se neste trabalho buscar a comparação dos resultados com os obtidos por [23], adotou-se o pré-processamento dos dados na etapa de criação e treinamento da LS-SVM, sem a padronização dos dados na etapa de sintonia do parâmetro γ. O ajuste desse parâmetro de folga da LS-SVM foi realizado tomando como base os valores candidatos $\gamma_0 \in [0,01; 0,05; 0,1; 0,5; 1; 5; 10; 50; 100; 500; 1000]$ adotados em [23]. Tendo determinado σ, todos os valores de γ foram testados em cada uma das dez partições, sendo escolhido aquele que possibilitou o melhor desempenho do classificador no maior número de partições. Só então o desempenho do classificador foi testado para as amostras de validação e de teste.

Grande parte dos algoritmos utilizados foi codificada para uso em *MATLAB*, cuja versão de trabalho foi a 7.6.0.324 (R2008a). Juntamente à orientação para trabalho com matrizes, a facilidade de codificação para execução em qualquer máquina em que o sistema estiver instalado contou para a escolha dessa plataforma, evitando a transposição dos códigos para a linguagens de nível mais baixo. Se por um lado a execução dos experimentos foi prejudicada pela necessidade de mais tempo e memória física das máquinas, por outro ganhou-se a possibilidade para testar outras técnicas já implementadas em sua forma eficiente nos *toolboxes*. Procurou-se ainda alterar os códigos de alguns dos métodos de modo a aproveitar as estruturas e comandos desenvolvidos para matrizes, o que melhorou consideravelmente a eficiência também dos algoritmos mais importantes nos experimentos: o de preenchimento e o de ordenação das matrizes. Conseguiu-se reduzir o número das operações de N^2 para pouco mais da metade ($(N+1)N/2$), além de diminuir o tempo total de realização do experimento com cada base.

O algoritmo 2 em pseudo código registra o procedimento que demanda o menor tempo de processamento para o preenchimento de uma matriz quadrada simétrica. Tal conjunto de operações é solicitado $n_{gen} \times n_{ind}$ vezes no pior caso, tal que n_{gen} é o número de gerações e n_{ind} o número de indivíduos na população do algoritmo evolucionário.

Algoritmo 2: Preenche Matriz

Data: Lista de amostras.
Result: Matriz preenchida com a relação entre os elementos da lista.

```
1  begin
2      Matriz ⟵ ∅
3      V_auxiliar ⟵ [1, ..., N]
4      for i = 1, ..., N do
5          for j = 1, ..., V_auxiliar(i) do
6              Matriz(i, j) ⟵ operacao(amostra_i, amostra_j)
7              Matriz(j, i) ⟵ operacao(amostra_j, amostra_i)
8          end
9      end
10 end
```

Tanto a matriz de *kernel* quanto a de proximidade dependem desse procedimento para que nelas sejam registrados os valores das relações entre as n amostras de cada conjunto segundo os parâmetros a serem ajustados. Uma vez que a função de custo consiste na similaridade entre duas matrizes, a realização eficiente do preenchimento e do número de cálculos possibilitam a investigação do método utilizando bases maiores de dados.

A função de complexidade $f(n)$ do algoritmo de cálculo da função de custo é quadrática. Como a cada avaliação de um conjunto de amostras é testado

um par de parâmetros dado por um indivíduo evolucionário, fica evidente a grande demanda de esforço computacional por parte da metodologia utilizada. O algoritmo de preenchimento é de complexidade quadrática, enquanto que o algoritmo FCM é de complexidade $O(Nic^2) \approx O(N)$ [6, 28, 32, 29]. Portanto, tem-se o comportamento global da metodologia ditado em maior medida pela avaliação da função de custo. O FCM pode assumir maior participação que o preenchimento no esforço computacional demandado caso não haja limitação do número de grupos a serem obtidos e nem do número de iterações do seu processo interno de otimização. No presente caso, os valores padrão dos parâmetros do método foram mantidos, deixando irrestrito o número de grupos possíveis em cada base de dados.

3.1.1 Métodos evolucionários de otimização

A criação das populações iniciais permitiu que os parâmetros σ e c em cada indivíduo assumissem valores entre 0,1 e 50. Contudo, tomou-se o cuidado de não permitir a cada iteração que algum indivíduo da população de soluções candidatas possuísse valores menores que 0,1 para σ (evitando *kernel* esparso) e 2 (mínimo permitido pelo FCM) para c. Caso isso ocorresse por efeito dos cruzamentos e das mutações dos indivíduos, o parâmetro abaixo do limite mínimo assumia o menor valor permitido para a variável.

Nos dois métodos utilizados, além da condição de parada por número máximo de iterações ou valor de A muito próximo de 1, foi adicionada uma condição de parada em função da estabilidade da melhor solução. Estabeleceu-se que após a décima iteração, o par de parâmetros candidato à resposta até então passa a ser comparado aos cinco pares anteriores de melhor desempenho. Ou seja, o valor de cada um dos parâmetros é comparado com a média dos cinco valores das iterações antecedentes e que geraram a melhor solução. Portanto, o algoritmo de otimização pode terminar antes do número máximo permitido de iterações se na melhor solução após a décima iteração cada um dos valores de σ e c do par candidato é igual à média dos valores correspondentes das cinco iterações imediatamente anteriores. A escolha dessa regra deu-se por dois motivos. O primeiro se deve ao fato de a inclusão de regras de parada deste tipo serem prática comum em algoritmos evolutivos quando esses últimos são aplicados em problemas em que as soluções sofrem pequena variação quando se aproximam do resultado desejado. Durante a realização dos experimentos, observou-se em alguns casos a estabilização da resposta após a décima iteração. Por ser muito custosa a realização de cada iteração, optou-se por interromper o algoritmo quando os melhores valores encontrados para os parâmetros não se alterassem.

Normalmente, a regra adicional de parada leva em consideração o valor da

avaliação da função objetivo e não o valor dos parâmetros. No entanto, foi possível notar que para um mesmo par de valores σ e c, a avaliação da função objetivo retorna valores ligeiramente diferentes. Como o FCM depende de um processo de otimização que retorna resultados semelhantes (mas não iguais) para um mesmo conjunto de entradas, a função objetivo também sofre de pequena alteração dos seus resultados para os mesmos dados. Logo, o segundo motivo da adoção da regra adicional com foco nos valores dos parâmetros e não da função objetivo justifica-se pela pequena variação dos resultados quando é atingida a melhor solução.

As subseções a seguir detalham as implementações feitas para os métodos de otimização utilizados. Os parâmetros escolhidos para os métodos, apresentados na Tabela 3.1, possibilitam a busca não exaustiva da solução desejada.

Tabela 3.1: Parâmetros dos algoritmos evolutivos utilizados na otimização.

Parâmetro	AG	PSO
Tamanho da população	50	
Máximo de gerações	30	
Valores mínimo e máximo das variáveis	0,1 e 50	
Porcentagem de cruzamento	80,0	-
Porcentagem de mutação	6,0	-
Inércia (ine)	-	1,0
Fator de correção (cor)	-	2,0

Algoritmo Genético (AG)

Um Algoritmo Genético simples foi implementado com a finalidade de satisfazer o problema de otimização formulado. Além de o AG possuir elitismo (preservando na população seguinte o melhor indivíduo da geração atual), foi escolhida a codificação real das variáveis (os genes guardam valores reais das variáveis). Não foi utilizado o operador de seleção, deixando apenas para o cruzamento dos indivíduos mais aptos e a mutação dos piores a incumbência de gerar nova população de soluções candidatas a cada geração.

Particle Swarm Optimization (PSO)

O algoritmo implementado teve como base o código [21], em que foram redefinidas as condições de parada e a geração da população inicial.

3.2 Bases de dados utilizadas nos experimentos

Seguindo o procedimento adotado em [23, 7], de todas as bases de dados utilizadas foram retiradas as amostras sem valores para uma ou mais caracte-

rísticas. Cada experimento contou com um conjunto específico de bases como será detalhado nas subseções a seguir.

3.2.1 Experimento 1

Em [36], três bases de teste foram geradas. Em cada uma delas, cada uma das duas classes foi constituída por amostras de distribuições normais com centros diferentes e desvio padrão entre 0,19 e 0,35. Utilizando os mesmo parâmetros para a geração desses conjuntos, três bases de teste foram utilizadas no primeiro experimento. Cada conjunto é caracterizado na Tabela 3.2, com complementação das informações na Tabela 3.4.

Tabela 3.2: Características das bases de dados do Experimento 1.

Grupo	Número de funções geradoras por classe	Parâmetro σ em [36]
1	1	0,59
2	2	0,68
3	3	0,66

3.2.2 Experimentos 2 e 3

É objetivo principal do segundo experimento avaliar a metodologia quanto à capacidade de encontrar parâmetros que permitam aos classificadores cumprir satisfatoriamente a classificação das amostras e conservar a estrutura dos dados. Espera-se poder estabelecer no terceiro experimento relações entre os resultados e informações geográficas reais.

As seguintes bases de dados de *UCI* [3] para classificação binária foram utilizadas nos testes:

1. Statlog (Australian Credit Approval) (acr);
2. Liver Disorders (bld);
3. Statlog (German Credit Data) (gcr);
4. Heart Disease (hea);
5. Ionosphere (ion);
6. Pima Indians Diabetes (pid);
7. Connectionist Bench (Sonar, Mines vs. Rocks) (snr);
8. Tic-Tac-Toe Endgame (ttt);
9. Breast Cancer Wisconsin (Original) (wbc);

Mais informações sobre as características de cada base são apresentadas nas Tabelas 3.3, 3.4 e 3.5. Além de incluir a base *img*, a Tabela 3.3 exibe os valores obtidos por [23] para as bases de dados reais utilizadas nos testes. Para estabelecer comparações com os resultados da metodologia proposta neste trabalho, foram utilizados como referência os valores de σ obtidos e citados em [23].

Tabela 3.3: Bases de dados e valores de referência [23] para σ.

Base	Abreviação	σ LS-SVM	σ SVM
Australian	acr	22,75	12,43
Bupa	bld	41,25	9
German	gcr	31,25	55
Heart	hea	5,69	7,15
Ionosphere	ion	3,3	3,3
Pima Indians Diabetes	pid	240	15,5
Sonar	snr	33	5,09
Tic Tac Toe	ttt	2,93	9
WDBC	wbc	6,97	19,5
Índices Minas Gerais	img	-	-

A Tabela 3.4 apresenta de forma comparativa as partições de amostras em cada validação cruzada e as principais características de todas as bases. Para testar a metologia em várias condições, percebe-se haver uma variedade considerável quanto ao tamanho e às características das bases em cada um dos experimentos. É preciso destacar que de cada número total de amostras N de cada base, N_{CV} foram utilizadas para treinamento na validação cruzada e N_{test} foram utilizadas como teste de generalização. A Tabela 3.4 ainda apresenta separadamente o número de características numéricas n_{num} e categóricas n_{cat}.

Tabela 3.4: Características das bases de dados.

	1	2	3	acr	bld	gcr	hea	ion	pid	snr	ttt	wbc	img
N_{CV}	40	80	121	460	230	666	180	234	512	138	638	455	572
N_{test}	-	-	-	230	115	334	90	117	256	70	320	228	281
N	60	120	180	690	345	1000	270	351	768	208	958	683	853
n_{num}	2	2	2	6	6	7	7	33	8	60	0	9	7
n_{cat}	0	0	0	8	0	13	6	0	0	0	9	0	0
n	2	2	2	14	6	24	13	33	8	60	9	9	7

Por sua vez, a Tabela 3.5 expõe as proporções de cada conjunto de amostras utilizadas para treinamento, validação e teste dos classificadores binários LS-SVM ajustados por *10-fold cross-validation*. Percebe-se que em nenhum dos casos a proporção de dois terços dos dados foi atingida nos grupos de treinamento. Contudo, a parcela utilizada como validação é a mesma da metologia utilizada em [23] e o conjunto reservado ao teste é de tamanho considerável para comparar os desempenhos com o mesmo número de amostras

em cada uma das bases.

Tabela 3.5: Porcentagem das amostras das bases em cada grupo (treinamento, validação e teste).

Base	Treinamento (%)	Validação (%)	Teste (%)
acr	55,07	33,33	11,45
bld	43,48	33,33	22,9
gcr	58,6	33,4	7,9
hea	37,04	33,33	29,26
ion	43,87	33,33	22,51
pid	56,25	33,33	10,29
snr	27,88	33,65	37,98
ttt	58,25	33,4	8,25
wbc	54,9	33,38	11,57

Índices socioeconômicos que compõem a base img

Visando estabelecer um grande conjunto de informações recentes para estudos de agrupamento, é proposta neste trabalho a base com índices socioeconômicos apurados para todos os 853 municípios do Estado de Minas Gerais. O conjunto permite o relacionamento dos resultados com a localização geográfica dos municípios através da visualização dos mapas, além de possibilitar um desafio para os métodos dessa dissertação. A Tabela 3.6, bem como os tópicos a seguir, detalha a base de dados *img*.

Tabela 3.6: Composição da base de dados *img*.

Índice	Ano	Fonte
PIB per capita	2003	IBGE [13]
Domicílios	2000	IBGE/IPEADATA [13, 17]
IDH	2000	PNUD/IPEADATA [12, 17]
IMRS	2004	DataGerais [14, 35]
Luz Elétrica	2000	IBGE/IPEADATA [13, 17]
PIB	2003	IBGE/IPEADATA [13, 17]
População	2000	IBGE/IPEADATA [13, 17]

Índice Mineiro de Responsabilidade Social - IMRS

O IMRS é um índice que busca expressar a situação do desenvolvimento social de cada município de Minas Gerais procurando mensurar os impactos das ações dos governos estadual e municipal, da sociedade civil e do mercado. Sua elaboração tenta suprir a demanda da população e dos administradores, analistas e formuladores de políticas públicas de desenvolvimento por dados que traduzam a informação de forma clara, agregada e sintética, com abrangência, comparabilidade, confiabilidade, reprodutividade e periodicidade

adequadas. Conforme determina o terceiro artigo da ementa da lei estadual 14.172 de 2002, a Fundação João Pinheiro (FJP) é a instituição responsável pela elaboração do IMRS desde a criação desse. A FJP conta com o apoio da Secretaria de Planejamento e Gestão (SEPLAG) para utilização e aperfeiçoamento das informações municipais.

Por definição, o índice, normalizado entre 0 e 1, deve contemplar nove dimensões: renda, saúde, educação, demografia, segurança pública, gestão, habitação e meio ambiente, cultura e desporto e lazer. Para cada uma das dimensões foram selecionados índices que retratam a situação em nível municipal de modo a avaliar a atuação de gestão pública e as iniciativas vinculadas à participação nas decisões em programas e políticas públicas prioritárias. Sendo assim, visam responder se houve sucesso das ações governamentais avaliadas. Ou melhor, se os governos federal e estadual atenderam adequadamente o município ou se a administração municipal é que não foi pró-ativa em responder aos programas estabelecidos nas esferas superiores. Para atingir esse objetivo, o reconhecimento da fragilidade da base de dados suscitou a escolha de cerca de 40 indicadores a fim de que a sua agregação desse consistência ao IMRS e compensasse as deficiências individuais de cada um. Espera-se que em sua próxima edição, o IMRS tenha seu cálculo baseado em um número muito menor de índices mantendo os seus fundamentos, dimensões e temas e reforçando a comparabilidade entre os municípios e a sua representatividade frente ao objetivo proposto.

Produto Interno Bruto - PIB

O Produto Interno Bruto (PIB) consiste na soma dos valores financeiros a preços de mercado de toda a produção econômica de bens e serviços (bens finais) gerados dentro de uma determinada região e/ou segmento da sociedade em um período de um trimestre ou um ano. É um importante indicador da atividade econômica e pode ser considerado a medida individual mais importante numa economia por equivaler à renda gerada. Há várias formas de calculá-lo, podendo ser dividido em PIB nominal, que mede o valor dos bens e serviços pelo valor atual de mercado, e em PIB real, que tenta medir o volume físico do produto. Sendo assim, mantendo-se a produção física inalterada, variações nos preços dos produtos alteram, em igual proporção, o PIB nominal, enquanto o PIB real fica inalterado. Já o PIB per capita é o produto ou renda média dos habitantes da região geográfica pesquisada.

A contabilização do PIB em cada unidade da federação e de cada município faz parte do projeto de Contas Regionais do Instituto Brasileiro de Geografia e Estatística (IBGE) em parceria com os Órgãos Estaduais de Estatística, Secretarias Estaduais de Governo e a Superintendência da Zona Franca de Manaus.

Em Minas Gerais, a parceria se dá entre o IBGE e a Fundação João Pinheiro. Cada conjunto de dados disponível possui uma unidade de referência para calculá-lo. Por exemplo, os dados do IBGE para PIB a preços correntes podem ser encontrados em 1000 Reais enquanto que os de PIB Per capita, em Reais. Por sua vez, os dados do IPEADATA podem ser obtidos em 1000 Reais de 2000. Além de cobrir a dimensão econômica do comportamento dos municípios do estado, a sua série possui dados para todos os municípios e possui metodologia consolidada. É evidente nos meios de comunicação e nos estudos de desenvolvimento de todos os setores da sociedade a grande importância que esse índice assume como medida da atividade econômica de uma região.

População

Entende-se como população um conjunto de pessoas que habitam uma determinada área geográfica. Deve haver pelo menos um elemento em comum entre os elementos desse conjunto que compõe o universo pesquisado. E é possível classificar a população total em residente (ou população de direito), que consiste no número de pessoas moradoras no domicílio ainda que ausentes na realização da pesquisa, e em população presente (ou população de fato), que consiste no número de pessoas presentes no domicílio ainda que não sejam moradoras do mesmo.

Dados como a população total (número de habitantes) de uma região são obtidos pela realização de recenseamentos (por exemplo, o Censo demográfico) e por aproximações. No Brasil, o Instituto Brasileiro de Geografia e Estatística está encarregado da condução e do emprego de tais metodologias. Embora a apuração desse dado se dê a cada década, com complementação a cada cinco anos e com aproximação por modelos estatísticos, há uma série de informações que podem ser extraídas a partir dos valores obtidos e em comparação com a série histórica. O valor absoluto e a variação entre os valores apurados para uma mesma localidade em anos diferentes são dados importantes. Além de expor a situação atual de uma região, capturam parte da dinâmica populacional que ocorreu. Daí se justificam a sua apuração e utilização em políticas públicas e privadas de desenvolvimento sócio-econômico nas regiões pesquisadas.

Um característica importante desses dados é a disparidade entre as magnitudes de população e de variação da população em algumas localidades frente ao comportamento da grande maioria dos municípios do Estado de Minas Gerais. Por esse motivo, deve-se ter atenção na normalização dos dados e a na escolha de um ano em que a amostragem apurou dados para todas as localidades no Estado.

Número de Domicílios

Entende-se por domicílio o local de moradia, estruturalmente separado e independente, constituído por um ou mais cômodos. A separação fica caracterizada quando o local de moradia é limitado por paredes, muros, cercas, etc., coberto por um teto, e permite que seus moradores se isolem, arcando com as suas despesas de alimentação e moradia. A independência fica caracterizada quando o local de moradia tem acesso direto, permitindo que seus moradores possam entrar e sair sem passar por local de moradia de outras pessoas. É decenal a publicação dos dados referentes ao número de domicílios. Logo, como esperado para séries constituídas por um pequeno número de pontos amplamente espaçados temporalmente, o que se tem são informações pontuais a cada período de amostragem.

Número de Domicílios com Iluminação Elétrica e Eletricidade

O Número de Domicílios com Iluminação Elétrica e Eletricidade fornece a quantidade de residências que possuem instalações elétricas em cada município. Este índice é relevante para uma série de estudos socioeconômicos uma vez que fornece a evolução do número de consumidores residências. Sabe-se que há uma forte correlação entre a taxa de eletrificação residencial e os demais indicadores socioeconômicos, o que confirma a importância do índice em questão. Apesar disso, a série de dados disponível dificulta o uso do mesmo em estudos de longo prazo por ser uma série com medição decenal.

Índice de Desenvolvimento Humano - IDH

Erroneamente apontado como representação da felicidade das pessoas ou do melhor lugar para se viver, o Índice de Desenvolvimento Humano (IDH) não contempla todos as dimensões de desenvolvimento, mas visa servir de base para análises que não utilizam apenas o PIB per capita como medida. As Nações Unidas e o Governo Federal Brasileiro utilizam a apuração desse índice para nortear as políticas públicas de promoção do desenvolvimento humano. Considerando apenas a série nova de IDHM para municípios, em que há valores para todos os municípios do estado, é um índice importante e bem estruturado.

As etapas necessárias à apuração do IDH, índice normalizado entre 0 e 1, partem do PIB per capita das localidades sendo então corrigido pelo poder de compra segundo a moeda de cada país. Em seguida, o referido PIB é calculado em dólar paridade poder de compra (ppc), excluindo assim as diferenças de custo de vida entre os países. O terceiro passo consiste em computar os dados de longevidade da população tendo como base a expectativa de vida

ao nascer. Por fim, avalia-se a educação pelo índice de analfabetismo e pela taxa de matrícula em todos os níveis de ensino. O Índice de Desenvolvimento Humano Municipal (IDH-M) resulta então da média aritmética simples dos subíndices Longevidade (IDH-Longevidade), Educação (IDH-Educação) e Renda (IDH-Renda).

3.3 Conclusões do capítulo

Como pretendeu-se apresentar neste capítulo, além das modificações efetuadas em cada um dos métodos para lidar com grandes bases de dados, foram utilizados diferentes conjuntos de amostras para evidenciar as características dos métodos em desafios diferentes e complementares. Tendo definidas neste capítulo as condições em que se deram os experimentos com bases sintéticas e bases reais, no próximo capítulo, os resultados são expostos em detalhes possibilitando a análise do desempenho e das características dos métodos.

CAPÍTULO

4

Resultados dos Experimentos

Os resultados apresentados neste capítulo seguem os procedimentos descritos no capítulo anterior para a aplicação dos métodos às bases escolhidas. Espera-se conseguir destacar as características dos métodos através da análise feita separadamente para cada teste.

4.1 Experimento 1

No primeiro experimento realizado, três conjuntos de teste foram gerados seguindo as mesmas distribuições de [36]. A Figura 4.1 apresenta exemplos das distribuições das três bases de duas classes. Percebe-se que não há sobreposição das distribuições das classes.

Aplicando as bases aos algoritmos de otimização, é possível notar, na Tabela 4.1, que os métodos conseguiram maximizar a função de custo. Em todos os casos, tanto os valores médios quanto os valores medianos apurados nas dez validações cruzadas aproximaram-se da unidade. Ou seja, as matrizes obtidas alcançaram elevado alinhamento empírico, muito próximo do alinhamento completo. Vale destacar que os valores alcançados pelo PSO são maiores que os obtidos pelo AG.

Na Tabela 4.2, que apresenta apenas os resultados do Algoritmo Genético, os valores de σ foram mais altos e com maior desvio padrão quando os dados foram normalizados. Com relação ao número de grupos, em nenhum dos casos os valores encontrados foram iguais ao número de funções geradoras. Porém, nas bases 2 e 3, os valores de c foram iguais ao número de classes.

Os resultados da Tabela 4.3 são melhores com relação ao número de grupos. Enquanto na base 1 a maioria dos valores encontrados é igual ao de

Tabela 4.1: Valores médio μ e mediano Md de A obtidos por AG e por PSO para os dados de teste.

	AG				PSO			
	Orig.		Norm.		Orig.		Norm.	
Base	A_μ	A_{Md}	A_μ	A_{Md}	A_μ	A_{Md}	A_μ	A_{Md}
1	0,9196	0,9164	0,9524	0,9613	0,9577	0,9546	0,9681	0,9635
2	0,9403	0,9409	0,9399	0,9418	0,9481	0,9499	0,9514	0,9504
3	0,9576	0,9586	0,9332	0,9300	0,9674	0,9673	0,9687	0,9755

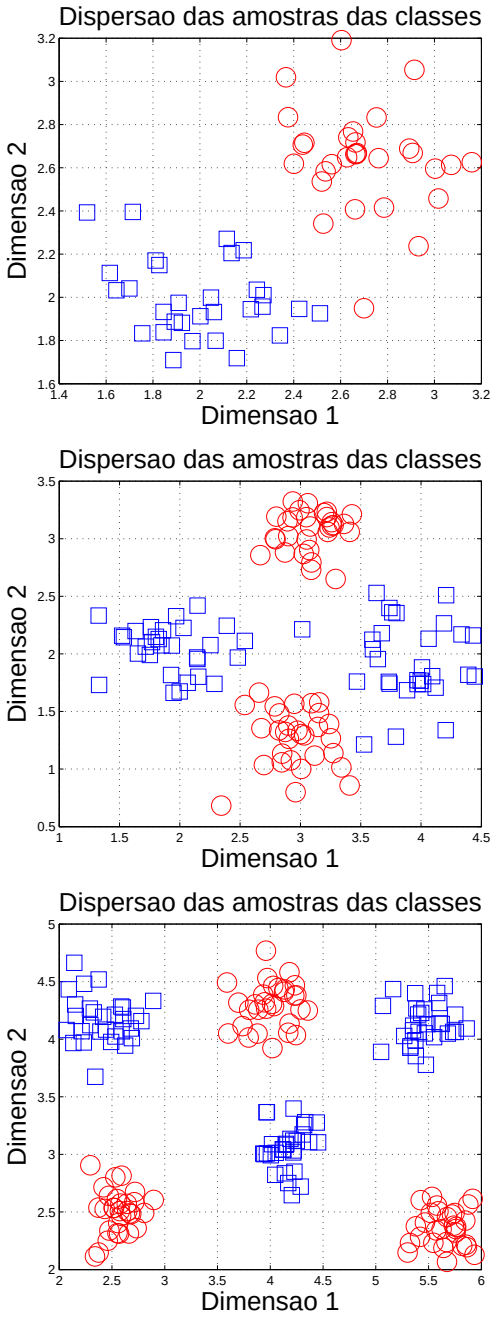

Figura 4.1: Dispersão das amostras das bases similares às de [36].

Tabela 4.2: Valores médio μ, desvio padrão e mediano Md de σ e de número de grupos encontrados pelo Algoritmo Genético com dados de teste originais e normalizados.

	Originais			Padronizados		
Base	σ	c_μ	c_{Md}	σ	c_μ	c_{Md}
1	0,4899 (0,4026)	8	4	0,7709 (0,7776)	16	12
2	2,1051 (1,2924)	2	2	2,2623 (1,5896)	9	2
3	1,9497 (0,7601)	3	2	4,5589 (3,7163)	2	2

Tabela 4.3: Valores médio μ, desvio padrão e mediano Md de σ e de número de grupos encontrados pela Otimização por Enxame de Partículas com dados de teste originais e normalizados.

	Originais			Padronizados		
Base	σ	c_μ	c_{Md}	σ	c_μ	c_{Md}
1	2,4316 (5,9509)	2	2	0,9619 (0,5962)	10	2
2	1,4747 (0,2923)	2	2	2,0987 (0,7747)	2	2
3	1,8085 (0,6051)	3	2	1,0412 (0,7103)	5	6

funções geradoras dos conjuntos, na base 2, os valores são iguais o número de classes. Na base 3, apenas os dados padronizados permitiram a obtenção do número de funções geradoras. Os valores médios de σ são diferentes nos três casos, mas próximos entre si nas bases 2 e 3.

A ordenação dos grupos em cada um dos conjuntos foi escolhida de modo a apresentar em sequência as amostras de cada grupo e os grupos de cada classe. Dessa forma, pode-se comparar o desempenho dos métodos de ordenação das amostras em cada uma das matrizes. A coloração de cada ponto da matriz é função do valor da similaridade e da proximidade entre um par de amostras. Quanto maiores são a similaridade e a proximidade, maior é o valor do elemento na matriz e mais avermelhada é a cor relativa atribuída ao ponto nas matrizes das figuras. Para os casos em que há pouca similaridade e proximidade, o valor é menor e, por consequência, mais azulada é a cor reservada ao valor nas matrizes. Nas matrizes ordenadas, os pontos com coloração mais avermelhada (grande afinidade inter grupos) devem ser agrupados nas submatrizes de afinidade sobre a diagonal principal, enquanto que os pontos de coloração azulada (pouca afinidade) e amarelada (afinidade relevante mas não tão elevada entre grupos) devem ficar fora da diagonal da matriz. Vale destacar que como a escala de cores não é fixa, as cores atribuídas aos valores presentes nas matrizes pode variar significativamente entre matrizes de *kernel* e de proximidade ordenadas resultantes de métodos diferentes.

A Figura 4.2, que apresenta os resultados do AG para dados originais, permite a identificação dos dois conjuntos de classes apenas nas duas matrizes superiores. Por sua vez, os dados padronizados possibilitaram que a maioria das matrizes da Figura 4.3 permitissem melhor identificação visual das

duas classes. Nas duas figuras, a matriz de *kernel* e o método Minus foram melhores que os seus concorrentes.

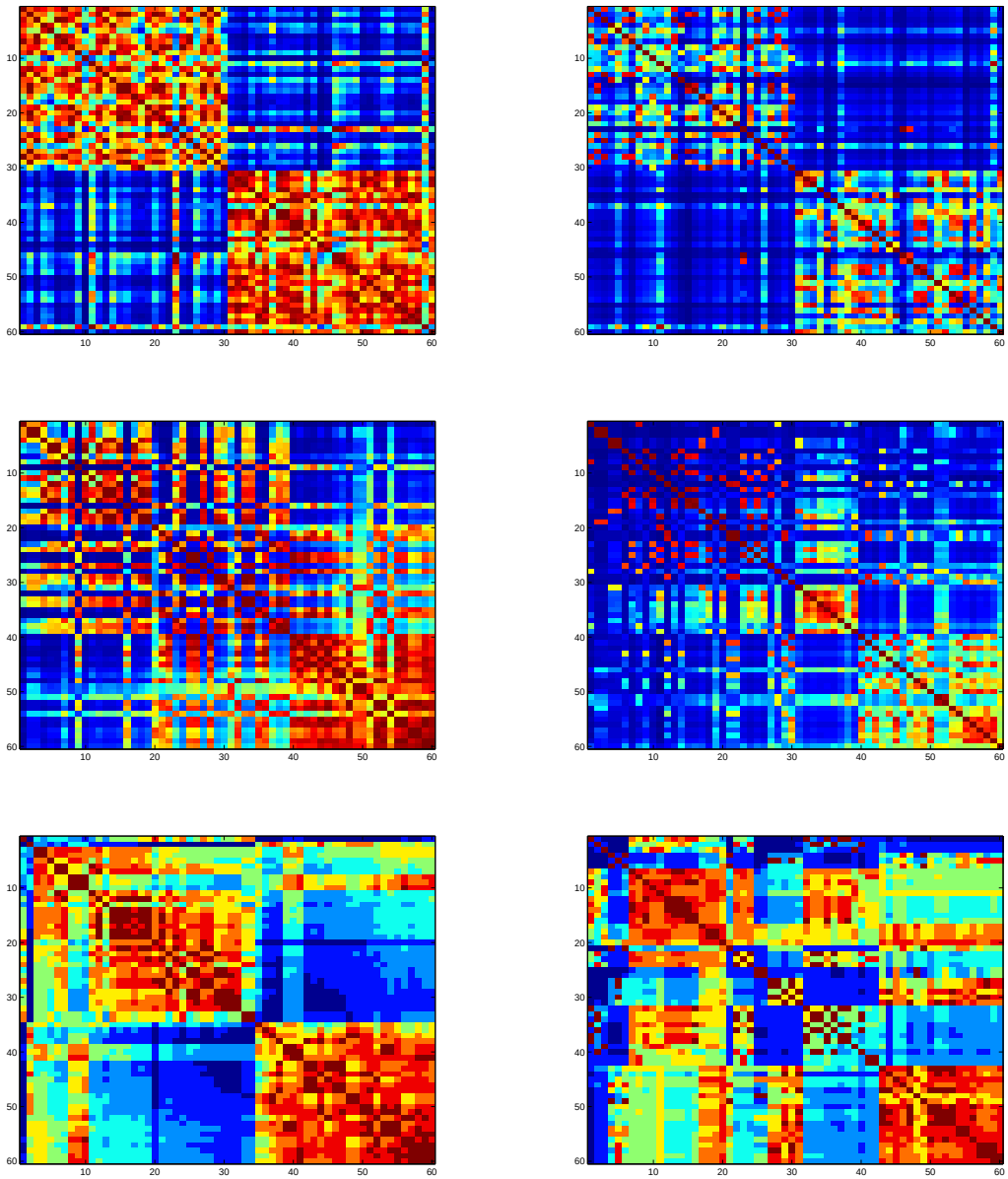

Figura 4.2: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base 1 original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

Para a Figura 4.4 e a Figura 4.5, resultantes dos parâmetros encontrados por PSO, a padronização dos dados não melhorou a visualização dos dois grupos. Apenas no caso dos dados ordenados sem padronização, ficou mais clara a existência de dois grupos. Ao contrário do caso anterior, foram as matrizes de proximidade que permitiram o melhor resultado.

A base 2, cujos resultados do AG estão apresentados nas Figuras 4.6 e 4.7, as quatro funções geradoras ficaram melhor destacadas nas matrizes de

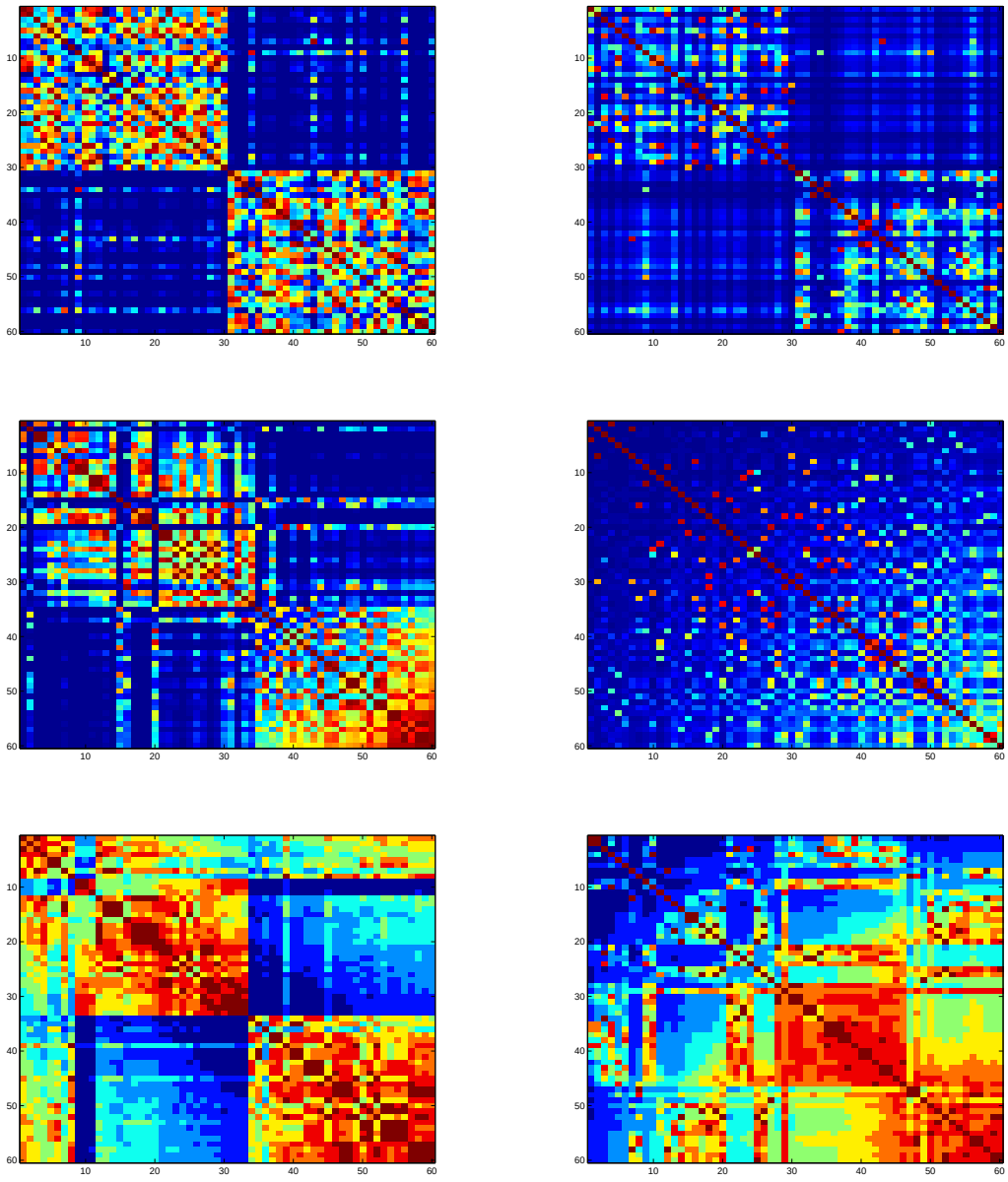

Figura 4.3: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base 1 padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

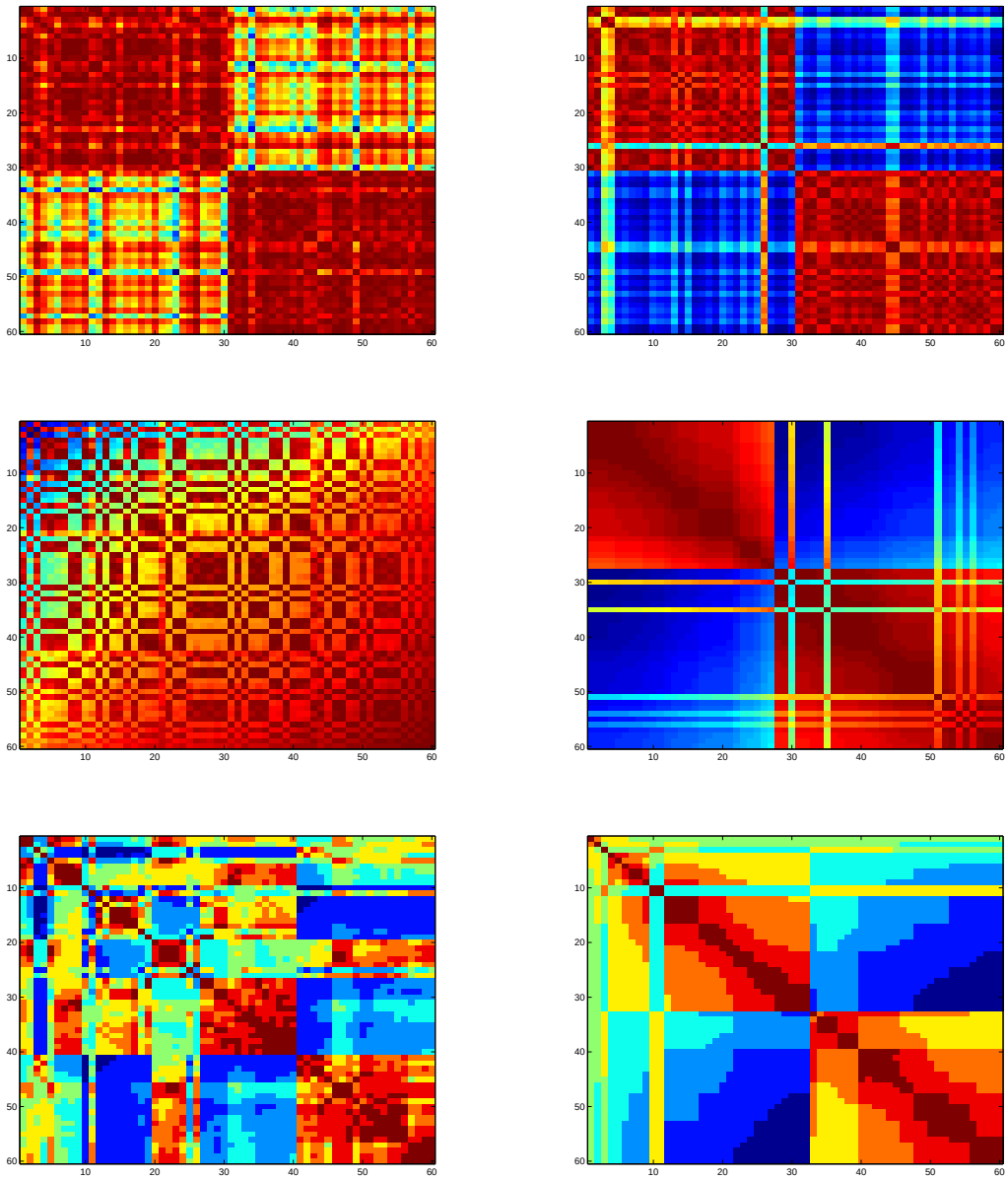

Figura 4.4: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base 1 original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

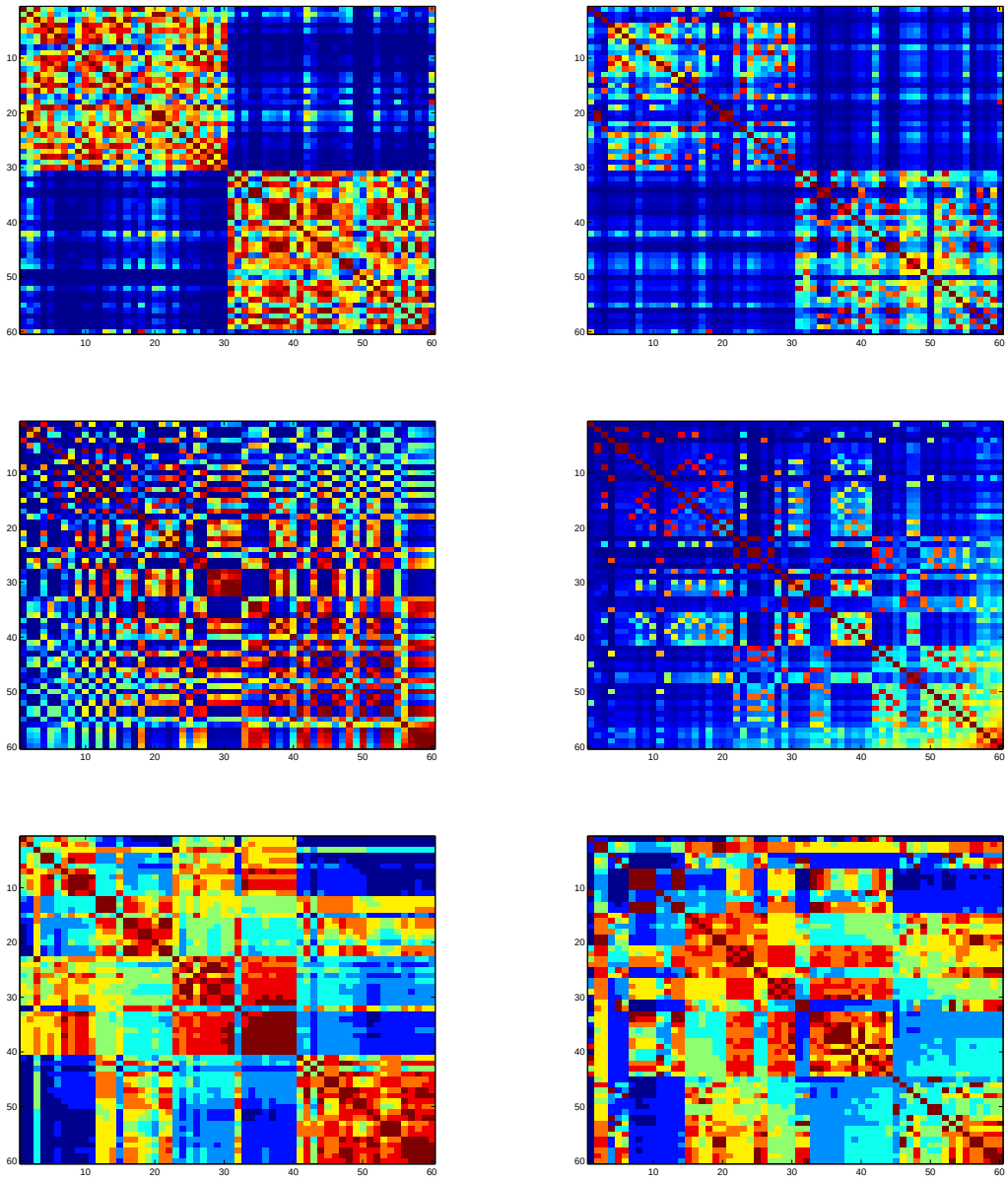

Figura 4.5: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base 1 padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

kernel ordenadas pelo Minus. Nas Figuras 4.8 e 4.9, até mesmo nas matrizes superiores a identificação do número correto de grupos não foi possível.

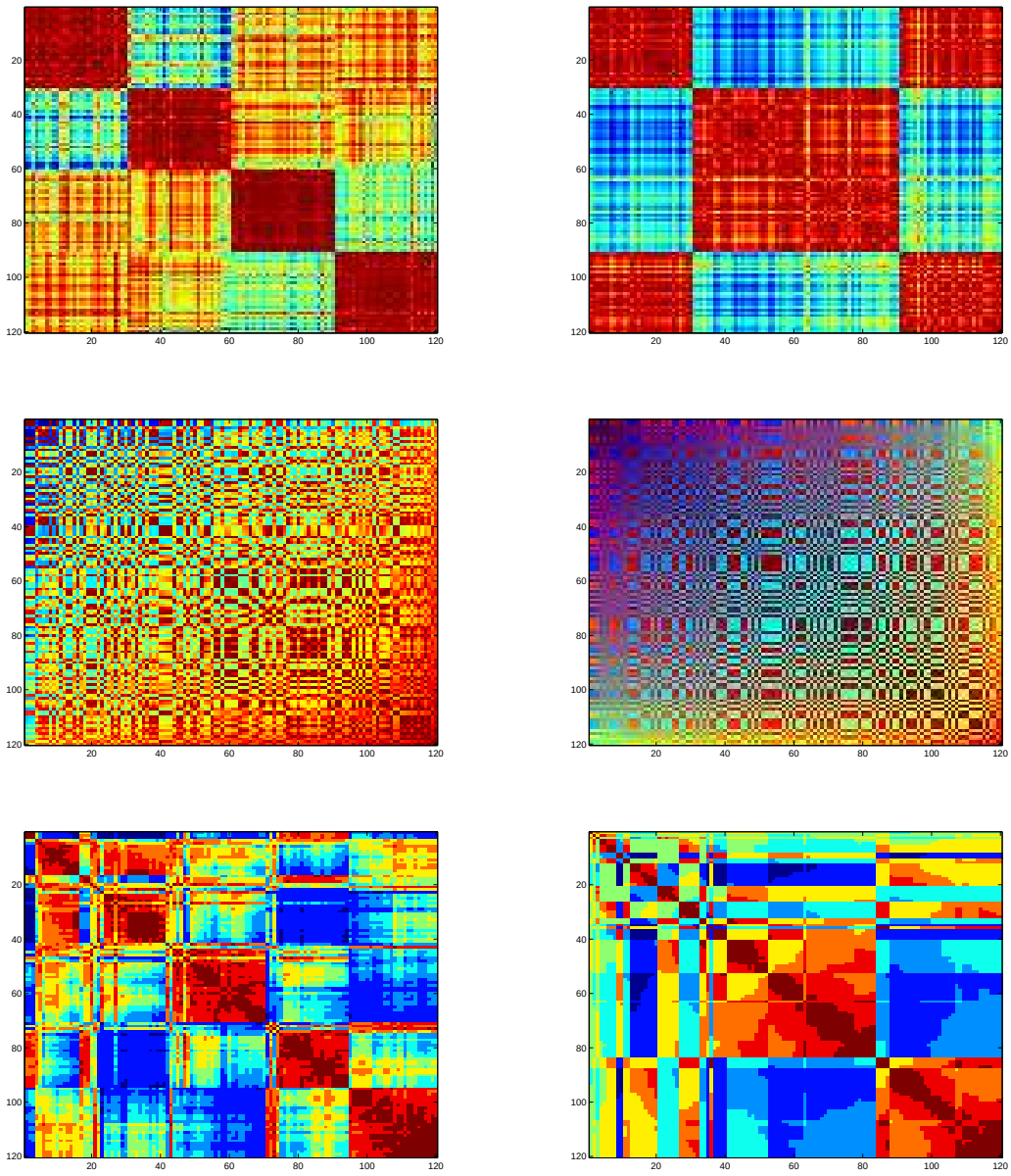

Figura 4.6: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base 2 original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

A matriz de kernel ordenada por Minus na Figura 4.10 permitiu melhor identificação das seis funções geradoras. Na Figura 4.11 entretanto, o resultado não é tão bom e tem qualidade comparável à da matriz de proximidade ordenada pelo mesmo método.

Para os resultados da base 3 com PSO, apenas a matriz de *kernel* ordenada por Minus nas Figuras 4.12 e 4.13 ficou próxima da original e o resultado desejável. As outras matrizes destes resultados, ainda que consigam reve-

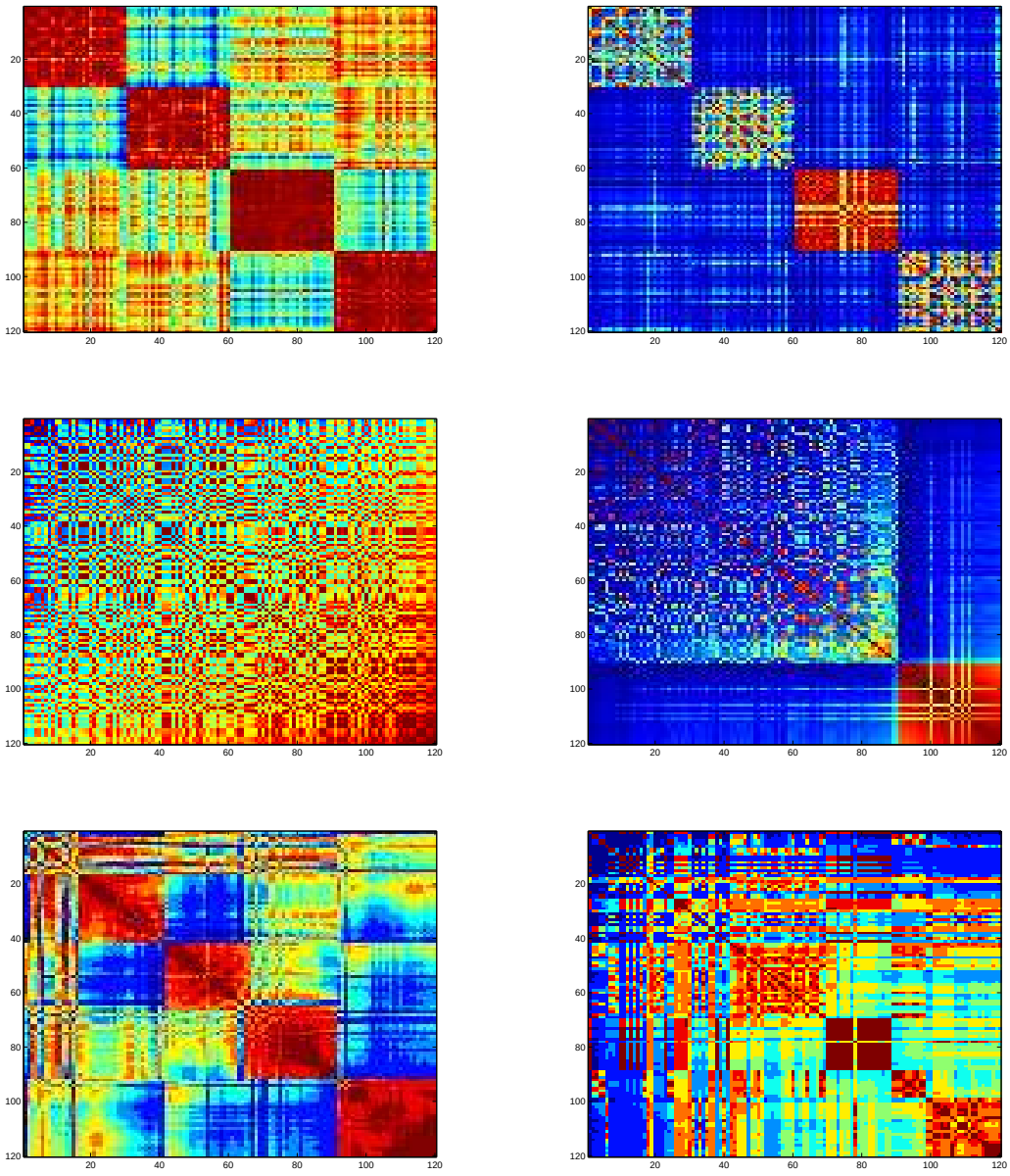

Figura 4.7: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base 2 padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

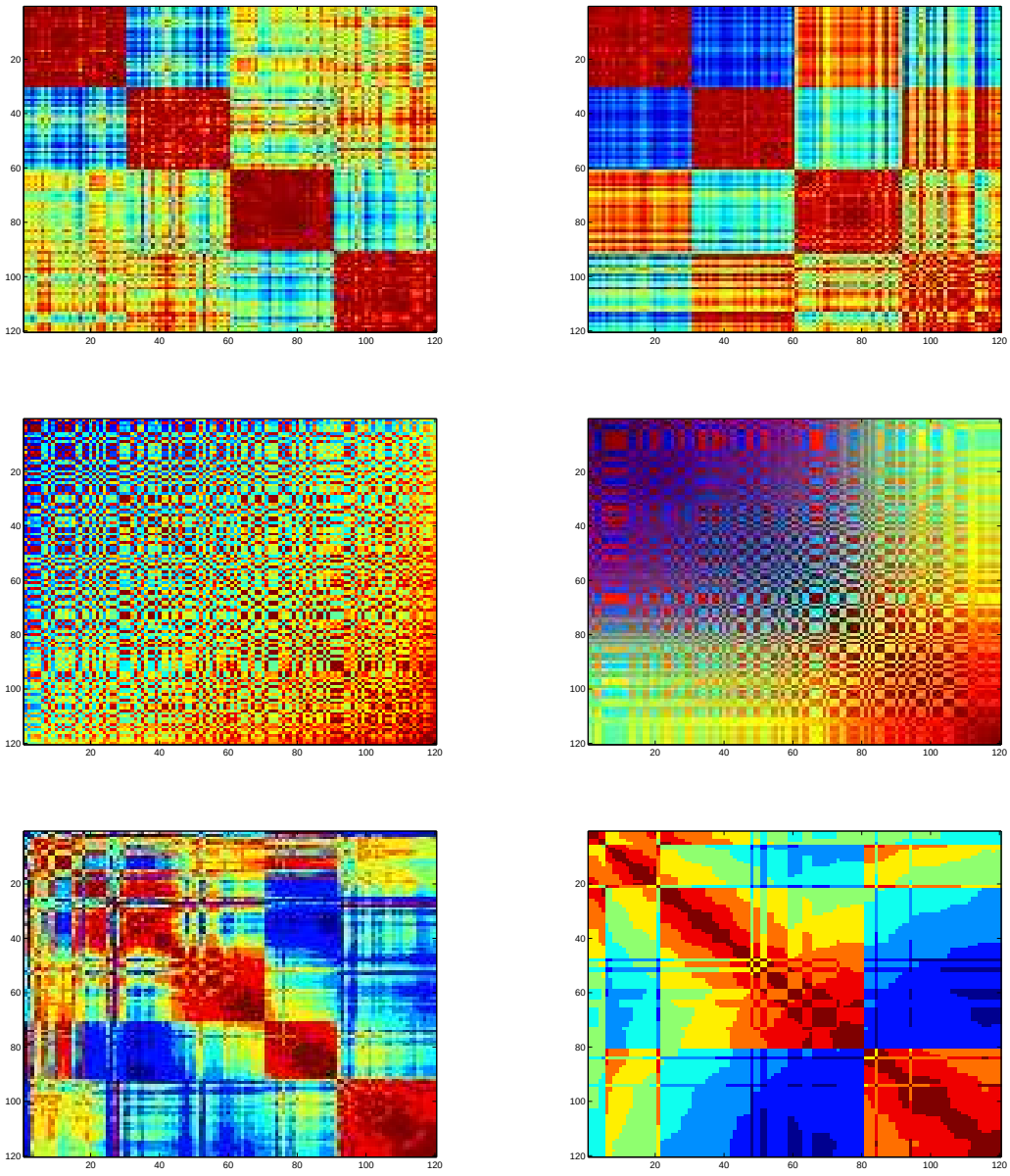

Figura 4.8: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base 2 original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

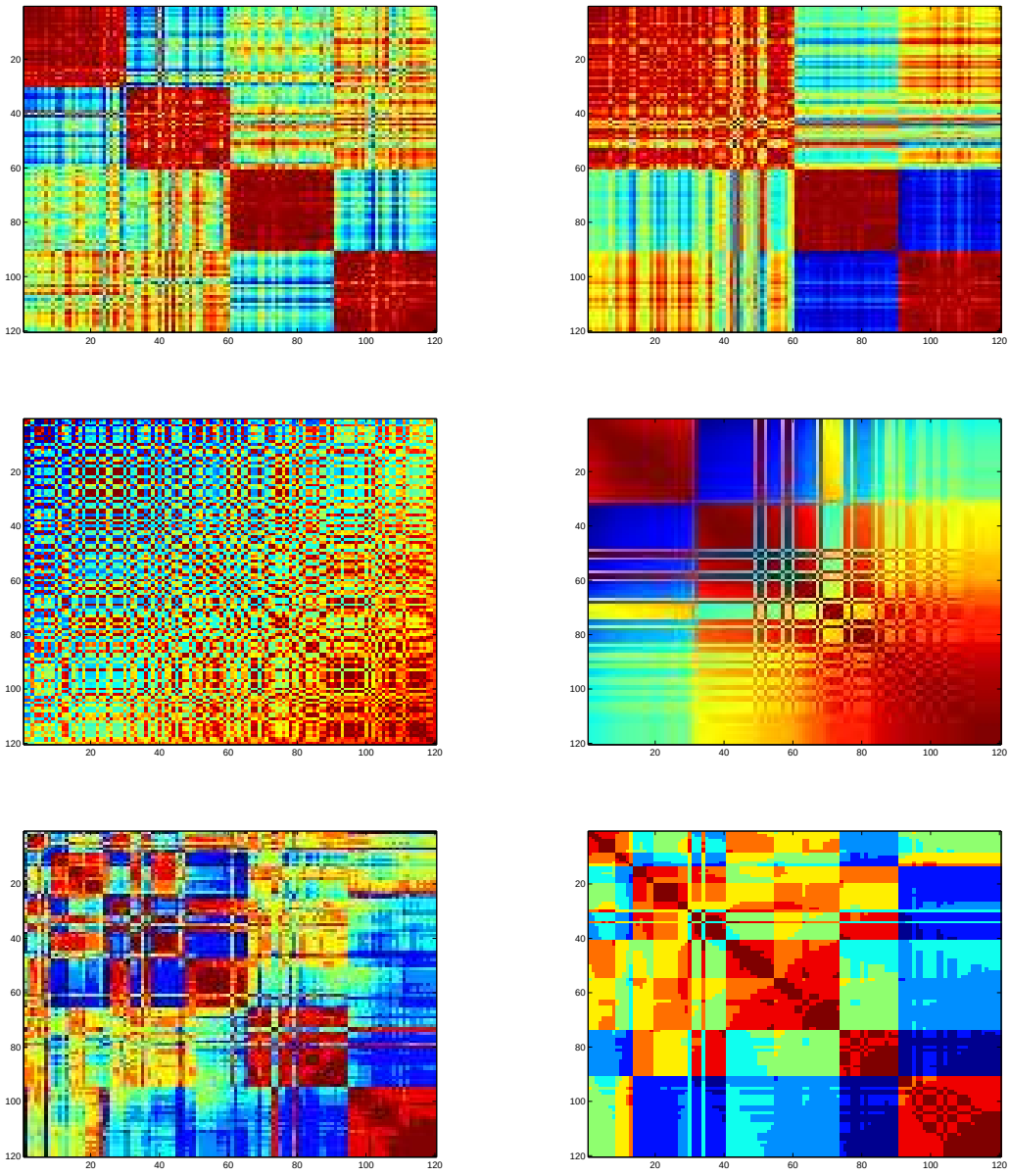

Figura 4.9: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base 2 padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

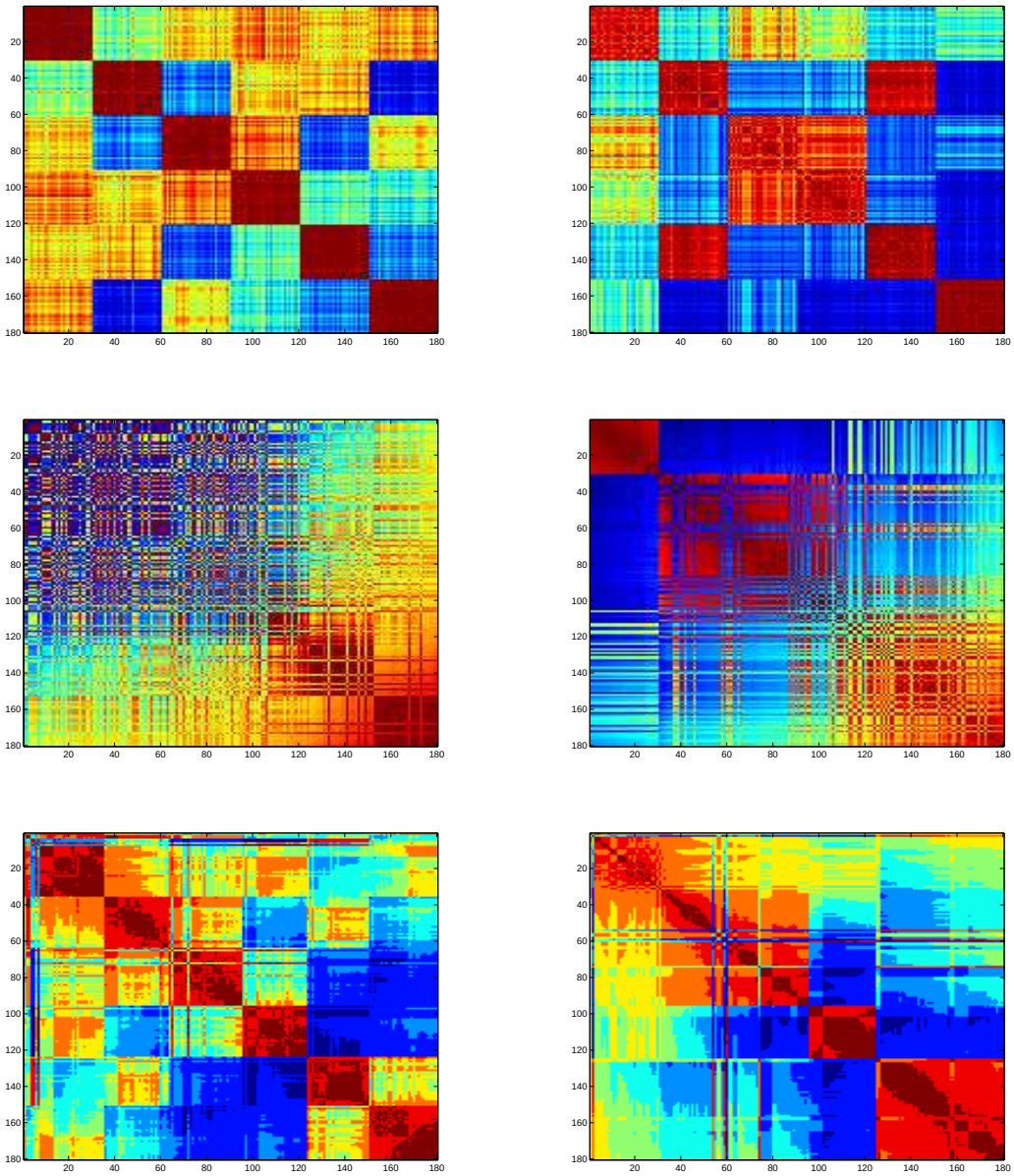

Figura 4.10: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base 3 original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

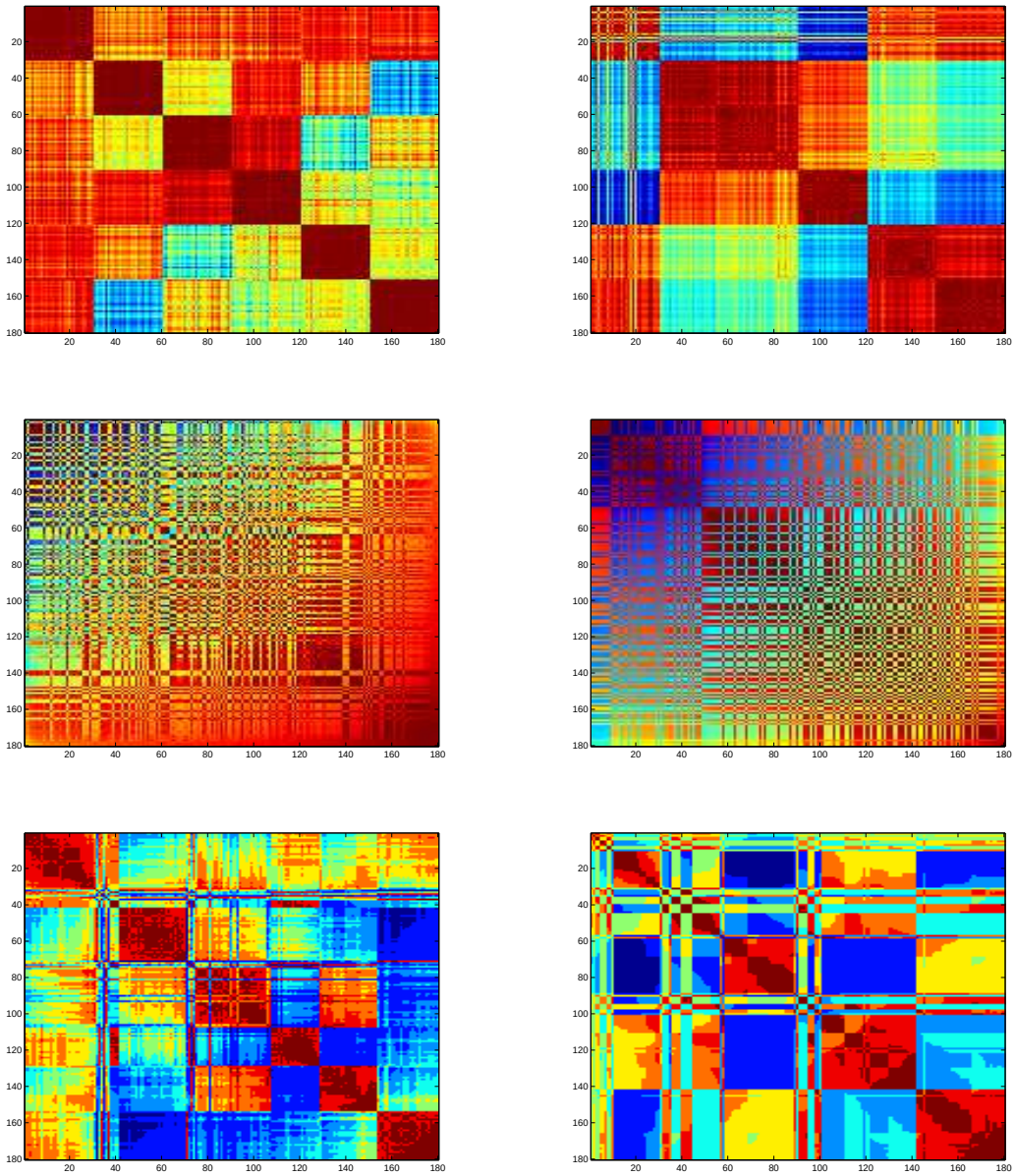

Figura 4.11: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base 3 padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

lar as funções geradoras (caso das matrizes originais) graças aos valores dos parâmetros, não permitem o mesmo quando são ordenadas pelo método do autovetor.

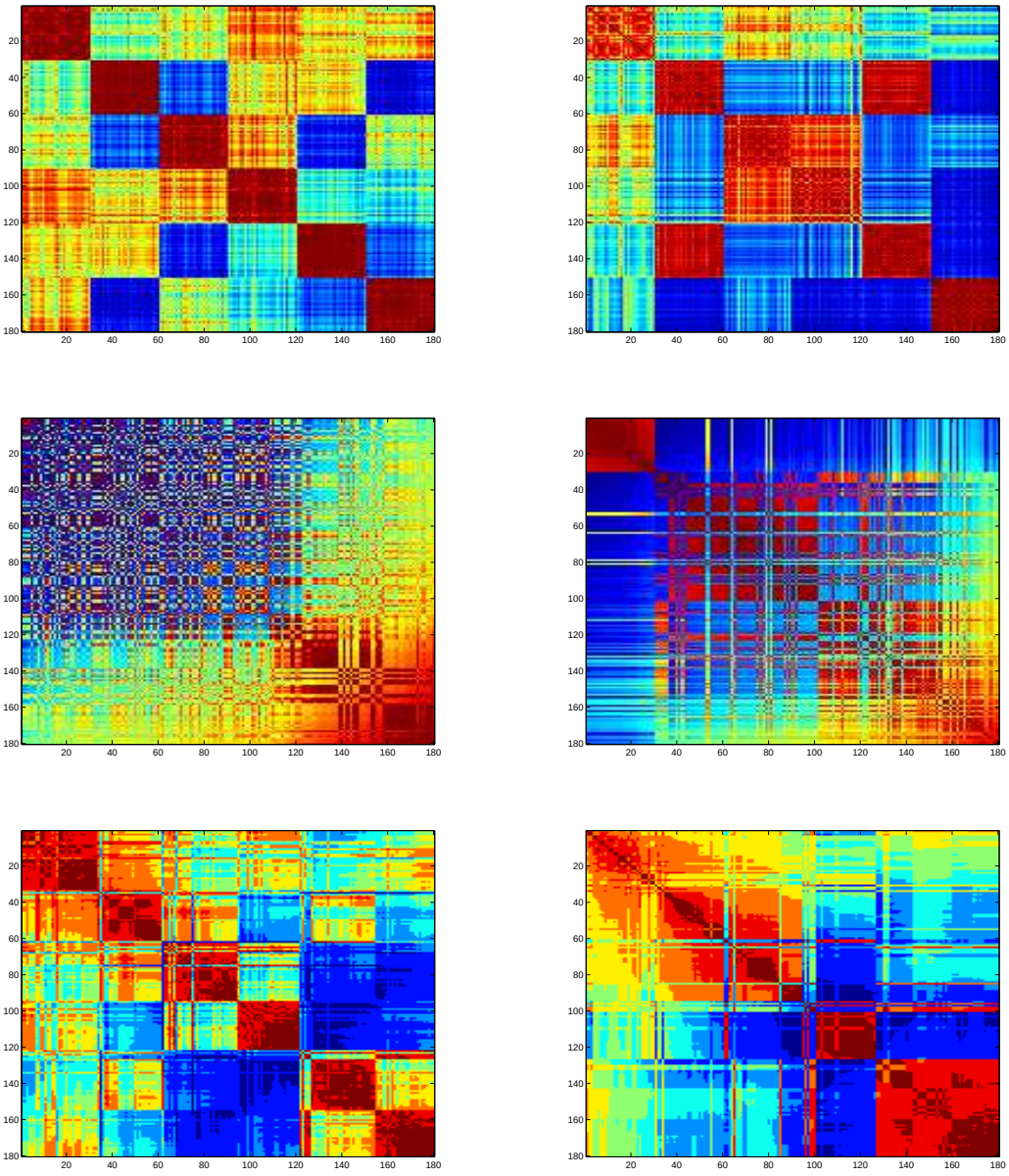

Figura 4.12: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base 3 original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

Percebe-se ao final deste experimento que ainda que os valores encontrados para os parâmetros não permitam a identificação correta das funções geradoras, a visualização das matrizes ordenadas se apresenta como boa ferramenta para agregar informação aos resultados. Comparando as matrizes originais com as ordenadas, percebe-se a superioridade do método Minus com nivelamento quartil, principalmente nas matrizes de *kernel*.

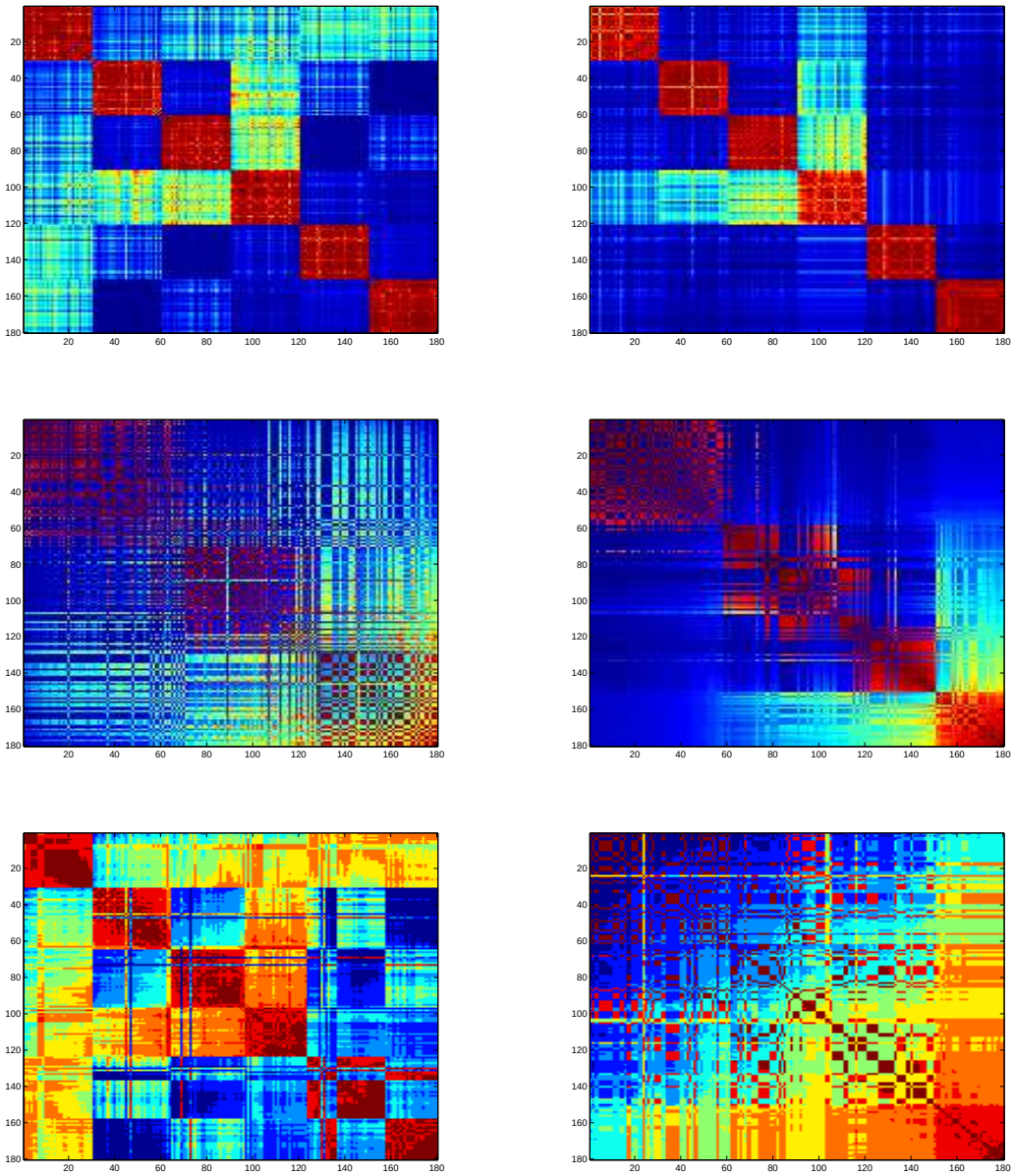

Figura 4.13: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base 3 padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

4.2 Experimento 2

No segundo experimento, que utiliza dados reais, a Tabela 4.4 revela que os algoritmos de otimização conseguiram atingir a unidade ou valores muito próximos de 1 na maioria das bases. Exceto para *img* e *acr*, os valores médios e medianos referentes tanto ao AG quanto ao PSO são próximos e elevados.

Tabela 4.4: Valores médio μ e mediano Md de A obtidos por AG e por PSO para os dados de referência.

	AG				PSO			
	Originais		Padronizados		Originais		Padronizados	
	μ	Md	μ	Md	μ	Md	μ	Md
acr	0,8673	0,8619	1,0000	1,0000	0,9350	0,9057	1,0000	1,0000
bld	0,9600	0,9616	0,9845	0,9851	0,9856	0,9863	0,9854	0,9856
gcr	0,9681	0,9715	1,0000	1,0000	0,9747	0,9744	1,0000	1,0000
hea	0,9108	0,9080	1,0000	1,0000	0,9474	0,9482	0,9998	1,0000
ion	0,9570	0,9576	0,9659	0,9654	0,9551	0,9568	0,9669	0,9661
pid	0,8785	0,8771	0,9995	1,0000	0,9638	0,9632	0,9997	1,0000
snr	0,9844	0,9832	0,9999	0,9999	0,9860	0,9867	1,0000	1,0000
ttt	0,9964	0,9965	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
wbc	0,9788	0,9789	0,9724	0,9772	0,9800	0,9795	0,9787	0,9785
img	0,1703	0,1668	0,9142	0,9129	0,6248	0,5056	0,9183	0,9197

Apenas as bases *bld* e *snr* tiveram resultados para σ na Tabela 4.5 muito próximos dos encontrados em [23]. As demais bases receberam parâmetros com valores muito aquém do trabalho de referência. Em geral, os valores médios encontrados para c são próximos aos medianos em todos os casos e, em sua maioria, baixos. Apenas as bases *acr* e *ttt* receberam valores elevados para o número de grupos mesmo quando os dados foram padronizados. Estes últimos resultados não ocorrem na Tabela 4.6, referente ao PSO. Em 29 dos 40 casos testados, é pequeno o valor encontrado para o número de grupos. Apenas em *acr* e *img*, os valores encontrados são maiores que 10, mas os resultados não se repetem para os dados padronizados. Os valores de σ são diferentes para os dados originais e padronizados em todas as bases. Pode ser notado que os desvios padrão obtidos pelo PSO grandes são mais frequentes, contrastando com os valores em geral baixos para o número de grupos. Cabe explicar que não apenas o método de otimização, mas também a padronização das características e o tipo desses últimas alteram os resultados. Por isso, pela alteração do método de otimização, verifica-se valores maiores de σ. Já pela padronização dos dados, é possível observar a alteração do número c de grupos utilizando o mesmo método. Por fim, a constituição de uma base por grande número de características categóricas como na base **ttt** fez com que o valor apurado para σ seja muito grande.

Tabela 4.5: Valores de referência [23], médio μ, desvio padrão e mediano Md de σ e de c encontrados pelo Algoritmo Genético com dados originais e normalizados. Valores de γ ajustados por procedimento posterior de teste de generalização.

	Referência		Originais				Normalizados			
Base	σ LS-SVM	σ SVM	σ	c_μ	c_{Md}	γ	σ	c_μ	c_{Md}	γ
acr	22,75	12,43	54.97 (12.97)	46	49	0,5	50.57 (5.14)	24	21	0,5
bld	41,25	9	49.03 (0.97)	4	2	5	9.28 (1.58)	5	6	5
gcr	31,25	55	41.86 (2.36)	4	2	0,5	48.98 (0.65)	11	11	0,5
hea	5,69	7,15	49.65 (1.61)	4	5	0,1	44.55 (8.28)	11	7	0,5
ion	3,3	3,3	19.80 (12.87)	2	2	1	36.88 (7.91)	2	2	1
pid	240	15,5	48.92 (0.92)	7	7	0,5	45.04 (17.45)	3	2	0,01
snr	33	5,09	32.33 (16.27)	5	2	5	49.07 (0.88)	35	40	10
ttt	2,93	9	49.36 (0.64)	39	38	0,01	49.91 (1.67)	41	40	50
wbc	6,97	19,5	10.56 (0.62)	2	2	0,01	3.99 (0.87)	2	2	0,01
img	-	-	36.13 (26.93)	48	48	-	2.72 (0.72)	4	4	-

Tabela 4.6: Valores de referência [23], médio μ, desvio padrão e mediano Md de σ e de c encontrados pela Otimização por Enxame de Partículas com dados originais e normalizados. Valores de γ ajustados por procedimento posterior de teste de generalização.

	Referência		Originais				Normalizados			
Base	σ LS-SVM	σ SVM	σ	c_μ	c_{Md}	γ	σ	c_μ	c_{Md}	γ
acr	22,75	12,43	391.96 (412.71)	25	28	5	841.07 (347.00)	2	2	10
bld	41,25	9	73.57 (10.90)	2	2	50	17.00 (17.22)	6	6	5
gcr	31,25	55	40.79 (0.88)	2	2	0,5	3677.20 (879.91)	6	2	50
hea	5,69	7,15	87.98 (11.95)	2	2	1	213.75 (141.12)	4	2	1
ion	3,3	3,3	8.97 (2.17)	2	2	1	40.76 (28.55)	2	2	5
pid	240	15,5	120.90 (6.88)	2	2	5	96.91 (62.09)	5	2	0,01
snr	33	5,09	469.11 (643.96)	3	2	1	12 (328.15)	2	2	1
ttt	2,93	9	17129.00 (5991.20)	22	3	1000	896.56 (171.89)	5	4	1000
wbc	6,97	19,5	9.91 (0.07)	2	2	5	3.43 (0.03)	2	2	0,01
img	-	-	11.68 (12.73)	389	377	-	24.74 (71.47)	3	3	-

Algumas observações podem ser feitas pela análise das Tabelas 4.5 e 4.6 , com a apresentação de todos os valores dos parâmetros ajustados. Ficam evidentes as diferenças entre os resultados de σ obtidos por AG e por PSO. Percebe-se não apenas a discrepância entre estes valores encontrados como também o fato de os resultados do PSO estarem significativamente distantes da região definida para os valores mínimo e máximo assumidos dos indivíduos da população inicial dos algoritmos de otimização. Sendo esse um fato corrente nos resultados do PSO, pode-se inferir que os operadores de evolução do AG não permitiram aos indivíduos subsequentes sair da região dos indivíduos iniciais. Outro ponto a destacar é que os valores de γ variaram muito com a padronização dos dados em dez dos dezoito casos. Nos casos restantes, os valores não se alteraram em cada um dos pares. A maioria dos valores para esse parâmetro é de pequena magnitude, principalmente se comparados aos valores respectivos de σ. Percebe-se ainda valores grandes para esse parâmetro nas bases constituídas por muitas características categóricas, como por exemplo, a base **ttt**. O cálculo da similaridade entre as amostras nesses casos em que os atributos são binários faz com que o valor ajustado para o raio da gaussiana seja muito alto.

Dos resultados apresentados no início desta seção fica evidente que a alteração da metodologia para a determinação de σ resulta em valores diferentes da referência [23] para o parâmetro em quase todos os casos.

4.2.1 Análise dos resultados para cada uma das bases

Base acr

Segundo os resultados de acurácia na Tabela 4.7, o desempenho para os dados de treinamento foi menor que o desejado. No entanto, os classificador cumpriu satisfatoriamente o seu objetivo com os dados de validação e de teste haja visto que os resultados são superiores a 80% para esses grupos e são um pouco maiores ou iguais aos alcançados com o parâmetro σ do trabalho de referência.

Em relação à visualização dos resultados nas matrizes das Figuras 4.14 e 4.15, a presença de conjuntos grandes de amostras similares só aparece na matriz inferior esquerda da Figura 4.14, o que confirma os resultados para o número de grupos apresentados na Tabela 4.5. A Figura 4.16 apresenta pelo menos um conjunto grande de amostras semelhantes em duas de suas matrizes ordenadas da esquerda. Porém, a Figura 4.17 não confirma o resultado, expondo matrizes parecidas às dos resultados com AG.

Tabela 4.7: Comparativo de desempenho por acc dos classificadores LS-SVM para base *acr*: [23] *versus* Presente Trabalho. Valores médio e (desvio padrão) por grupo de amostras.

	Treinamento	Validação	Teste
AG e Orig.	0,91 (0,01) - 0,89 (0,01)	0,85 (0,02) - 0,86 (0,01)	0,86 (0,05) - 0,87 (0,05)
AG e Norm.	0,9 (0,01) - 0,89 (0,01)	0,87 (0,03) - 0,87 (0,02)	0,87 (0,04) - 0,87 (0,04)
PSO e Orig.	0,95 (0,01) - 0,88 (0,01)	0,85 (0,03) - 0,86 (0,02)	0,85 (0,04) - 0,87 (0,04)
PSO e Norm.	0,95 (0,01) - 0,88 (0,01)	0,85 (0,03) - 0,86 (0,03)	0,85 (0,03) - 0,87 (0,04)

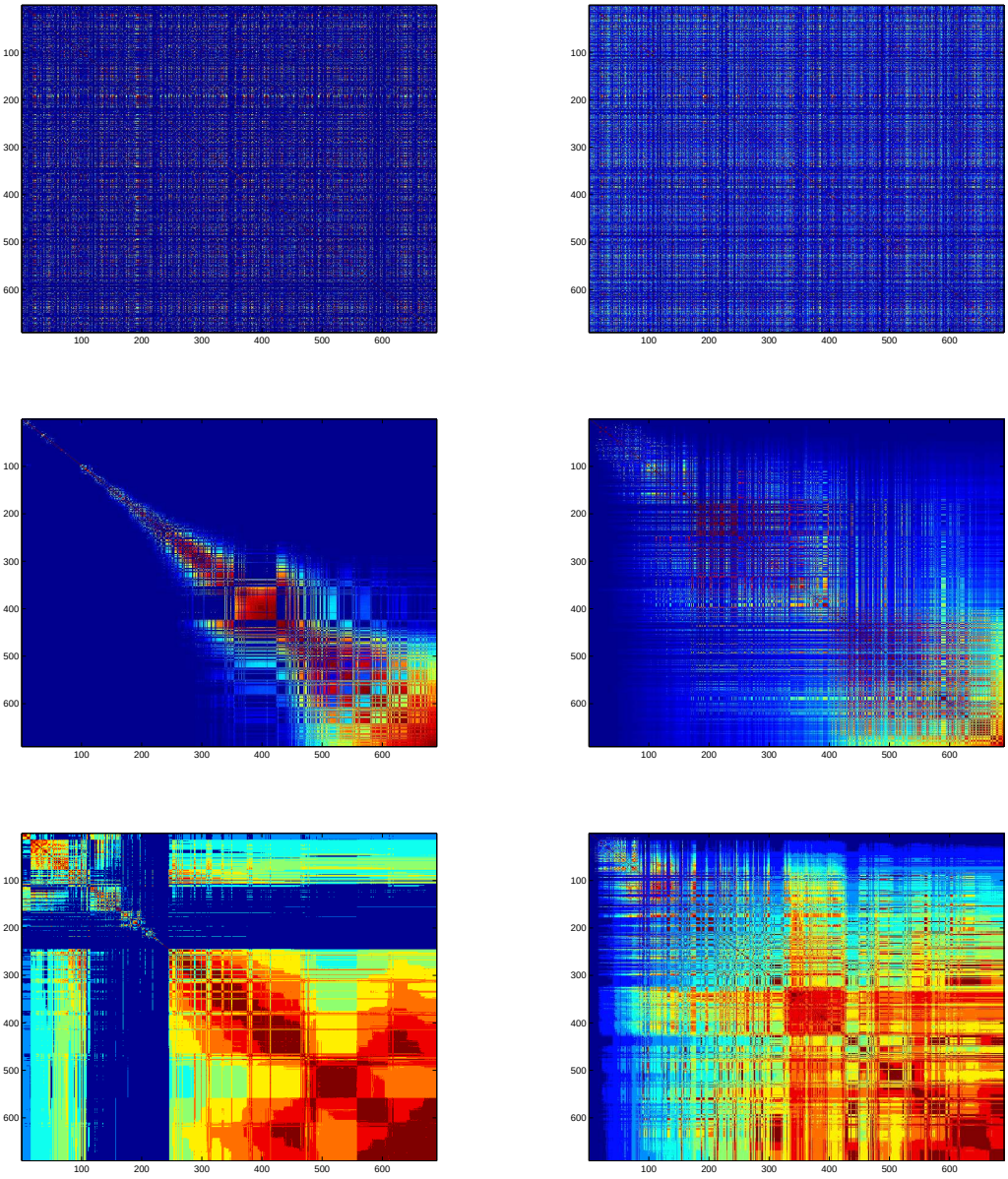

Figura 4.14: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *acr* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

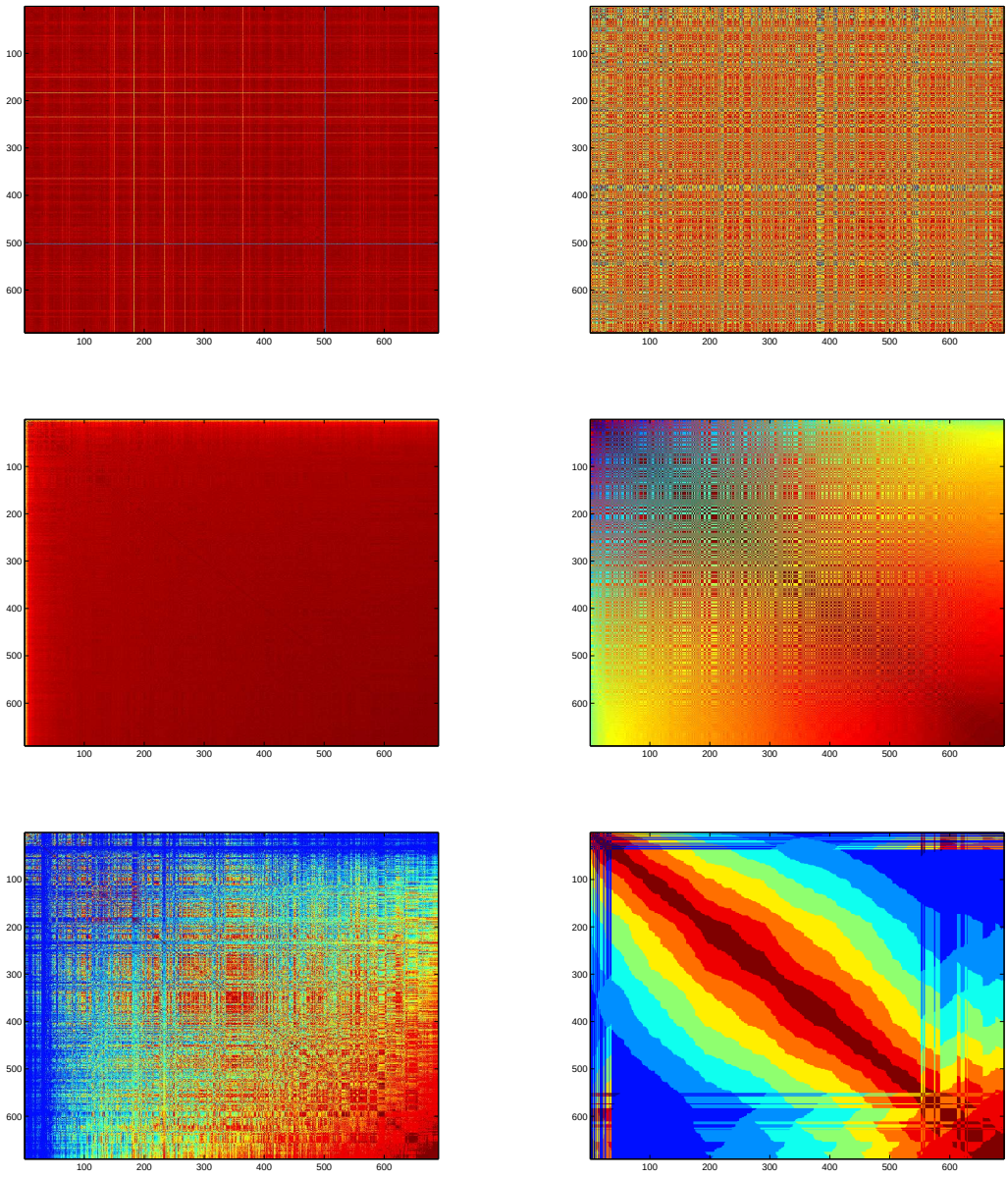

Figura 4.15: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *acr* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

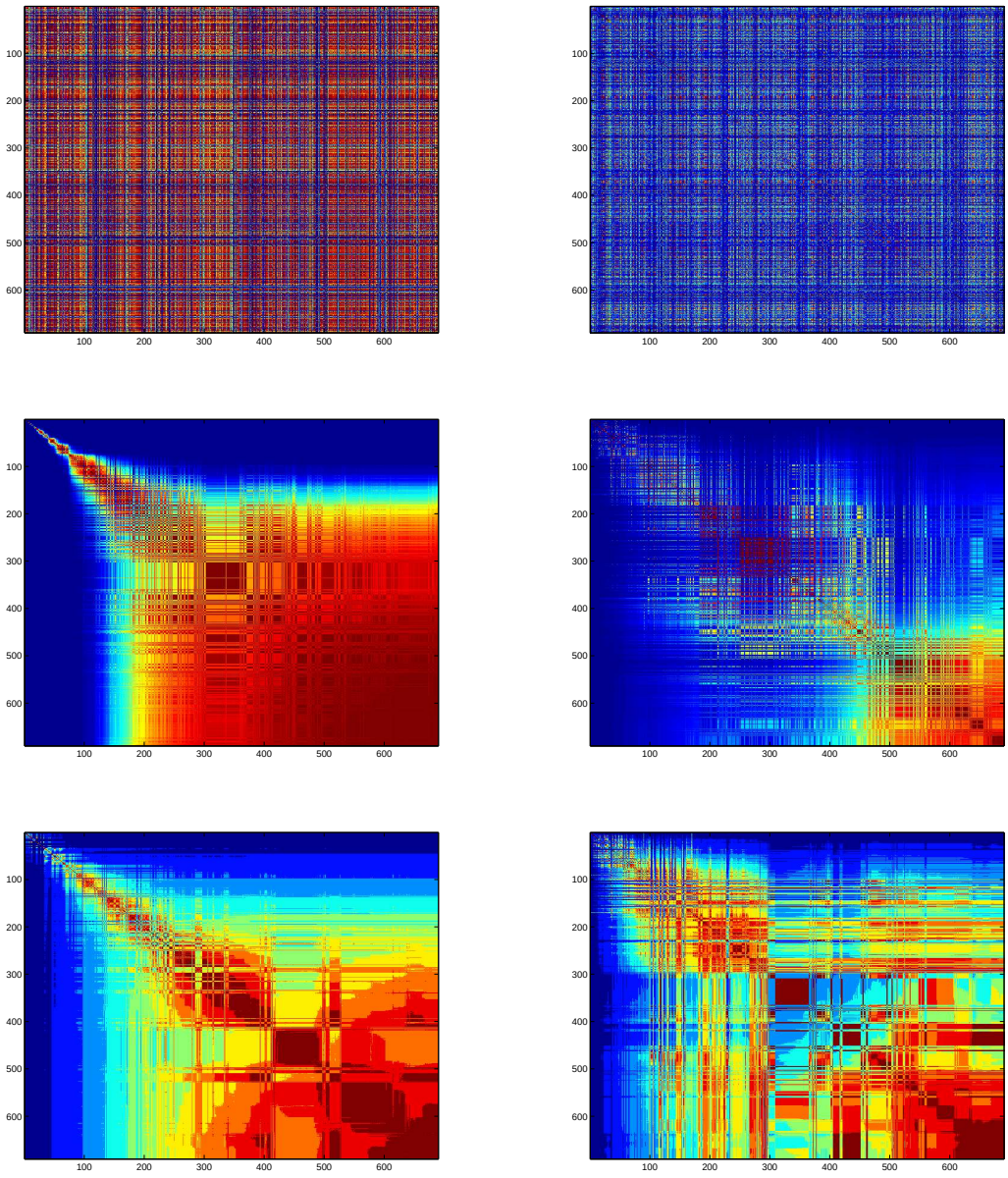

Figura 4.16: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *acr* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

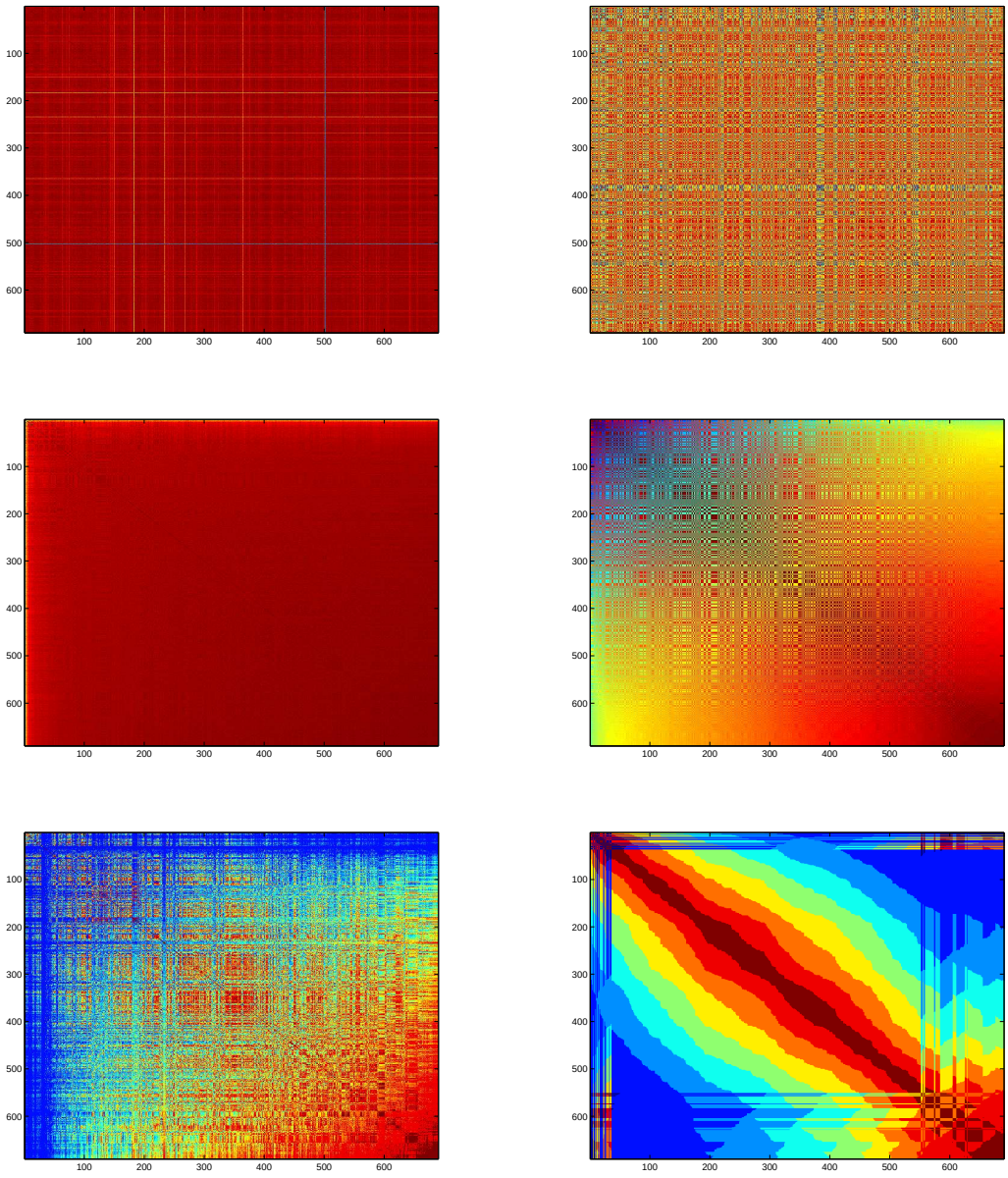

Figura 4.17: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *acr* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

Tabela 4.8: Comparativo de desempenho por acc dos classificadores LS-SVM para base *bld*: [23] *versus* Presente Trabalho. Valores médio e (desvio padrão) por grupo de amostras.

	Treinamento	Validação	Teste
AG e Orig.	0,78 (0,02) - 0,79 (0,02)	0,7 (0,05) - 0,7 (0,04)	0,71 (0,06) - 0,7 (0,06)
AG e Norm.	0,86 (0,03) - 0,79 (0,03)	0,68 (0,03) - 0,7 (0,04)	0,7 (0,05) - 0,71 (0,04)
PSO e Orig.	0,79 (0,01) - 0,82 (0,02)	0,68 (0,05) - 0,68 (0,04)	0,71 (0,04) - 0,7 (0,04)
PSO e Norm.	0,82 (0,02) - 0,78 (0,03)	0,69 (0,04) - 0,7 (0,04)	0,69 (0,03) - 0,7 (0,02)

Tabela 4.9: Comparativo de desempenho por acc dos classificadores LS-SVM para base *gcr*: [23] *versus* Presente Trabalho. Valores médio e (desvio padrão) por grupo de amostras.

	Treinamento	Validação	Teste
AG e Orig.	0,83 (0,01) - 0,85 (0,01)	0,76 (0,02) - 0,76 (0,02)	0,78 (0,06) - 0,78 (0,06)
AG e Norm.	0,83 (0,01) - 0,86 (0,01)	0,75 (0,02) - 0,75 (0,02)	0,76 (0,05) - 0,76 (0,04)
PSO e Orig.	0,82 (0,01) - 0,84 (0,01)	0,77 (0,02) - 0,77 (0,02)	0,78 (0,05) - 0,78 (0,05)
PSO e Norm.	0,79 (0,01) - 1 (0)	0,76 (0,02) - 0,7 (0,01)	0,76 (0,05) - 0,7 (0,06)

Base bld

A Tabela 4.8 mostra que o desempenho dos classificadores foi tão bom quanto os da referência. Isso porque as acurácias para os dados de validação e de teste ficaram muito próximas às desejadas, embora, para os dados de treinamento, o mesmo não tenha ocorrido. Por sua vez, as Figuras 4.18 e 4.19 destacam pelo menos um grupo de amostras nas matrizes ordenadas. Nas Figuras 4.20 e 4.21, a existência de um grupo de amostras pode ser observada, em acordo com os resultados do AG. Cabe destacar certa similaridade das ordenações obtidas pelos dois métodos nas matrizes das quatro figuras em questão.

Base gcr

Os classificadores projetados para esta base alcançaram desempenho satisfatório para os três grupos de dados, principalmente o de treinamento nos quatro casos segundo a Tabela 4.9. Nas Figuras 4.22 e 4.24, resultantes dos testes com dados sem padronização, a existência de dois grupos de amostras fica evidenciado nas matrizes ordenadas. Percebe-se um grupo maior de amostras na submatriz diagonal inferior e um menor na submatriz diagonal superior. O mesmo resultado não é tão destacado nas Figuras 4.23 e 4.25, exceto nas matrizes de *kernel* ordenadas pelo método Minus.

Base hea

Para a base em questão, o desempenho dos classificadores é superior à referência apenas para os dados de treinamento. Conforme apresenta a Tabela 4.10, o desempenho para os dados de validação e de teste é bom, mas

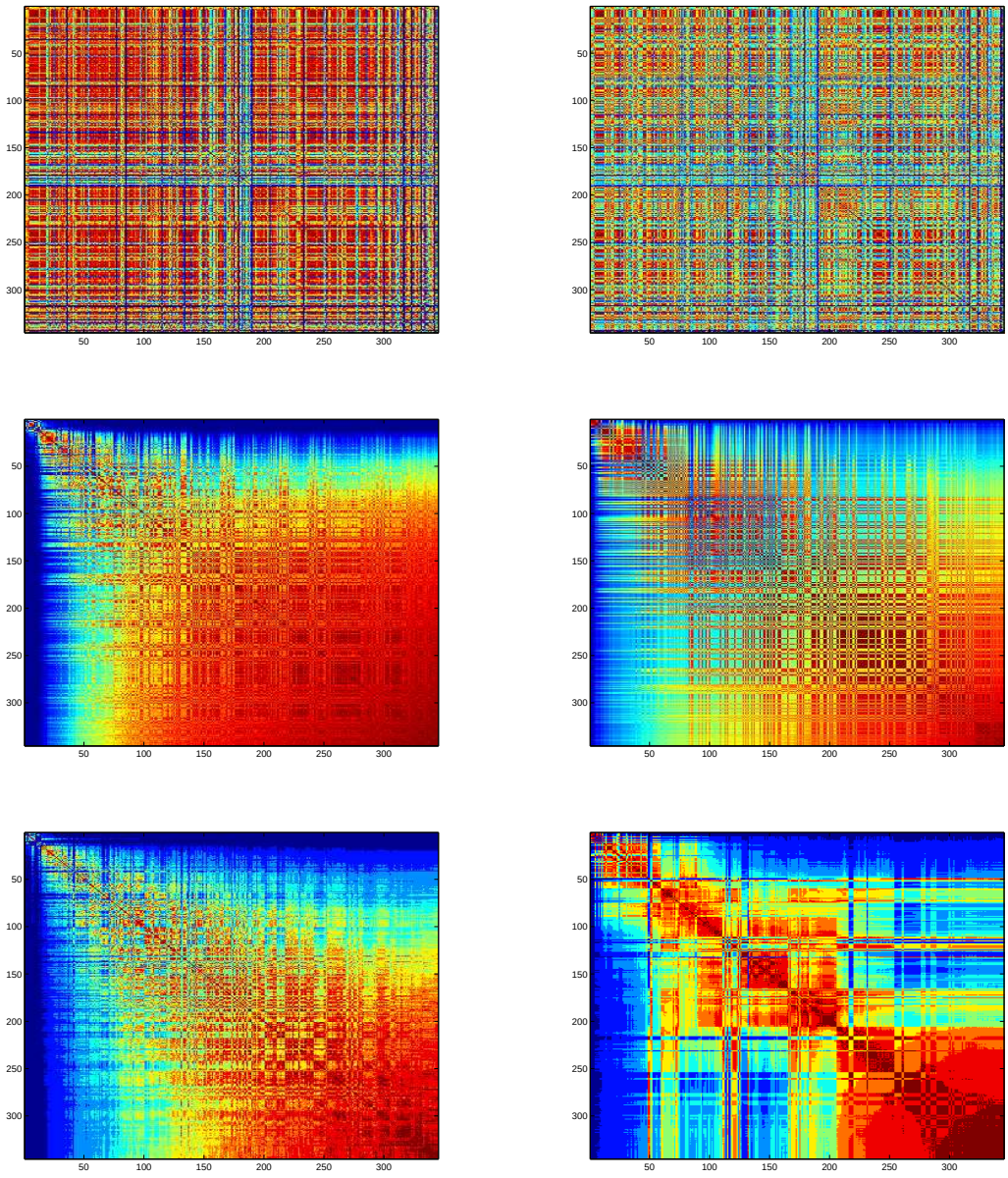

Figura 4.18: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *bld* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

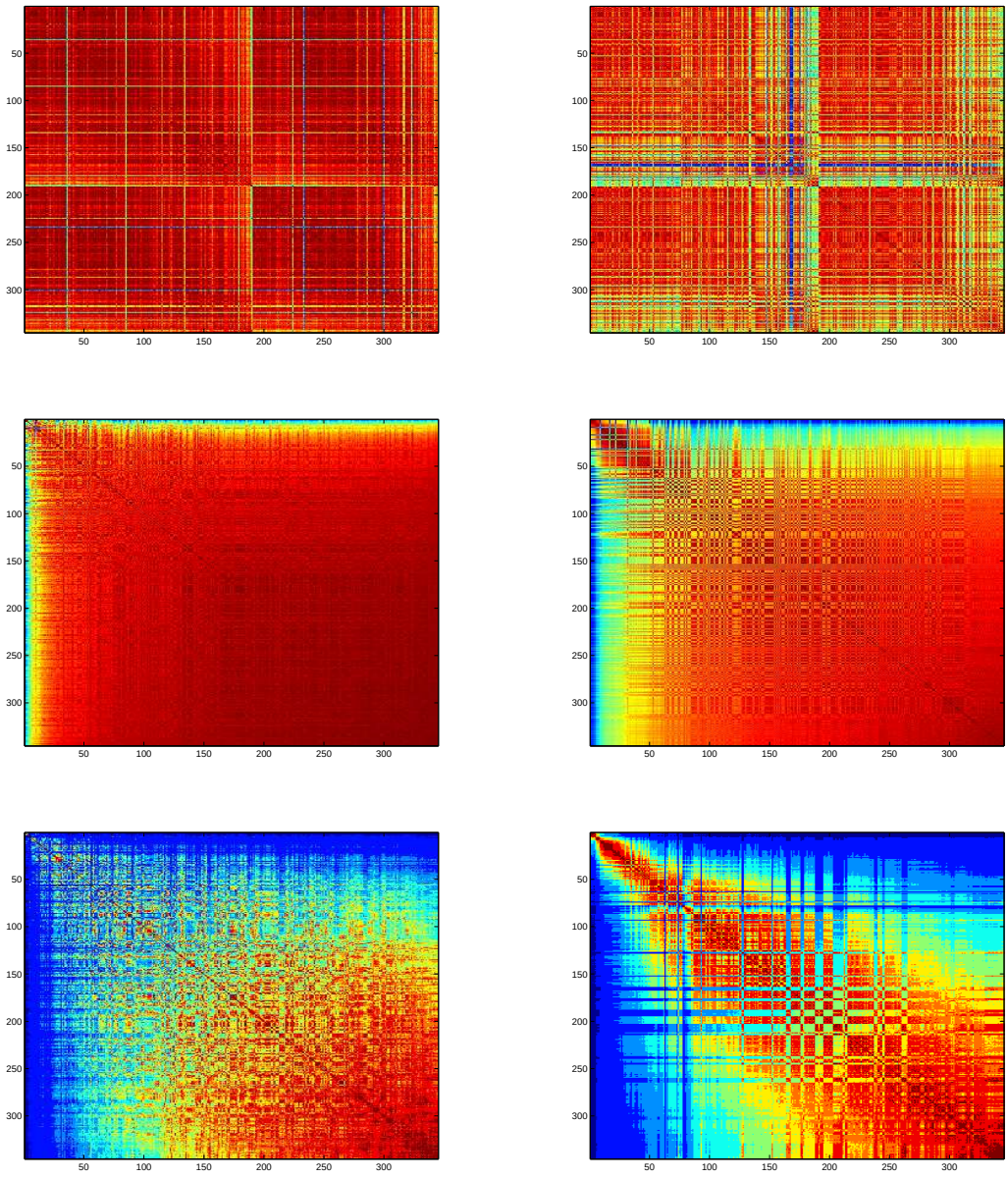

Figura 4.19: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *bld* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

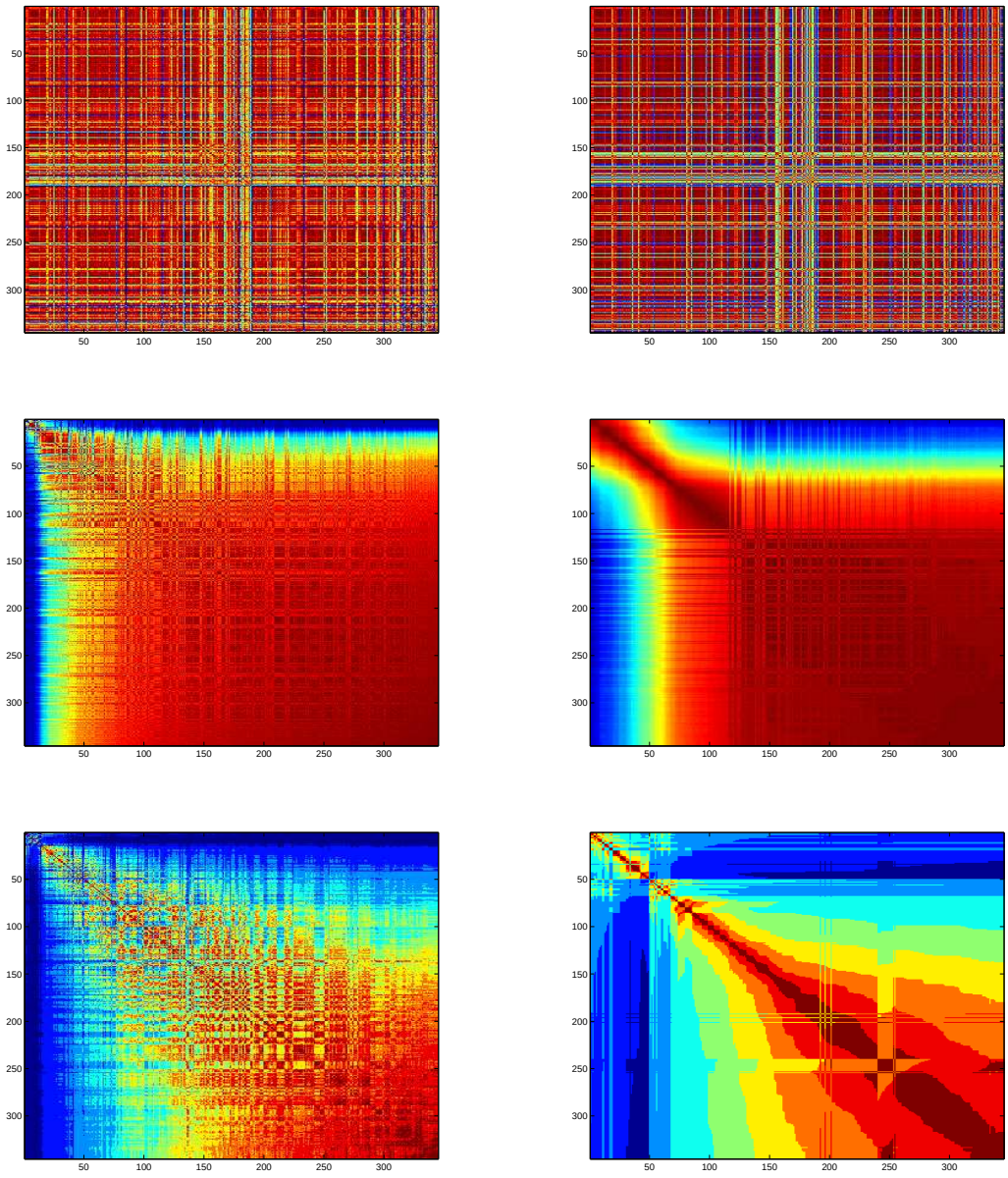

Figura 4.20: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *bld* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

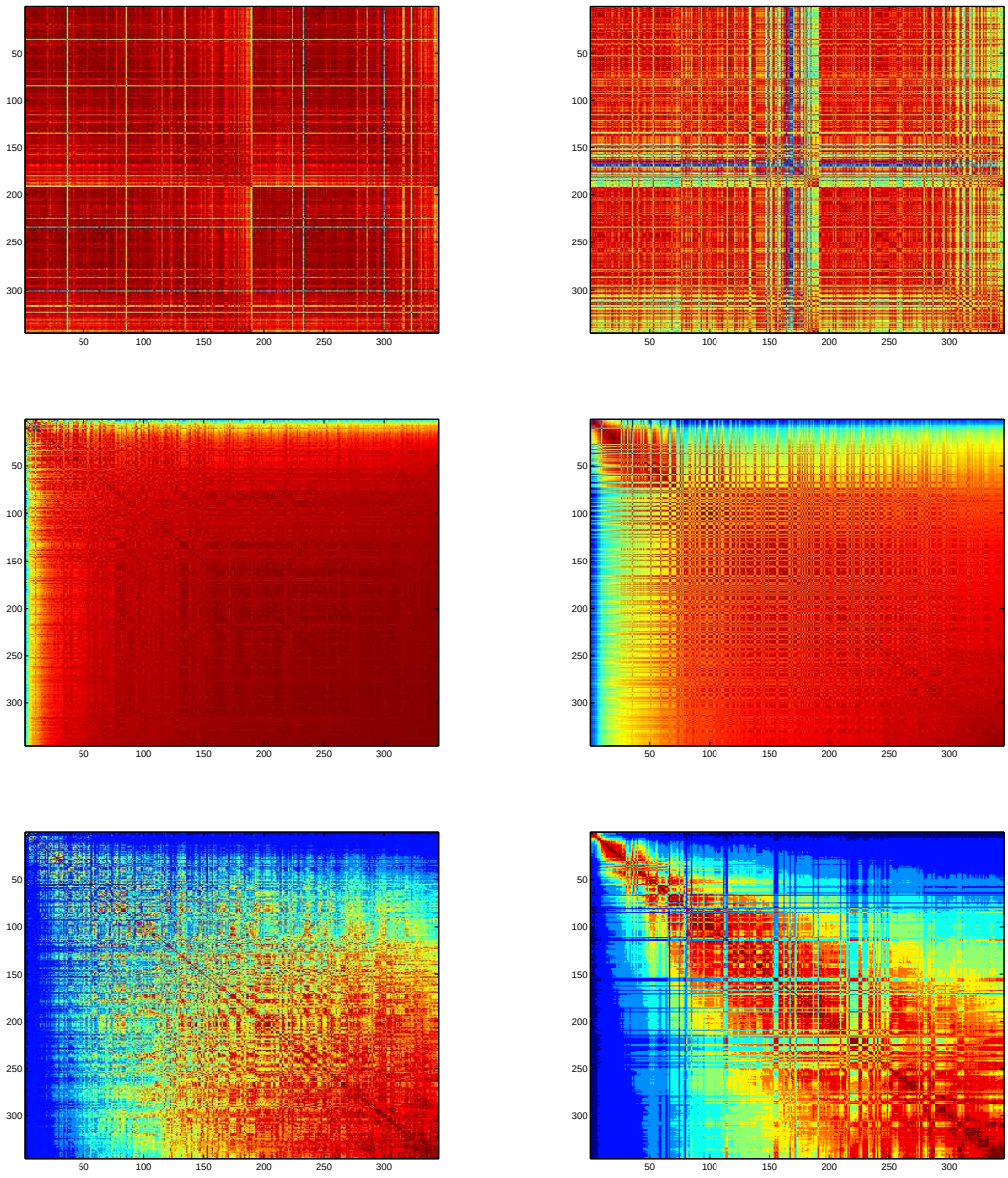

Figura 4.21: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *bld* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

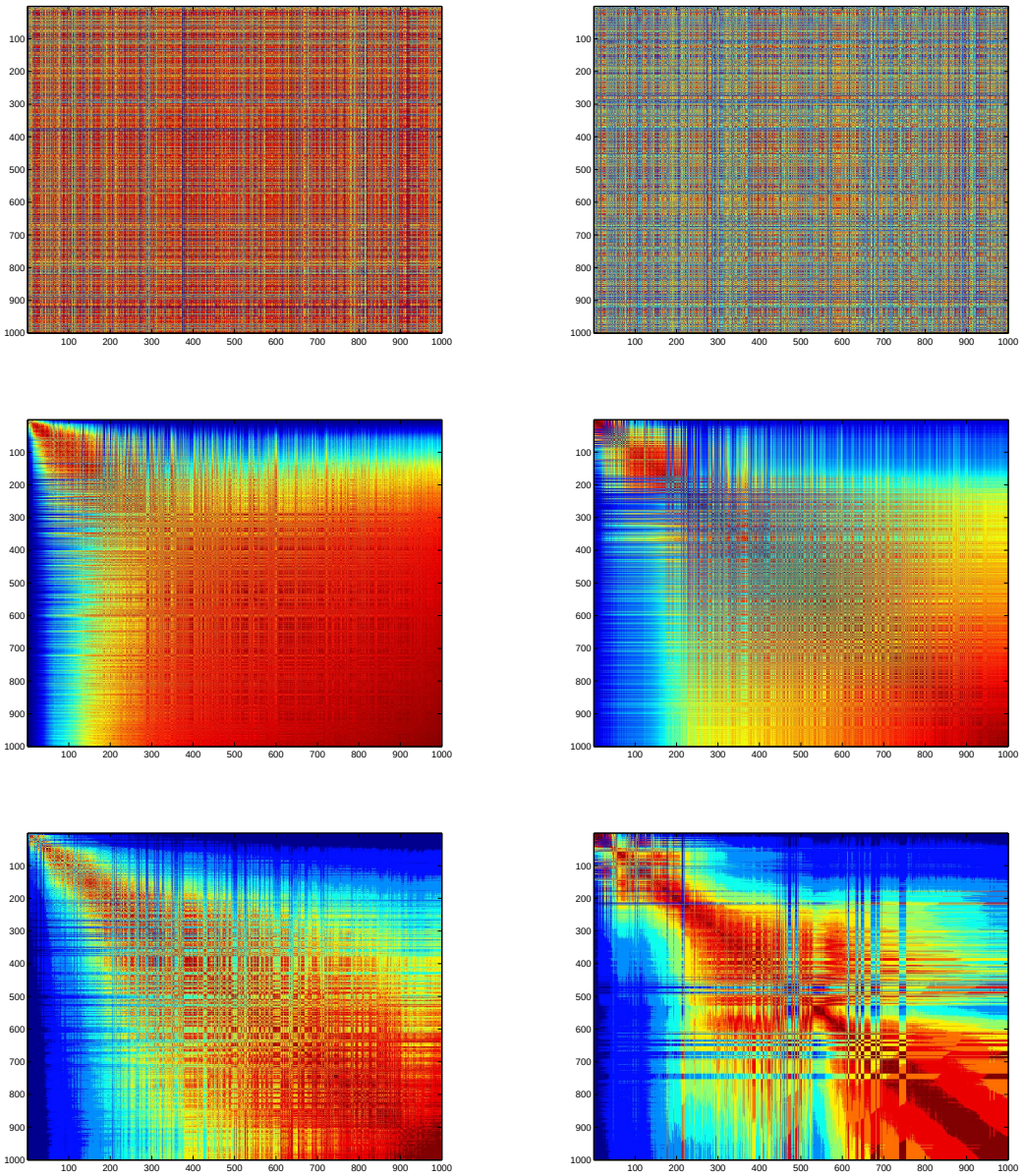

Figura 4.22: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *gcr* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

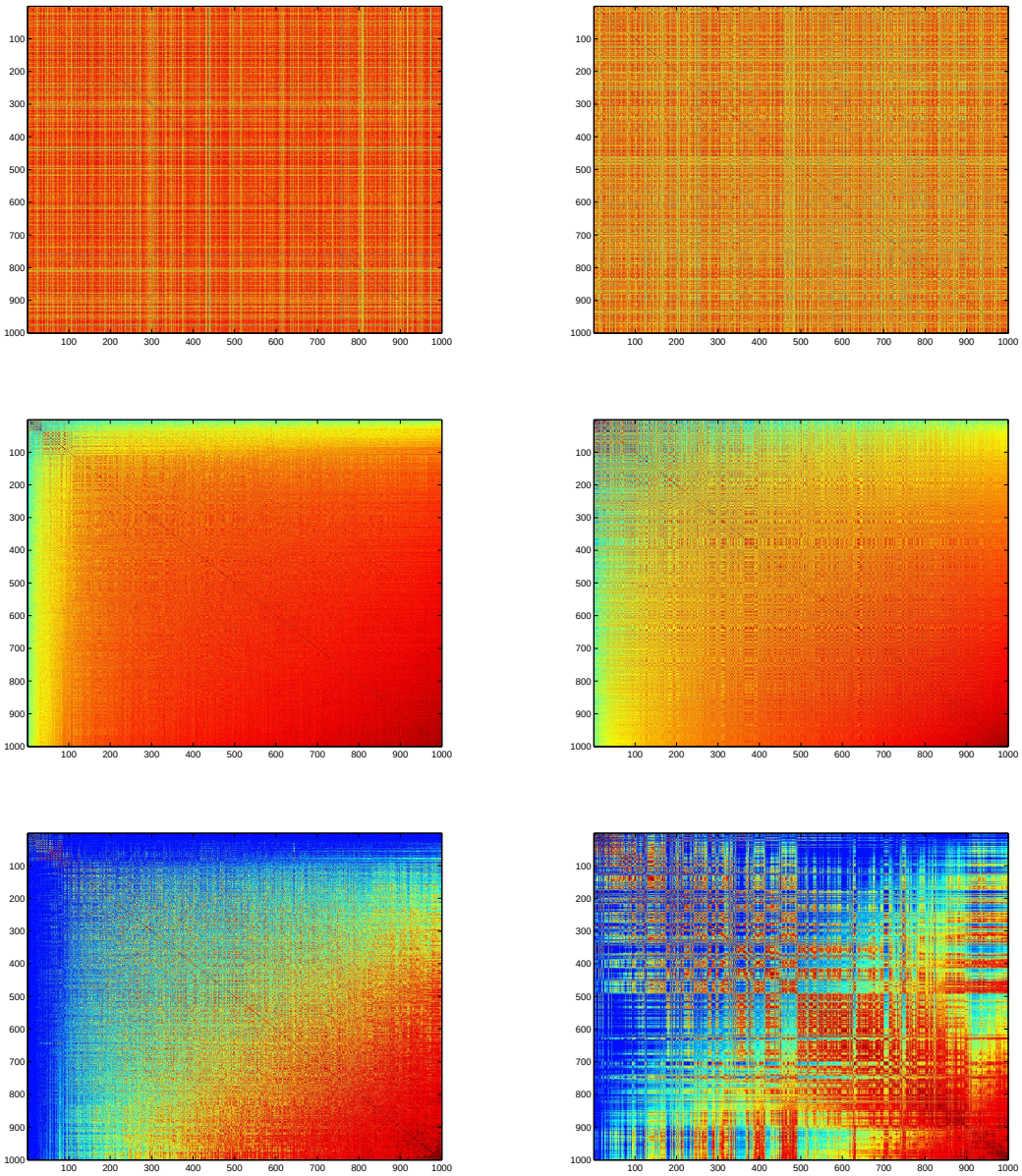

Figura 4.23: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *gcr* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

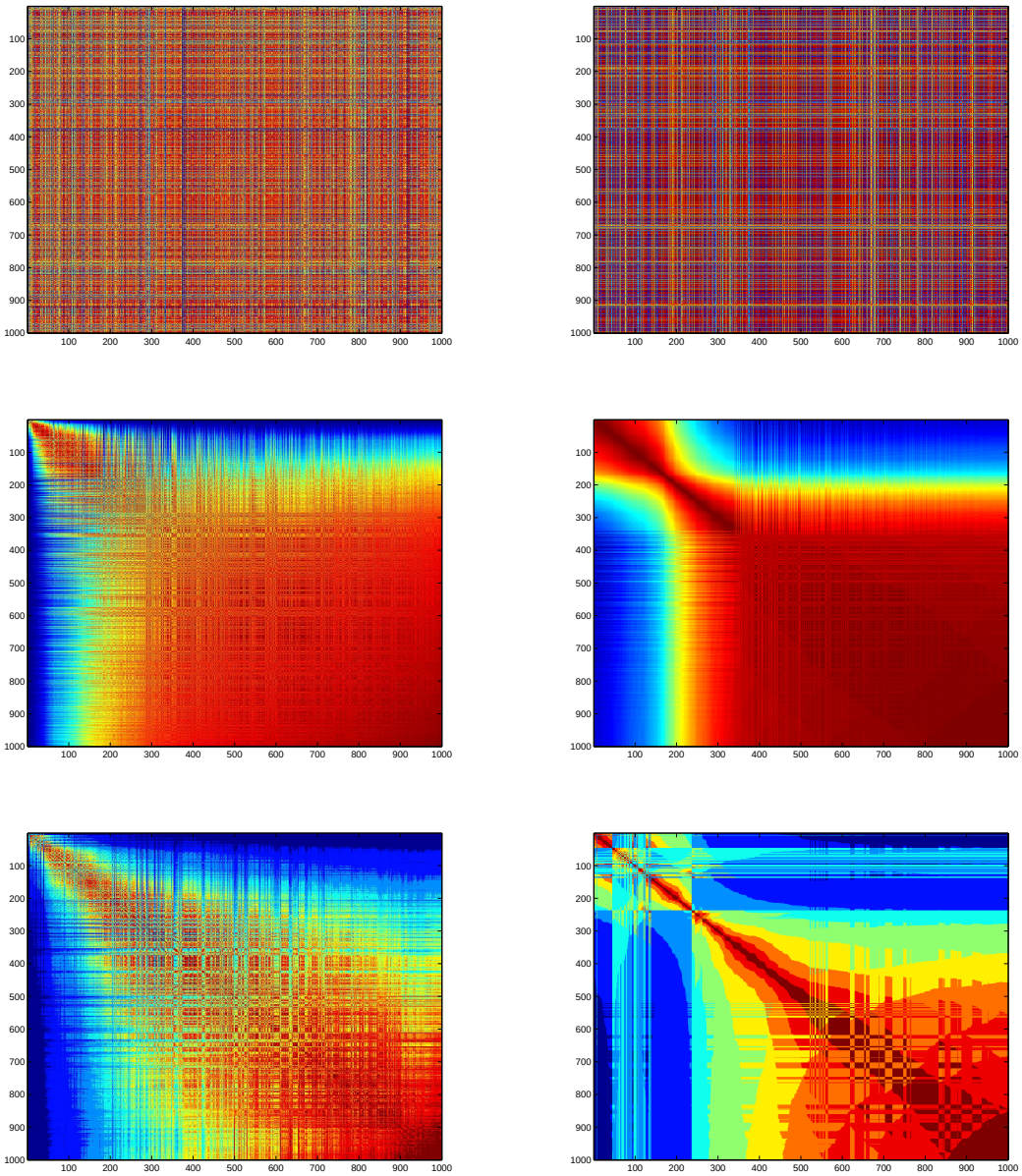

Figura 4.24: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *gcr* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

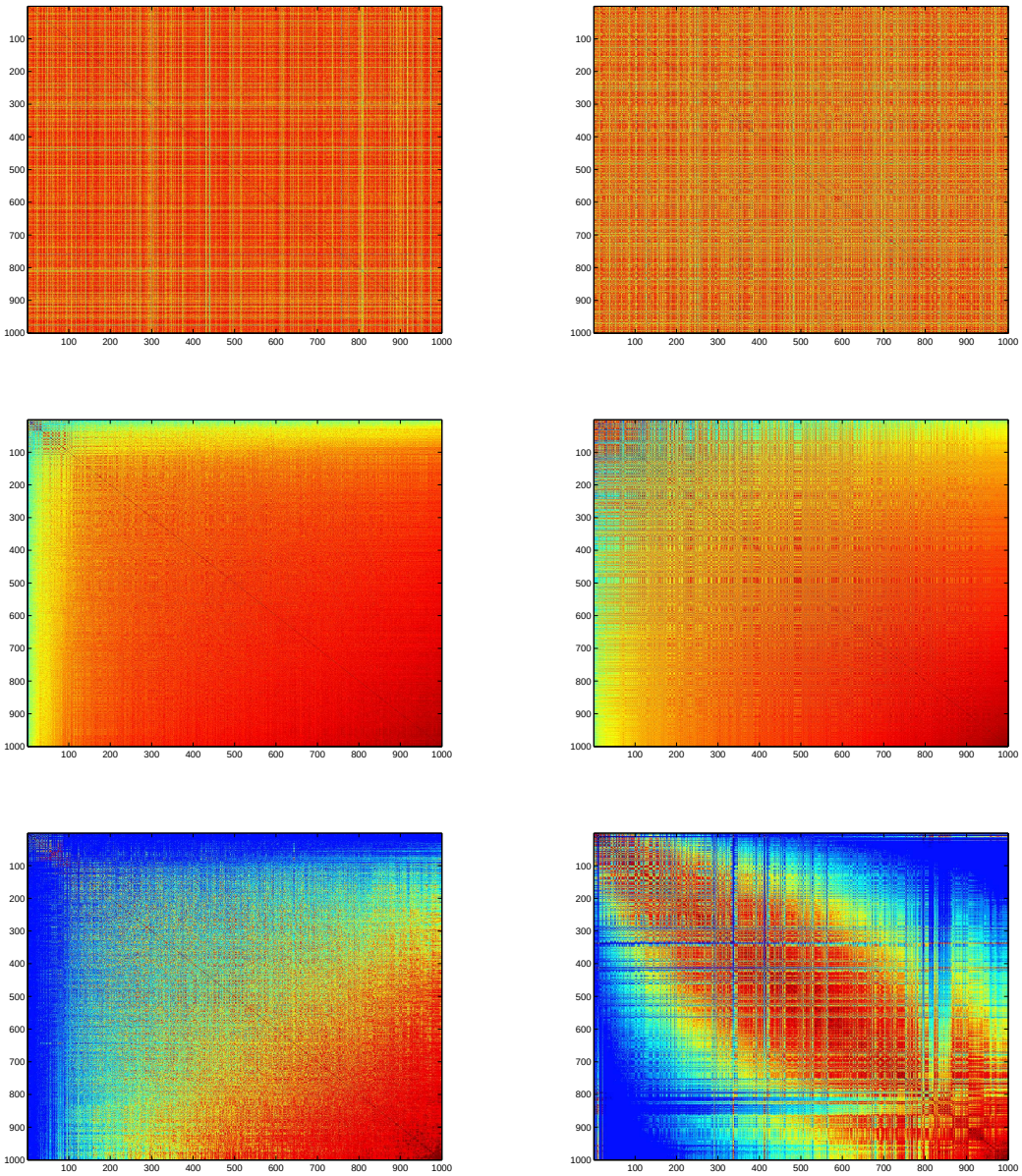

Figura 4.25: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *gcr* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

Tabela 4.10: Comparativo de desempenho por acc dos classificadores LS-SVM para base *hea*: [23] *versus* Presente Trabalho. Valores médio e (desvio padrão) por grupo de amostras.

	Treinamento	Validação	Teste
AG e Orig.	0,85 (0,03) - 0,89 (0,04)	0,81 (0,04) - 0,74 (0,06)	0,81 (0,03) - 0,73 (0,07)
AG e Norm.	0,88 (0,04) - 0,97 (0,02)	0,84 (0,04) - 0,79 (0,05)	0,82 (0,02) - 0,77 (0,04)
PSO e Orig.	0,88 (0,03) - 1 (0,01)	0,84 (0,03) - 0,8 (0,03)	0,83 (0,04) - 0,8 (0,03)
PSO e Norm.	0,87 (0,02) - 0,99 (0,01)	0,83 (0,03) - 0,79 (0,05)	0,85 (0,03) - 0,8 (0,04)

significativamente abaixo do *benchmark*.

Quanto aos resultados para os dados padronizados, merecem destaque as matrizes de proximidade ordenadas por Minus nas Figuras 4.27 e 4.29. Nos dois casos, há duas pequenas submatrizes bem delineadas no canto direito inferior. Outro ponto a ser ressaltado é a grande semelhança de todas as matrizes das referidas figuras ainda que os parâmetros encontrados sejam diferentes. As Figuras 4.26 e 4.28 não guardam semelhanças entre si e nem mesmo entre as suas matrizes, exceto a matriz de proximidade dessa segunda figura. Percebe-se que a ordenação pelos dois métodos obteve resultados que fornece informações complementares. Enquanto a ordenação por autovetor revela dois grupos bem destacados, a ordenação pelo Minus aponta a existência de dois subgrupos bem definidos na submatriz diagonal inferior do primeiro resultado.

Base ion

Percebe-se na Tabela 4.11 o alto desempenho dos classificadores ajustados, especialmente com relação aos dados de treinamento. Entretanto, os resultados obtidos com os parâmetros da referência são ligeiramente maiores para os dados de validação e de teste, que têm maior peso na avaliação final das máquinas de aprendizado.

Por apresentarem matrizes muito semelhantes, as Figuras 4.30, 4.31, 4.32 e 4.33 apontam a existência de três grupos de amostras da base em questão. Percebe-se pelo menos um grupo de amostras bem delineado nas matrizes ordenadas pelo autovetor. Mas é nas matrizes ordenadas utilizando o Minus que ficam destacadas submatrizes bem definidas sobre a diagonal principal, especialmente no canto direito inferior de cada matriz, onde há duas submatrizes em destaque. No canto esquerdo superior, outras duas submatrizes também podem ser notadas, mas a forte inter-relação entre suas amostras não permitem a desassociação destas últimas.

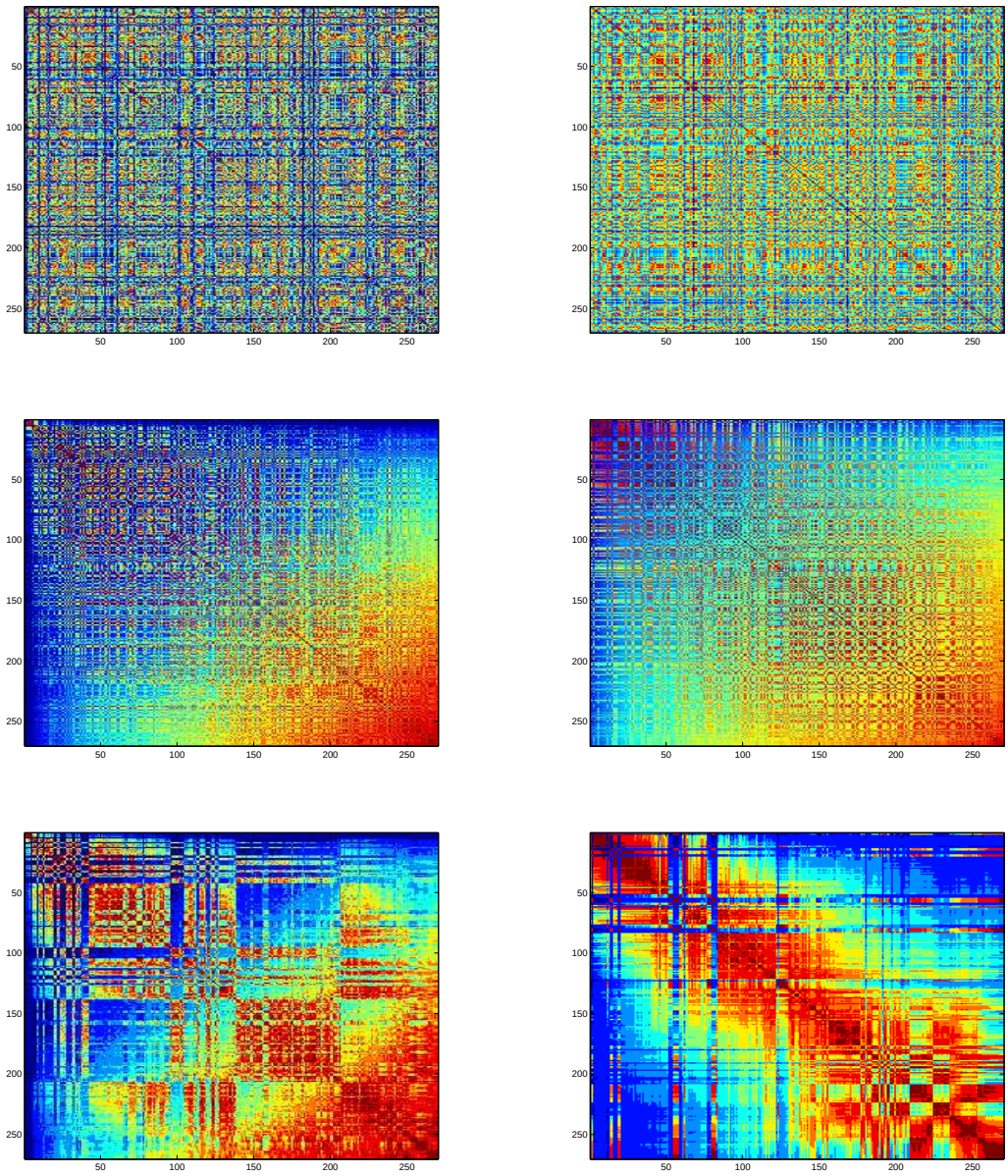

Figura 4.26: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *hea* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

Tabela 4.11: Comparativo de desempenho por acc dos classificadores LS-SVM para base *ion*: [23] *versus* Presente Trabalho. Valores médio e (desvio padrão) por grupo de amostras.

	Treinamento	Validação	Teste
AG e Orig.	0,99 (0,01) - 1 (0)	0,95 (0,01) - 0,93 (0,03)	0,94 (0,04) - 0,92 (0,04)
AG e Norm.	0,98 (0,01) - 1 (0)	0,94 (0,02) - 0,93 (0,02)	0,94 (0,02) - 0,92 (0,04)
PSO e Orig.	0,99 (0) - 1 (0)	0,94 (0,02) - 0,93 (0,02)	0,94 (0,02) - 0,93 (0,04)
PSO e Norm.	0,99 (0,01) 1 (0)	0,95 (0,02) - 0,9 (0,03)	0,95 (0,03) - 0,91 (0,03)

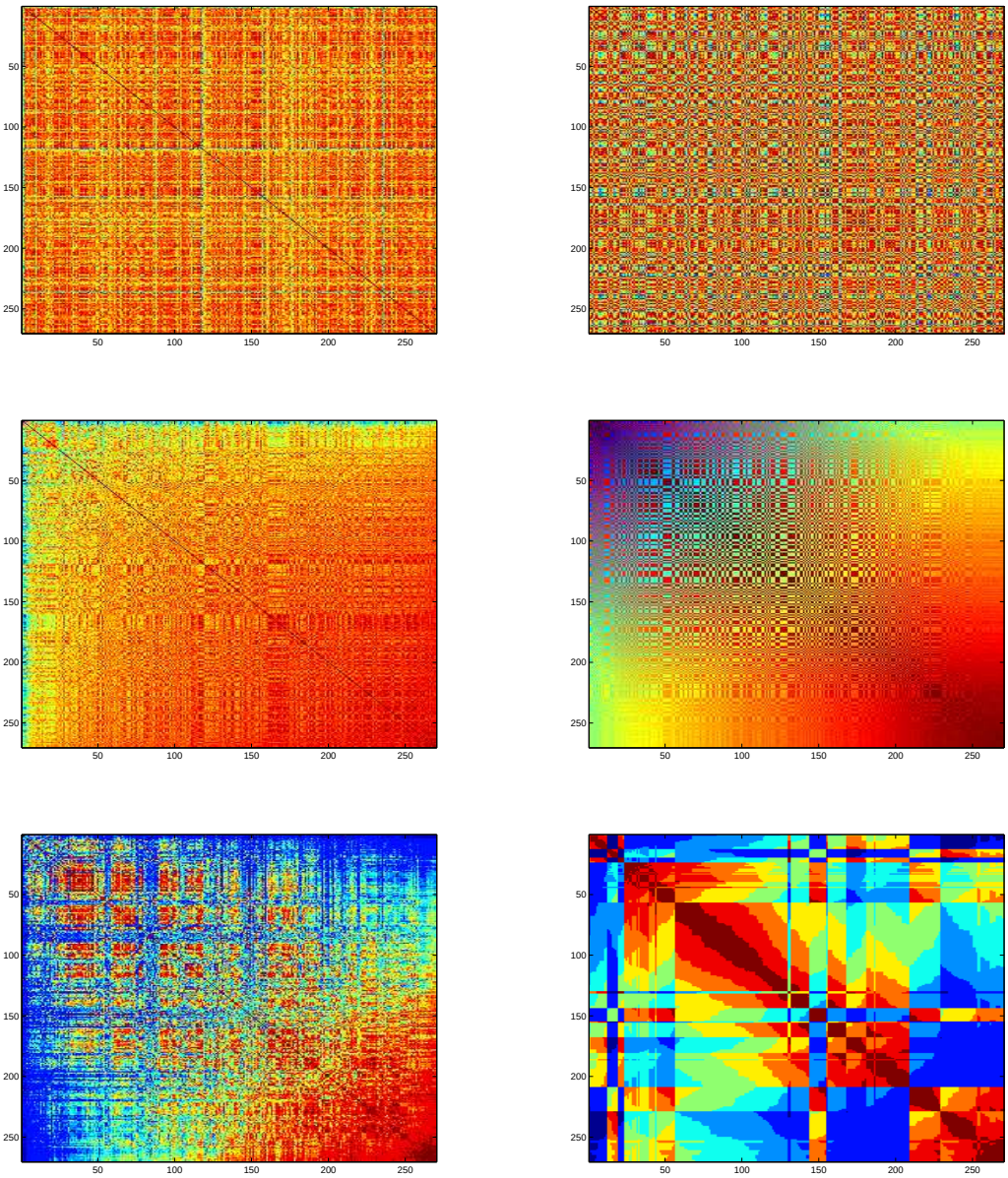

Figura 4.27: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *hea* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

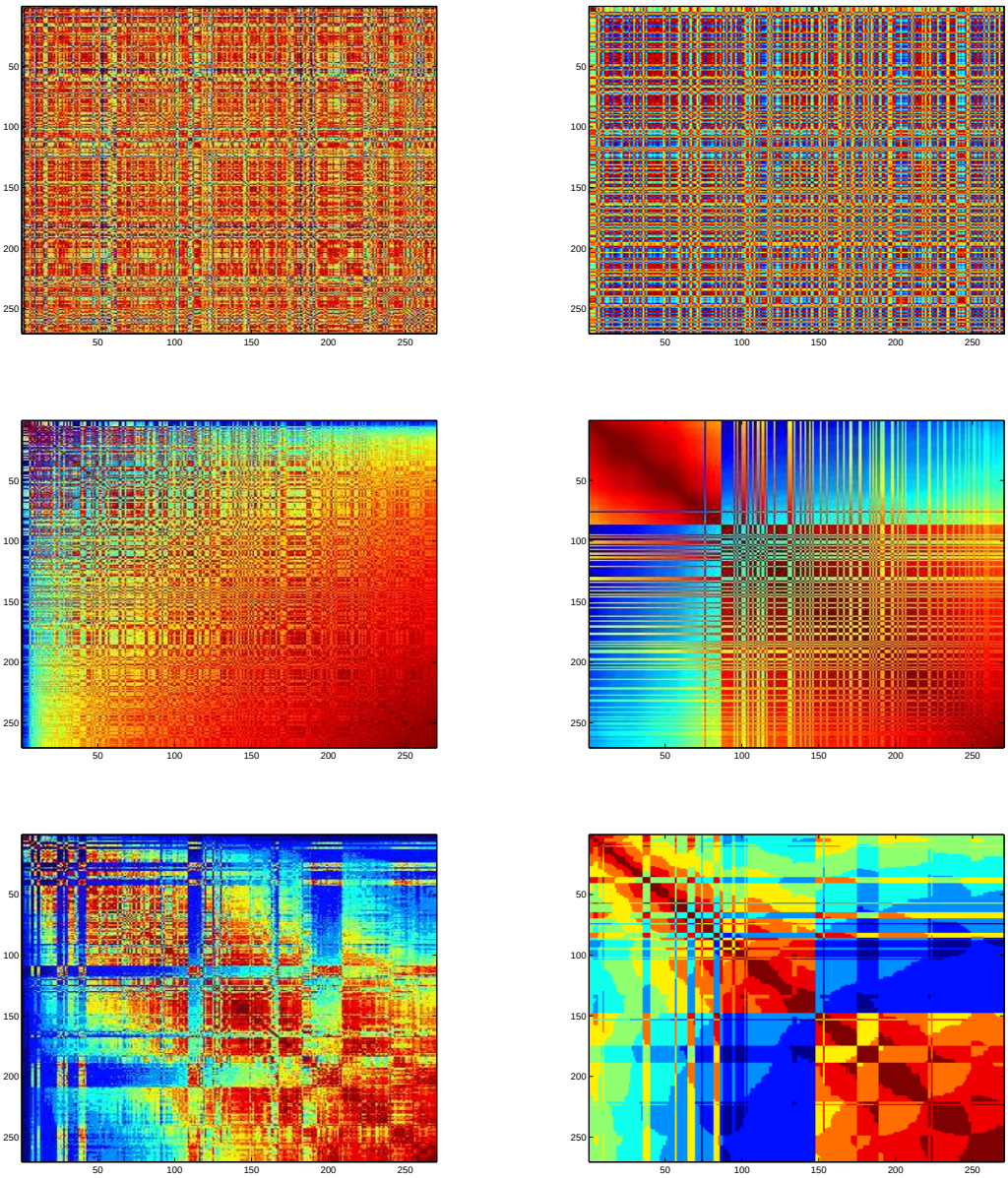

Figura 4.28: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *hea* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

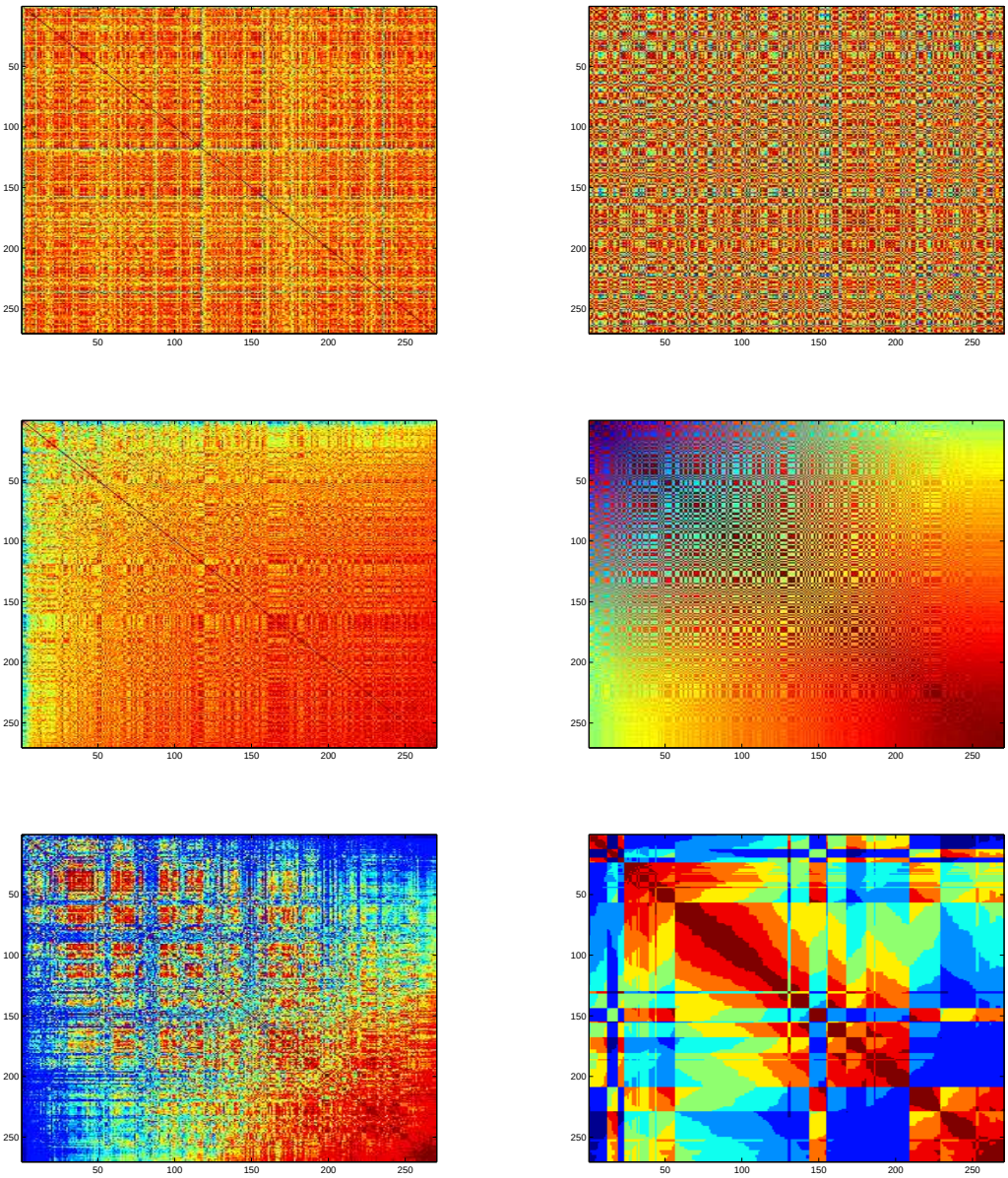

Figura 4.29: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *hea* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

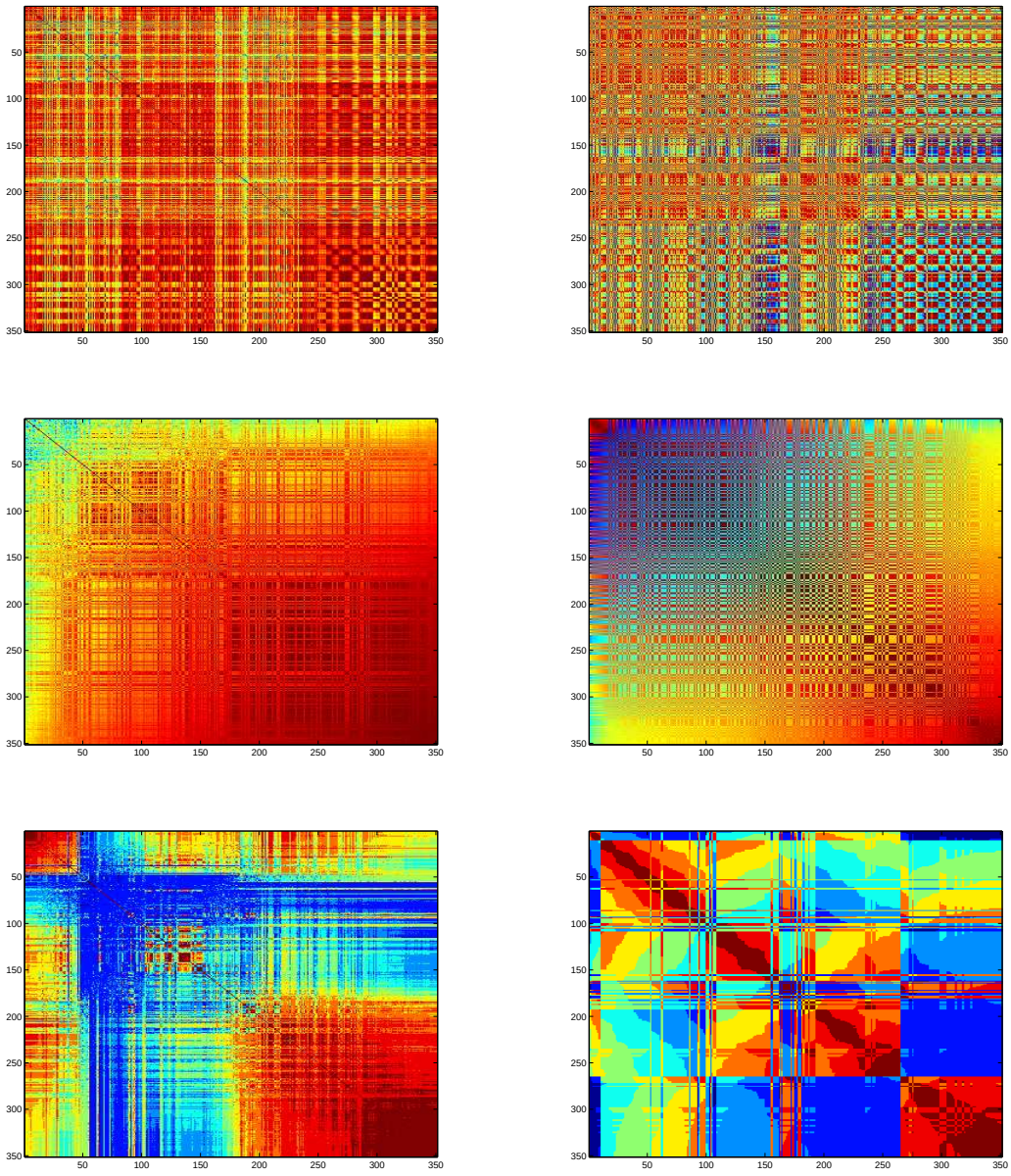

Figura 4.30: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *ion* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

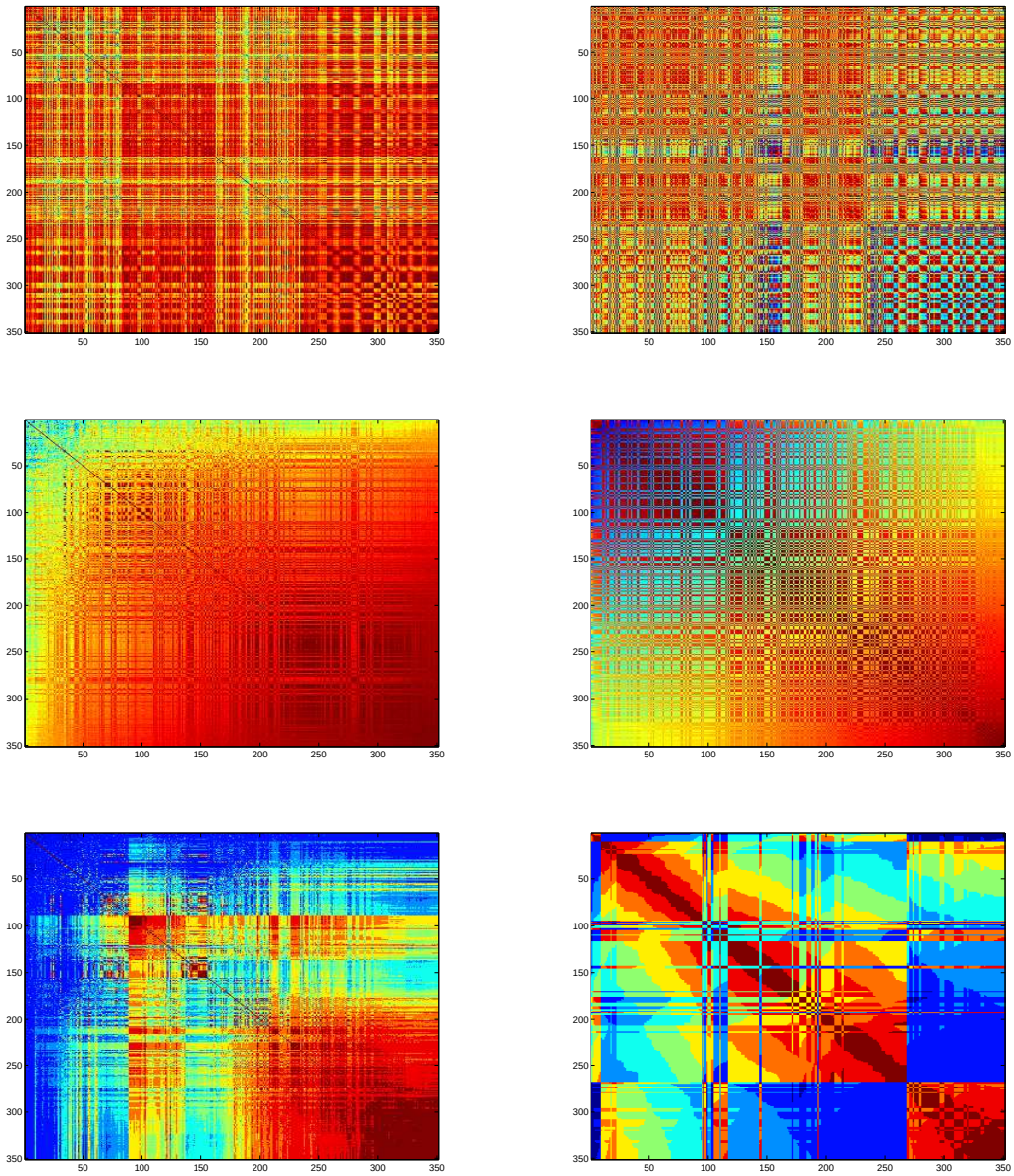

Figura 4.31: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *ion* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

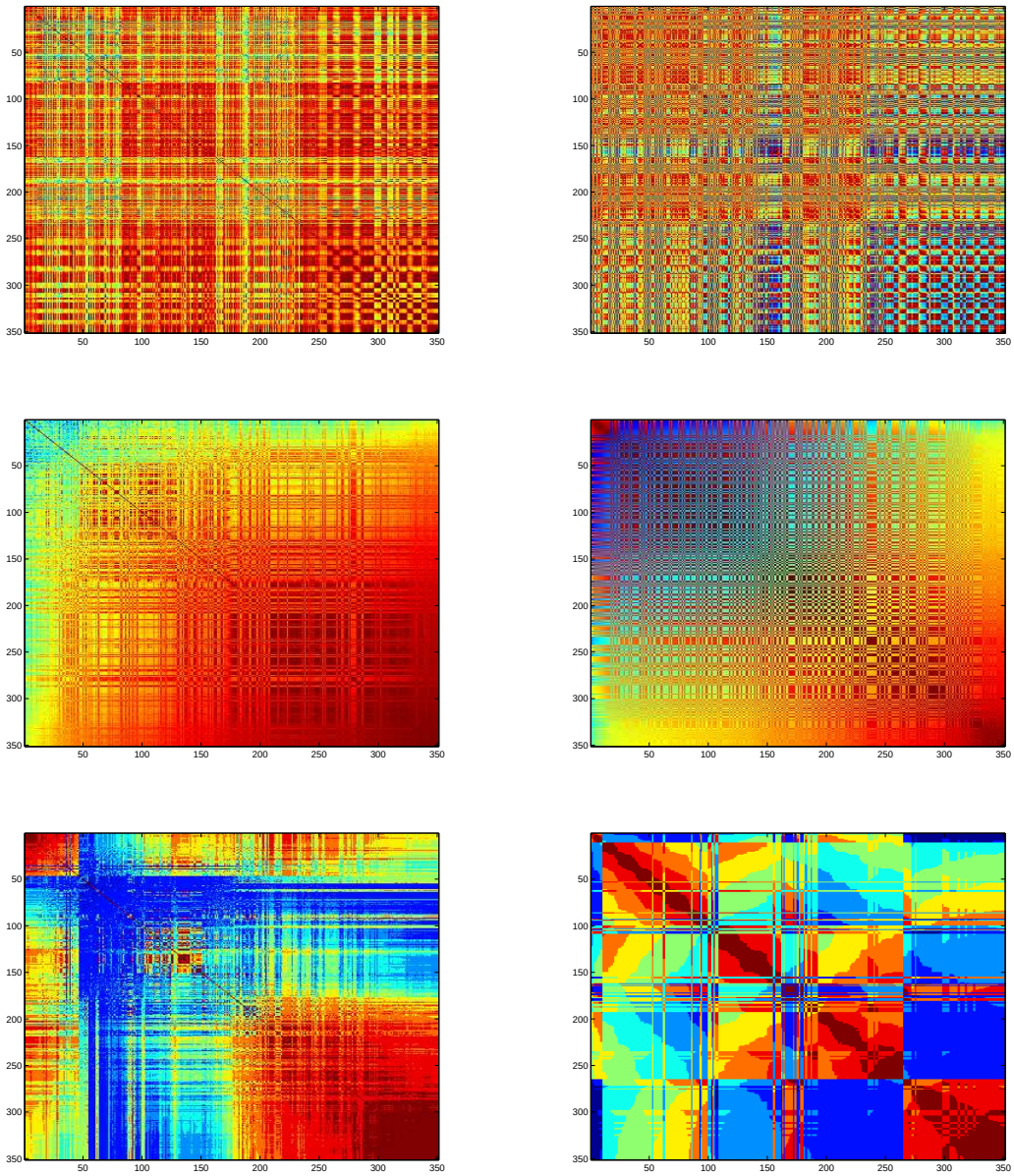

Figura 4.32: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *ion* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

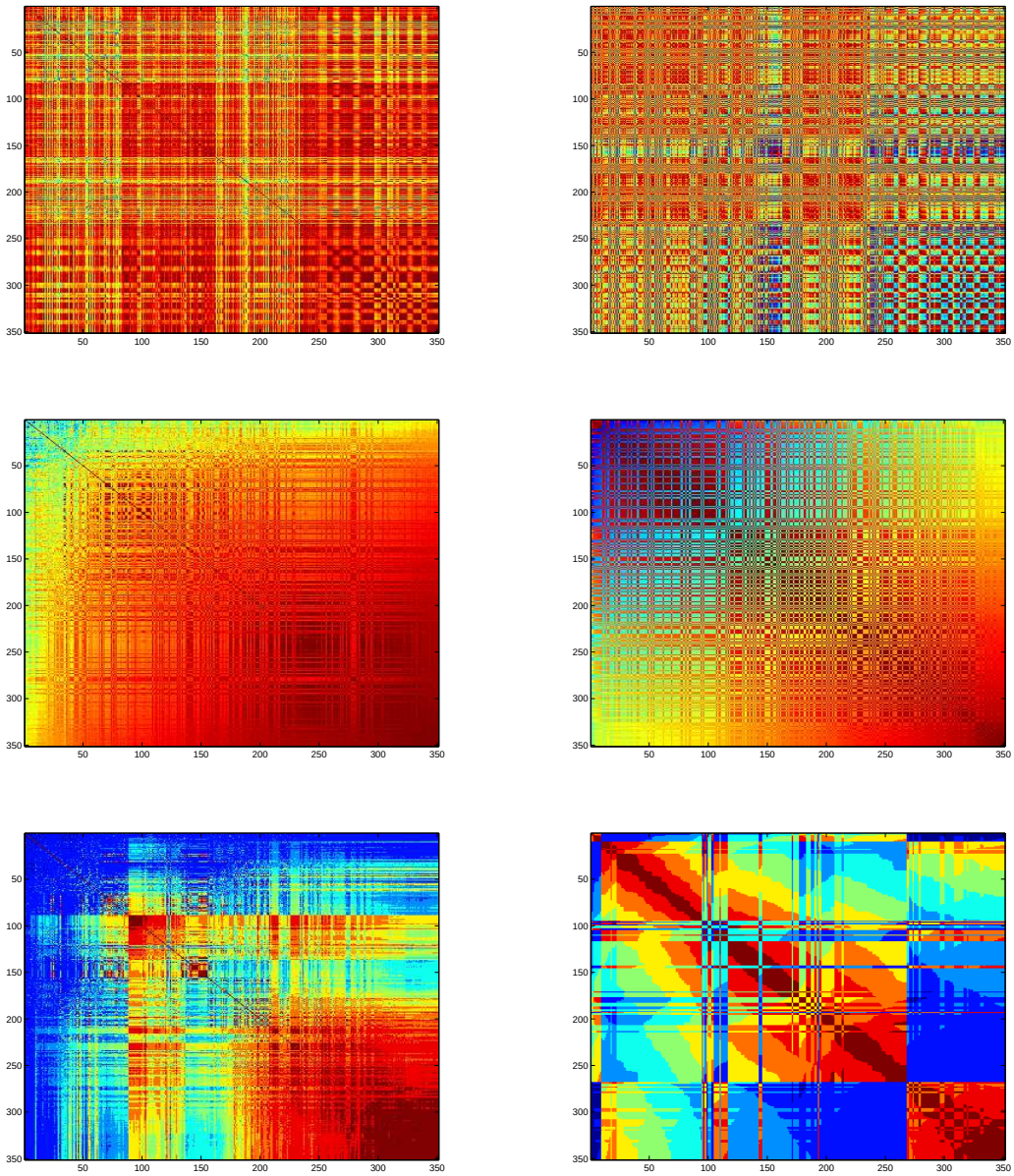

Figura 4.33: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *ion* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

Tabela 4.12: Comparativo de desempenho por acc dos classificadores LS-SVM para base *pid*: [23] *versus* Presente Trabalho. Valores médio e (desvio padrão) por grupo de amostras.

	Treinamento	Validação	Teste
AG e Orig.	0,8 (0,02) - 0,77 (0,01)	0,77 (0,02) - 0,76 (0,03)	0,75 (0,06) - 0,74 (0,05)
AG e Norm.	0,35 (0,02) - 0,35 (0,02)	0,35 (0,02) - 0,35 (0,02)	0,33 (0,09) - 0,33 (0,09)
PSO e Orig.	0,81 (0,01) - 0,79 (0,01)	0,77 (0,03) - 0,77 (0,03)	0,75 (0,04) - 0,75 (0,04)
PSO e Norm.	0,35 (0,01) - 0,35 (0,01)	0,34 (0,03) - 0,34 (0,03)	0,37 (0,07) - 0,37 (0,07)

Base pid

Os dados em sua forma original permitiram a obtenção de classificadores com melhor desempenho que os com dados padronizados, como revelam os resultados registrados na Tabela 4.12. Como é possível notar, os classificadores dos dados padronizados tiveram desempenho menor ou igual à metade dos alcançados utilizando dados sem alteração. Com relação ao desempenho dos classificadores de referência, pode-se afirmar que as LS-SVMs obtidas neste trabalho têm a mesma capacidade de classificar corretamente as amostras da base em questão.

As Figuras 4.34 e 4.36, embora diferentes, apontam a existência de dois grupos. As matrizes ordenadas na primeira figura são muito semelhantes à matriz de *kernel* ordenada por Minus na segunda figura. Por sua vez, na segundo figura, a matriz de proximidade ordenada por Minus aponta para a mesma direção das matrizes ordenadas pelo autovetor. A matriz de proximidade ordenada por Minus na Figura 4.35 também concorda com este resultado. Na Figura 4.37, apenas a matriz de *kernel* ordenada pelo Minus destaca-se pelo grande grupo de amostras sobre quase toda extensão da diagonal principal da matriz.

Base snr

Como registram os resultados da Tabela 4.13, os classificadores obtidos têm desempenho igual ao da referência. Nem a normalização das bases e nem o método de otimização tiveram influência sobre os resultados, que em geral na comparação de todos os desempenhos são próximos.

As Figuras 4.38 e 4.40, assim como as Figuras 4.39 e 4.41, formam pares de resultados quase idênticos não fosse a existência de pequenas diferenças nas permutações das linhas e colunas. No primeiro par, as matrizes ordenadas apontam claramente para a divisão do conjunto de amostras em dois grandes grupos, com grande concordância visual dos resultados da ordenação da matriz de *kernel*. No segundo par de figuras, os dois grupos são evidenciados apenas na matriz de proximidade ordenada pelo método Minus.

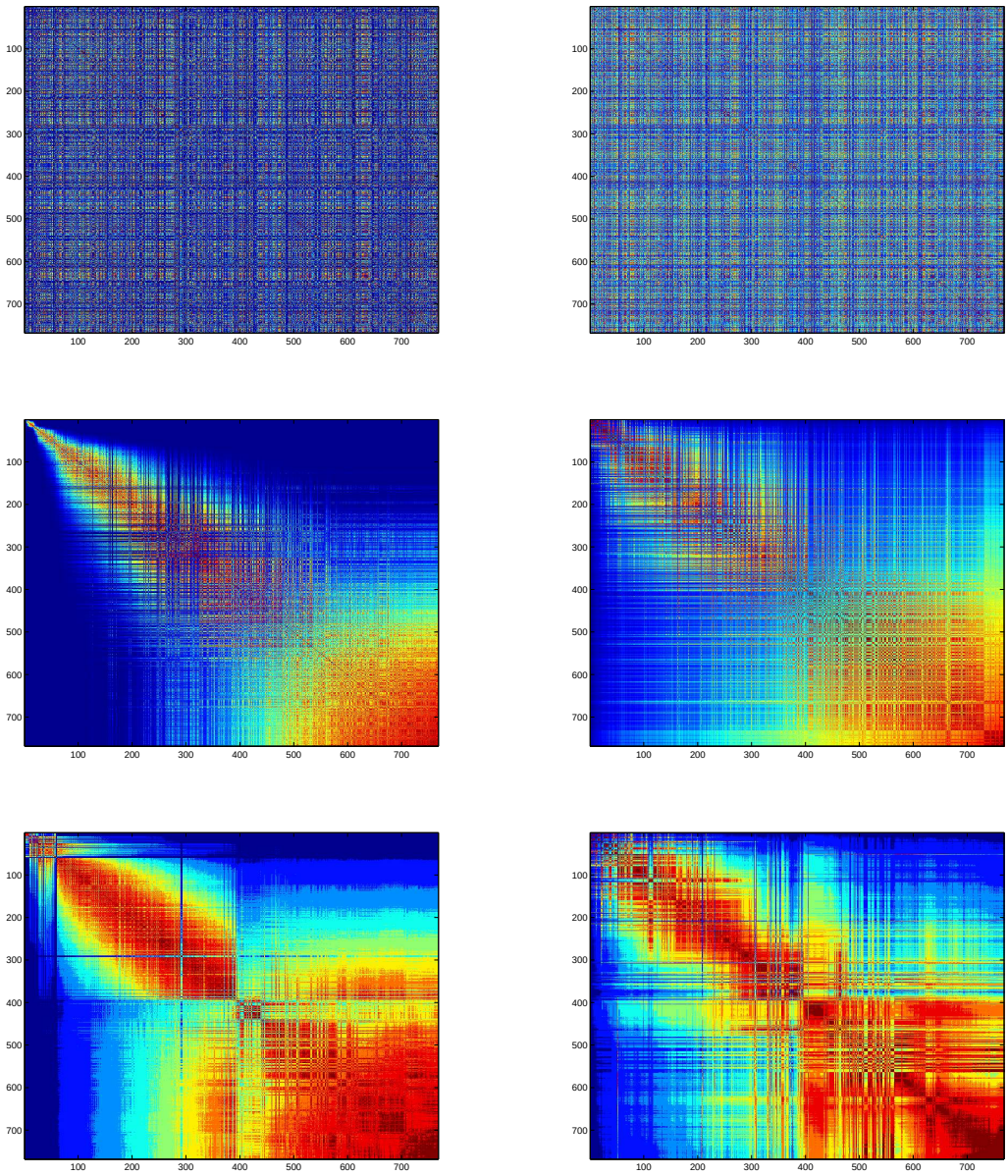

Figura 4.34: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *pid* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

Tabela 4.13: Comparativo de desempenho por acc dos classificadores LS-SVM para base *snr*: [23] *versus* Presente Trabalho. Valores médio e (desvio padrão) por grupo de amostras.

	Treinamento	Validação	Teste
AG e Orig.	1 (0) - 1 (0)	0,72 (0,05) - 0,72 (0,05)	0,69 (0,06) - 0,68 (0,06)
AG e Norm.	1 (0) - 1 (0)	0,74 (0,07) - 0,72 (0,08)	0,74 (0,05) - 0,71 (0,06)
PSO e Orig.	0,9 (0,04) - 1 (0)	0,71 (0,05) - 0,71 (0,05)	0,72 (0,06) - 0,72 (0,07)
PSO e Norm.	0,82 (0,05) - 1 (0)	0,71 (0,04) - 0,7 (0,05)	0,69 (0,06) - 0,68 (0,05)

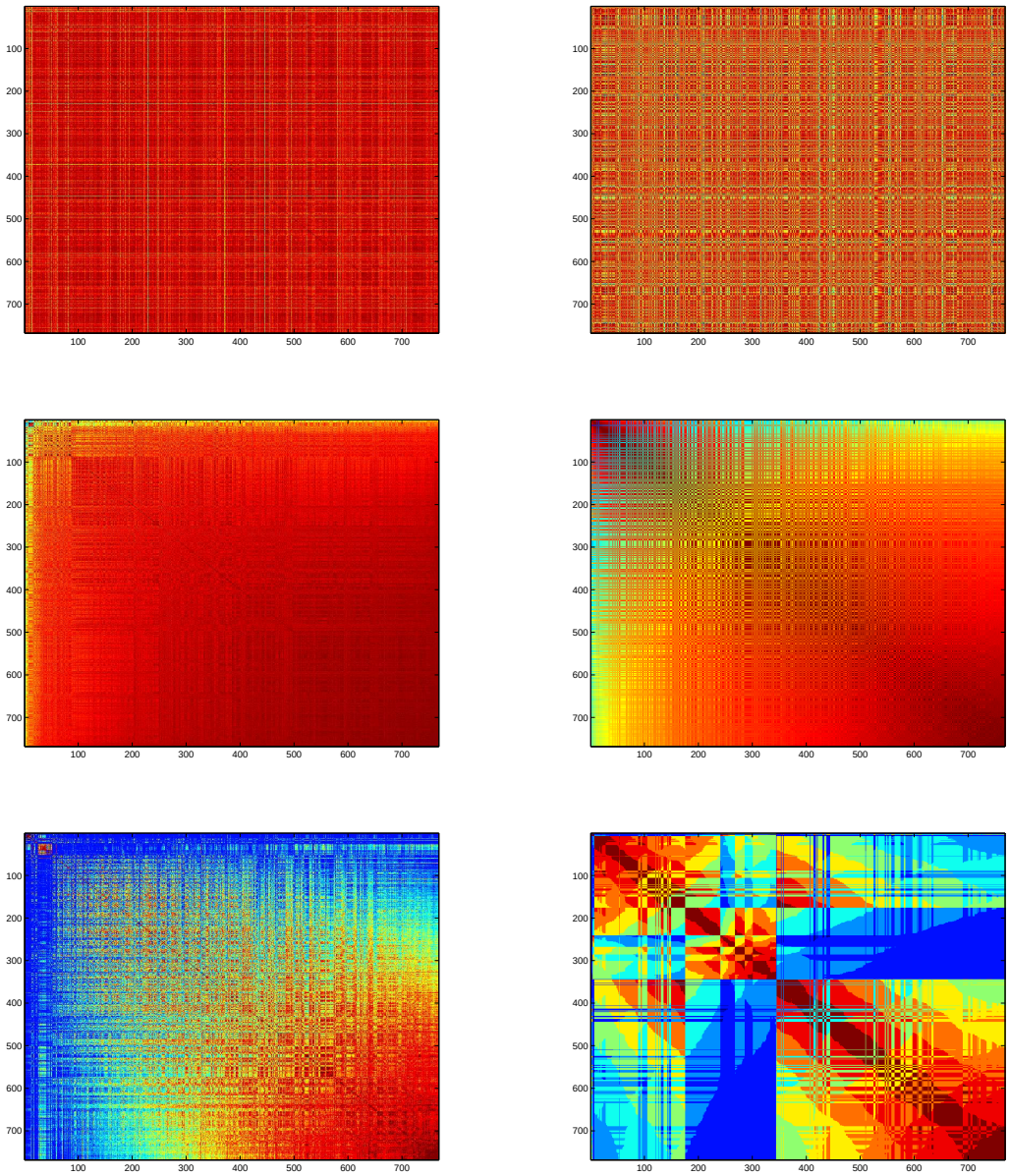

Figura 4.35: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *pid* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

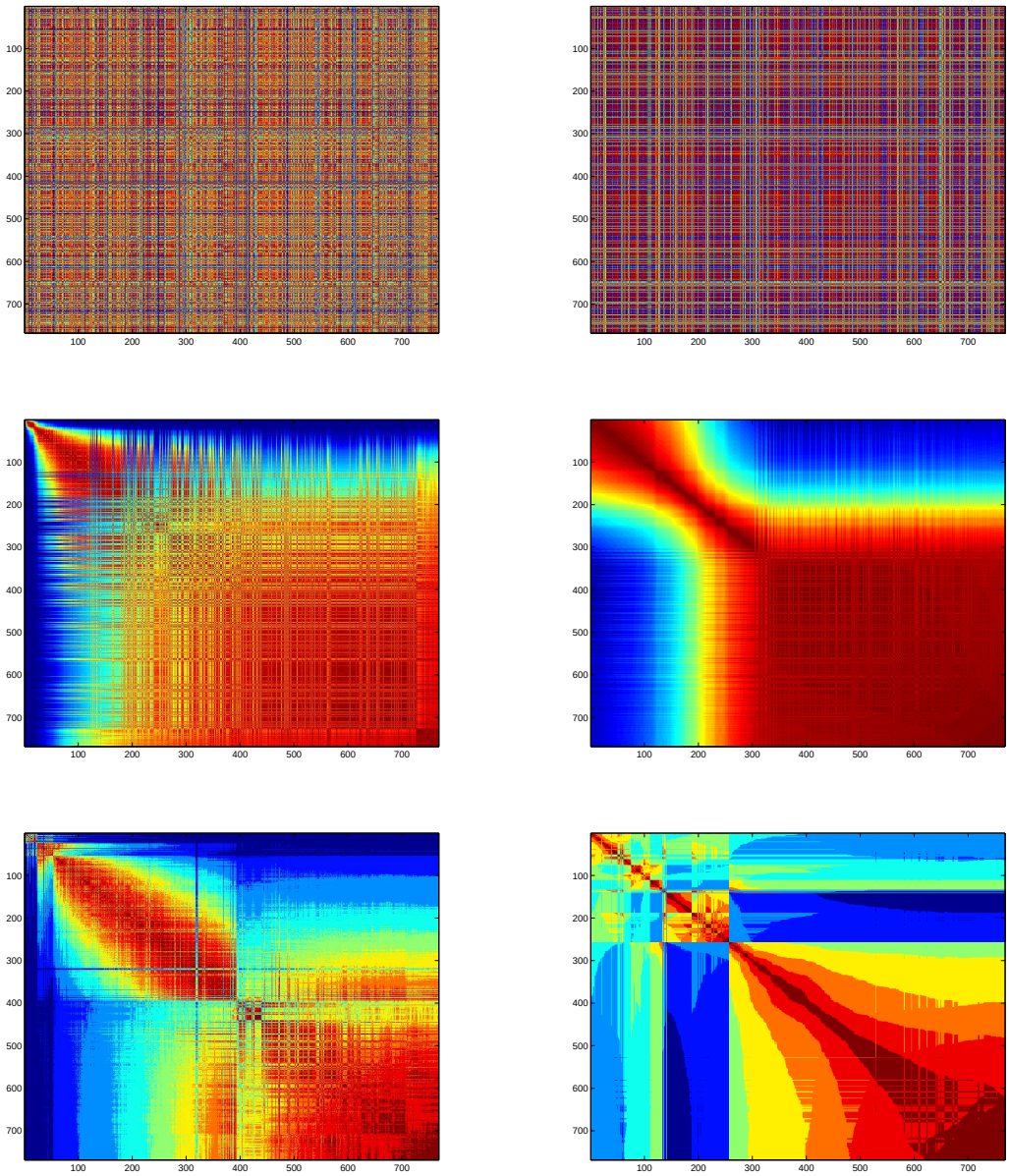

Figura 4.36: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *pid* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

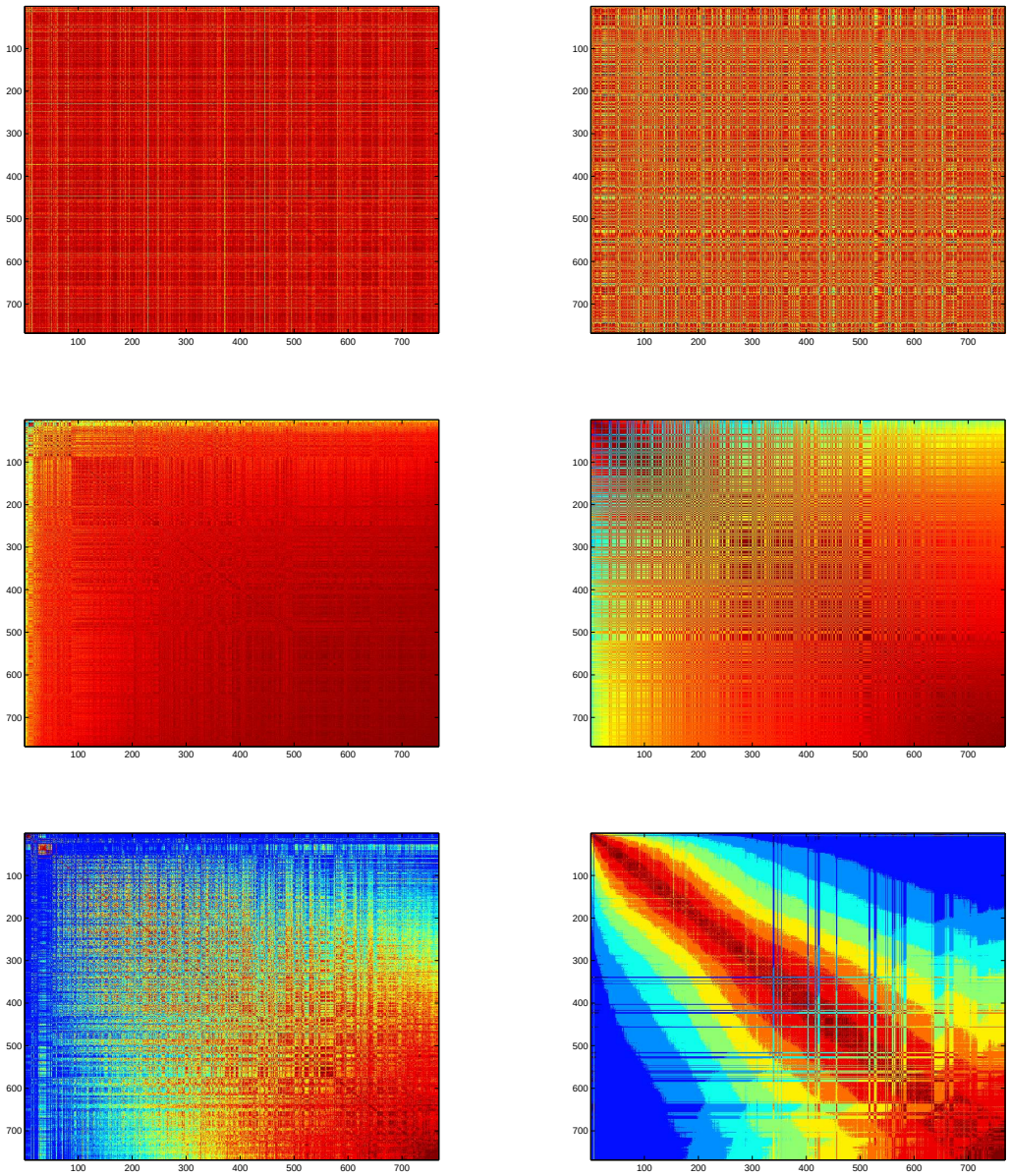

Figura 4.37: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *pid* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

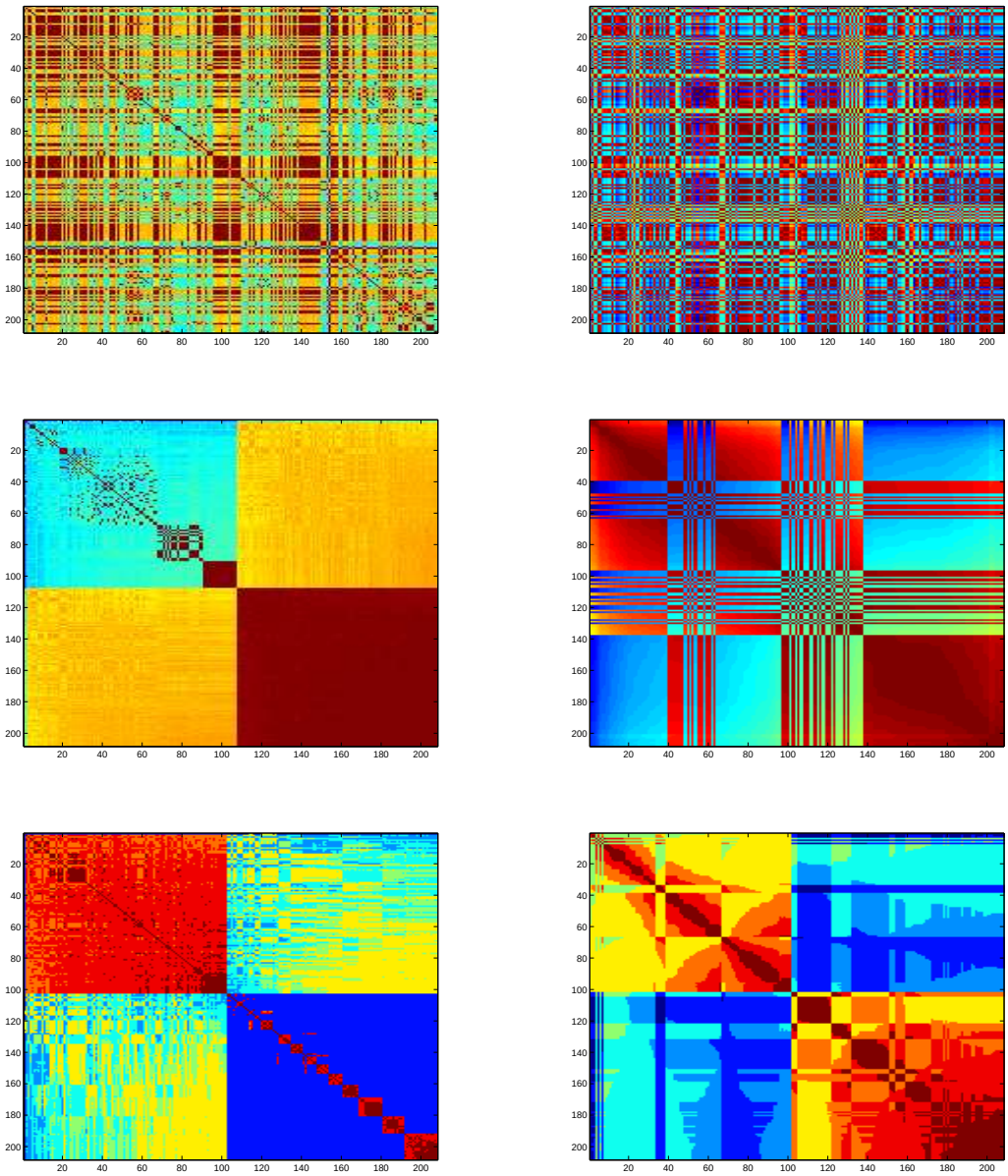

Figura 4.38: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *snr* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

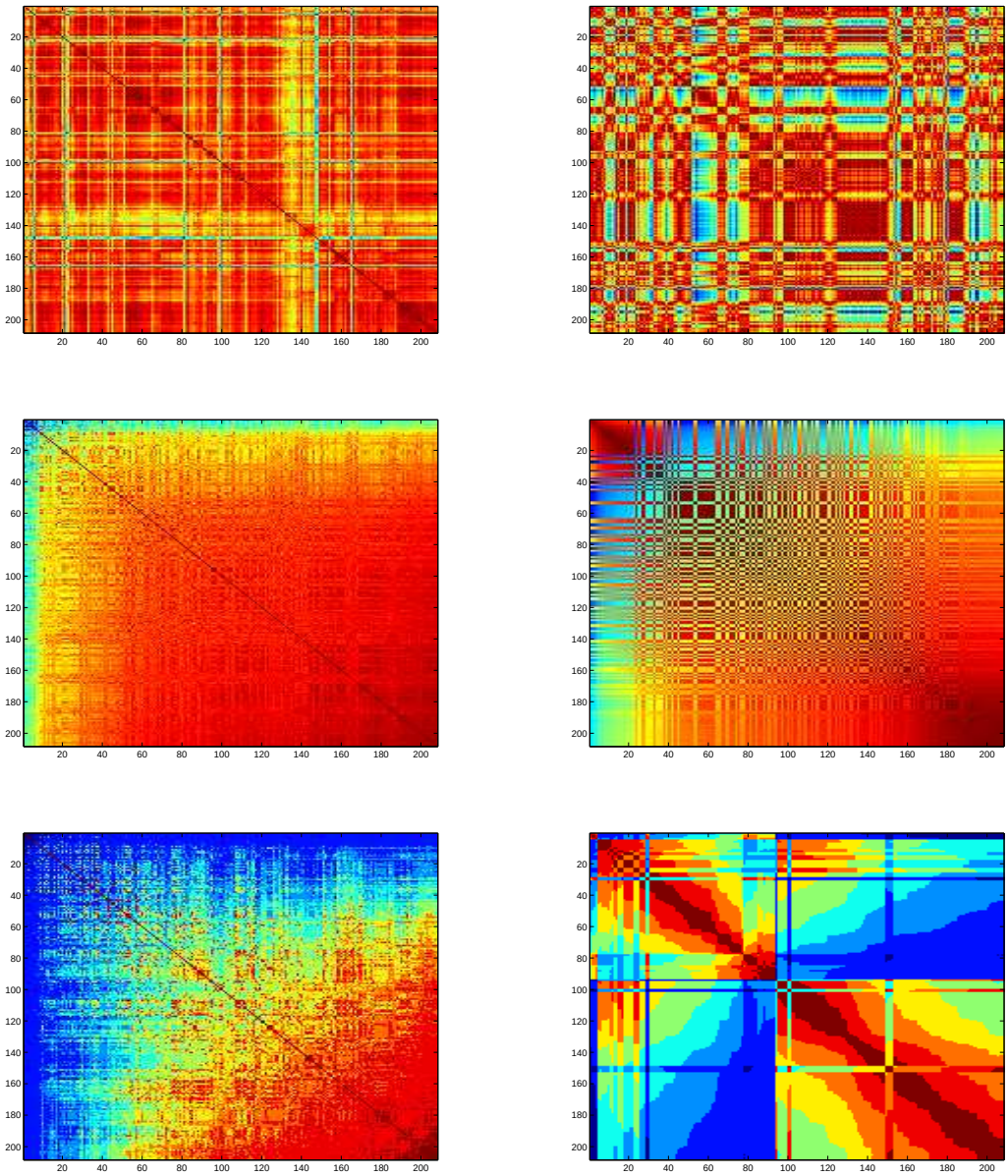

Figura 4.39: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *snr* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

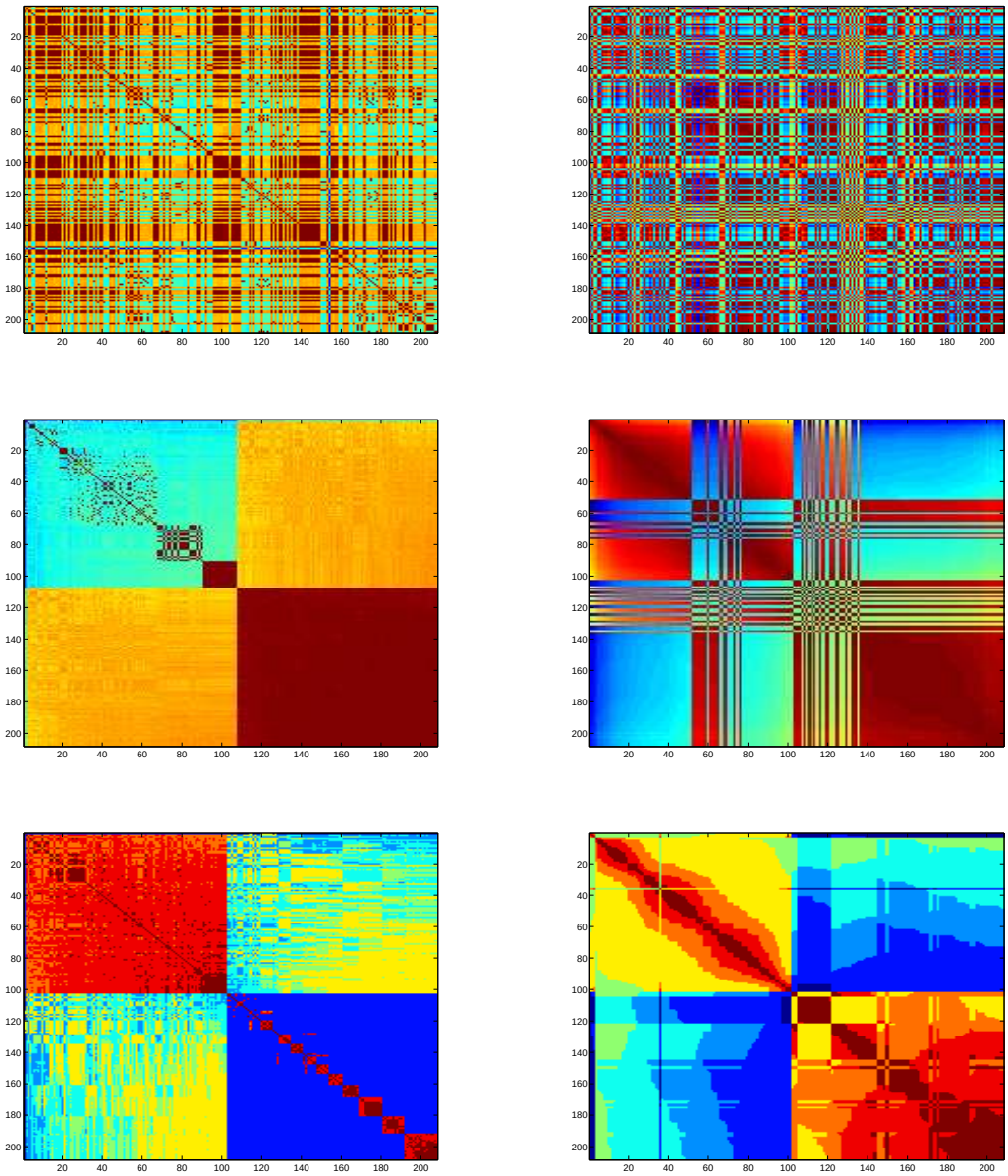

Figura 4.40: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *snr* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

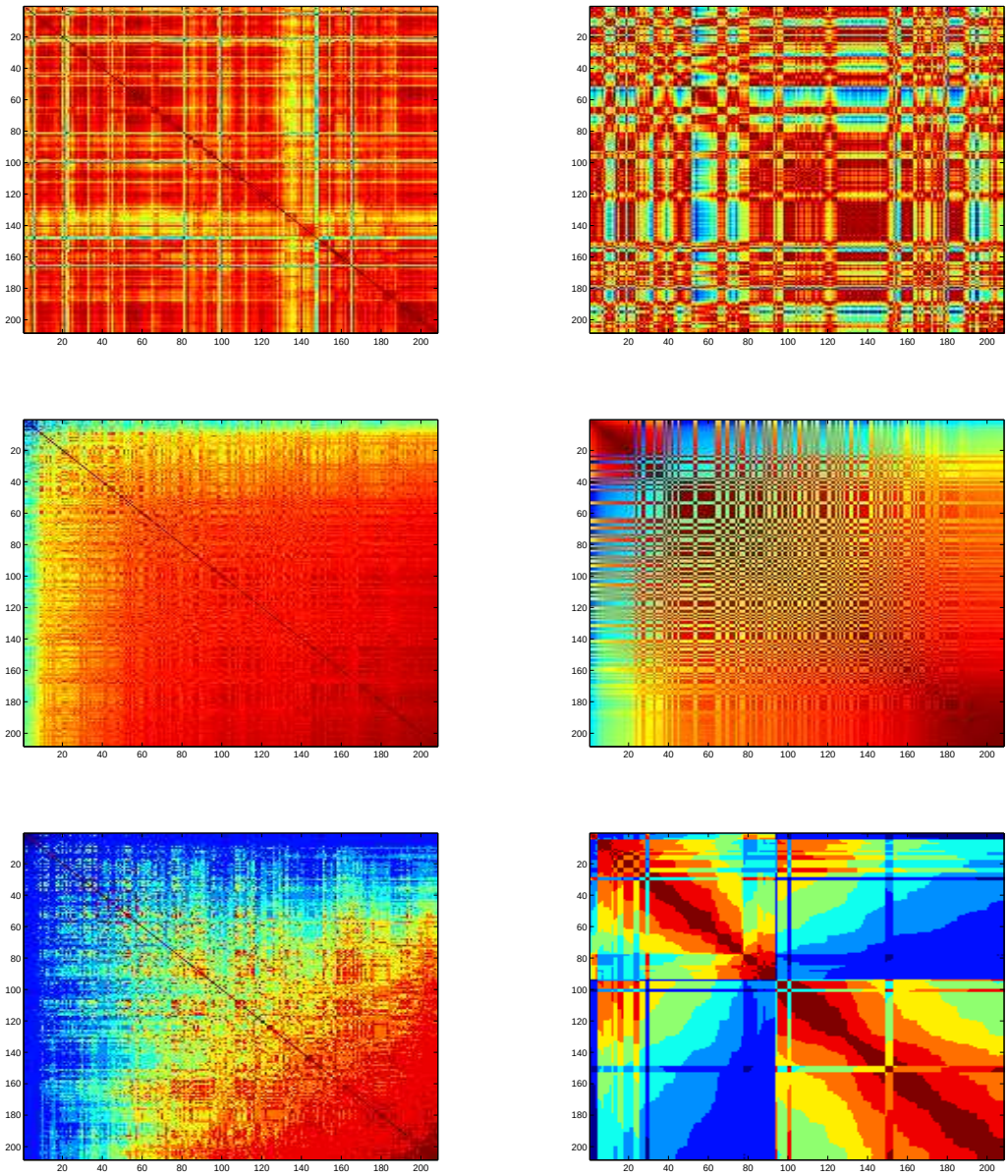

Figura 4.41: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *snr* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

Tabela 4.14: Comparativo de desempenho por acc dos classificadores LS-SVM para base *ttt*: [23] *versus* Presente Trabalho. Valores médio e (desvio padrão) por grupo de amostras.

	Treinamento	Validação	Teste
AG e Orig.	0,65 (0,02) - 0,65 (0,02)	0,66 (0,03) - 0,66 (0,03)	0,68 (0,03) - 0,68 (0,03)
AG e Norm.	0,99 (0) - 1 (0)	0,99 (0,01) - 0,92 (0,02)	0,98 (0,02) - 0,91 (0,05)
PSO e Orig.	0,99 (0) - 1 (0)	0,98 (0) - 0,65 (0,02)	0,98 (0,02) - 0,66 (0,04)
PSO e Norm.	0,9 (0,01) - 1 (0)	0,86 (0,02) - 0,9 (0,02)	0,86 (0,05) - 0,91 (0,05)

Tabela 4.15: Comparativo de desempenho por acc dos classificadores LS-SVM para base *wbc*: [23] *versus* Presente Trabalho. Valores médio e (desvio padrão) por grupo de amostras.

	Treinamento	Validação	Teste
AG e Orig.	0,86 (0,14) - 0,68 (0,09)	0,81 (0,17) - 0,65 (0,02)	0,81 (0,16) - 0,64 (0,06)
AG e Norm.	0,97 (0,01) - 0,96 (0,01)	0,96 (0,01) - 0,96 (0,02)	0,97 (0,02) - 0,97 (0,02)
PSO e Orig.	1 (0) - 1 (0)	0,96 (0,01) - 0,96 (0,02)	0,97 (0,02) - 0,97 (0,01)
PSO e Norm.	0,96 (0,01) - 0,96 (0,01)	0,97 (0,01) - 0,96 (0,01)	0,97 (0,02) - 0,97 (0,02)

Base ttt

Apenas os classificadores ajustados por AG utilizando dados sem padronização tiveram desempenho inferior a 0,9 para todos os grupos, segundo a Tabela 4.14. De modo geral, os melhores resultados foram atingidos pelos classificadores para os dados padronizados. Ainda assim, os resultados foram inferiores aos da referência. Quanto a visualização dos resultados nas matrizes ordenadas, em nenhuma das Figuras 4.42, 4.43, 4.44 e 4.45 grupos bem definidos foram evidenciados.

Base wbc

A padronização das amostras possibilitou melhores desempenhos com o AG, embora o mesmo não tenha acontecido com o PSO. Ainda segundo os registros da Tabela 4.15, os classificadores com melhores desempenhos são muito bons e comparáveis aos da referência.

As matrizes das Figuras 4.46, 4.47, 4.48 e 4.49 impressionam pela sua clareza e semelhança com suas correspondentes em cada figura mesmo quando ordenadas por métodos diferentes. As duas submatrizes sobre a diagonal principal são destacadas de tal forma que fica evidente a quantidade de grupos de amostras.

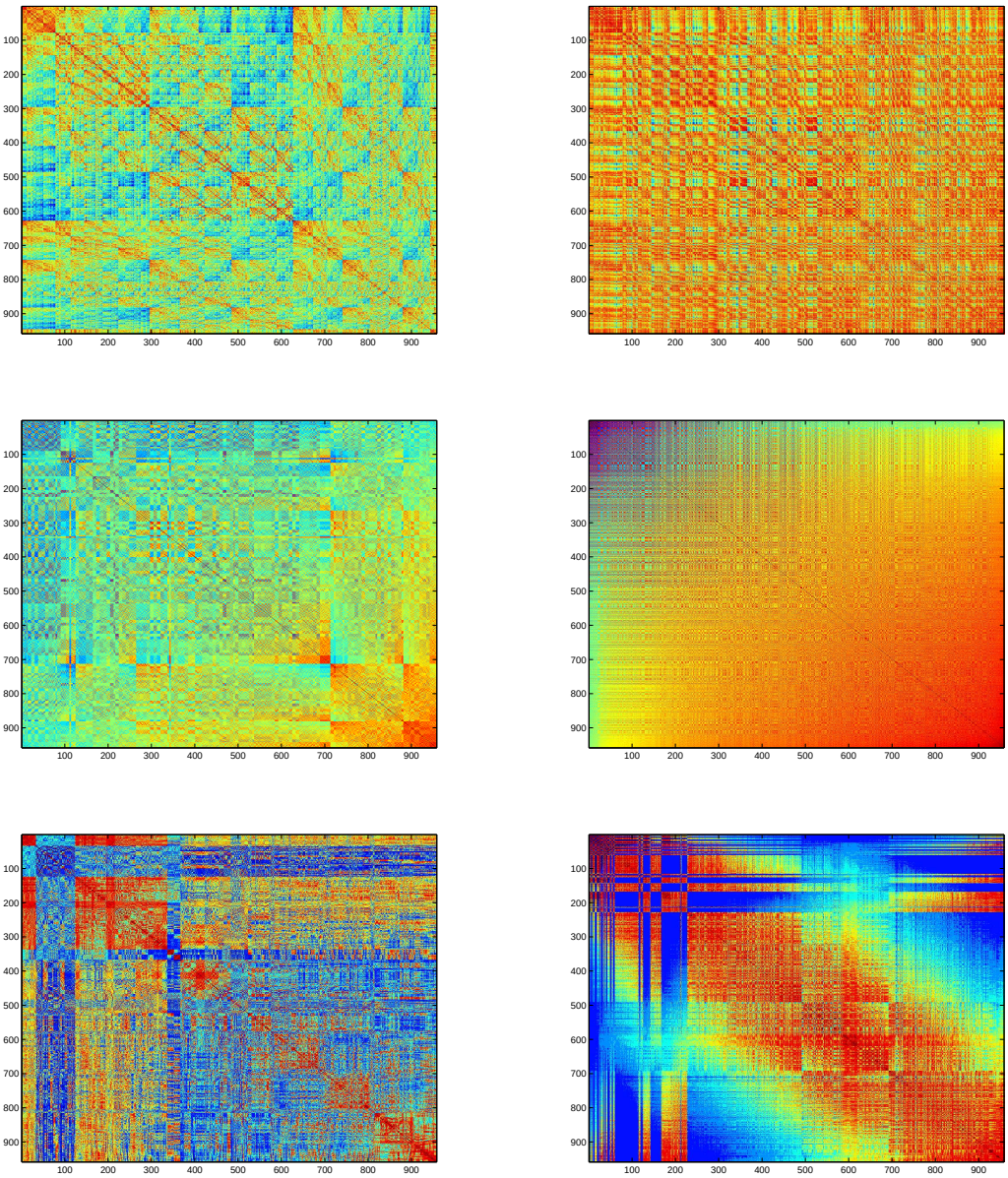

Figura 4.42: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *ttt* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

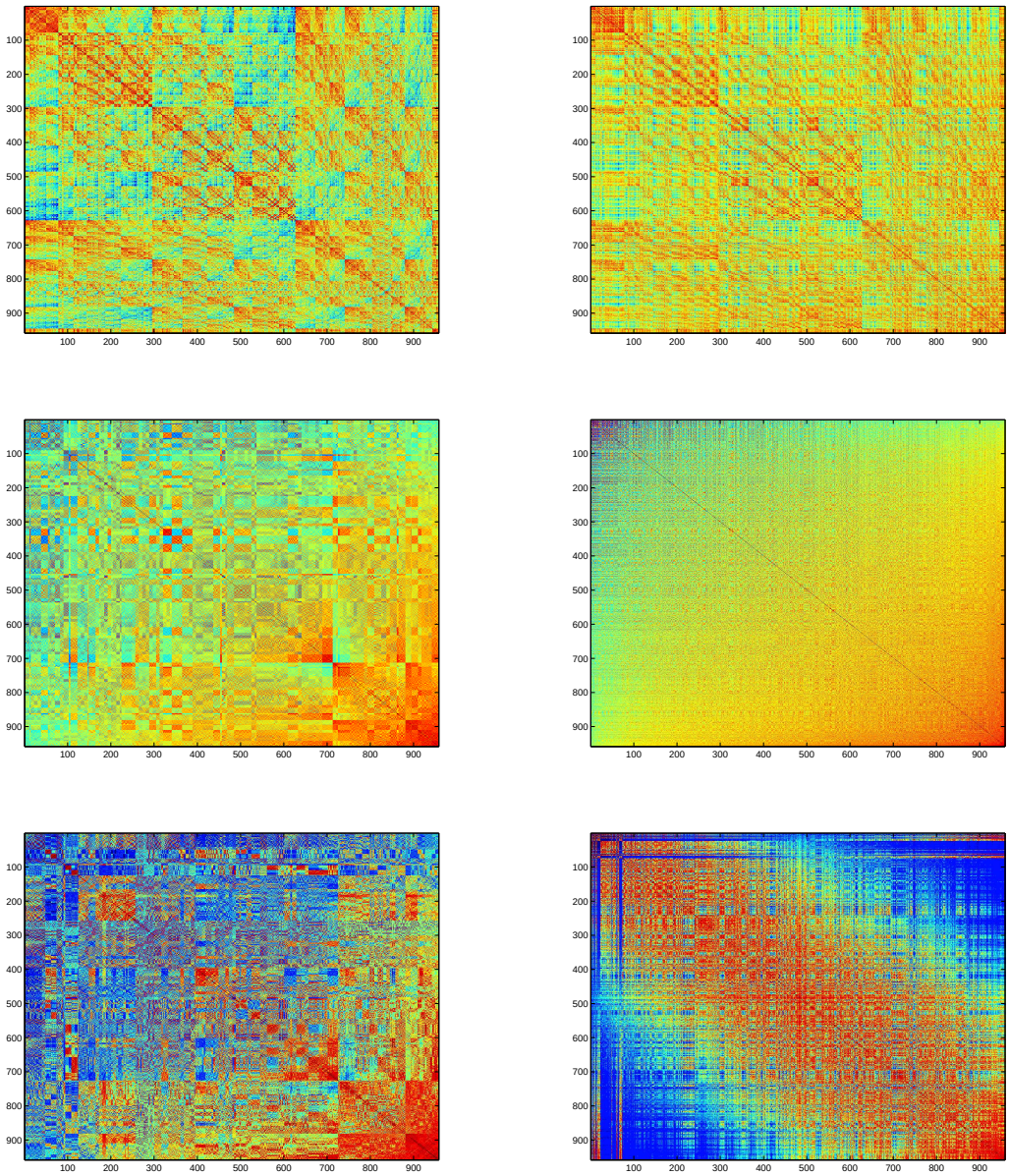

Figura 4.43: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *ttt* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

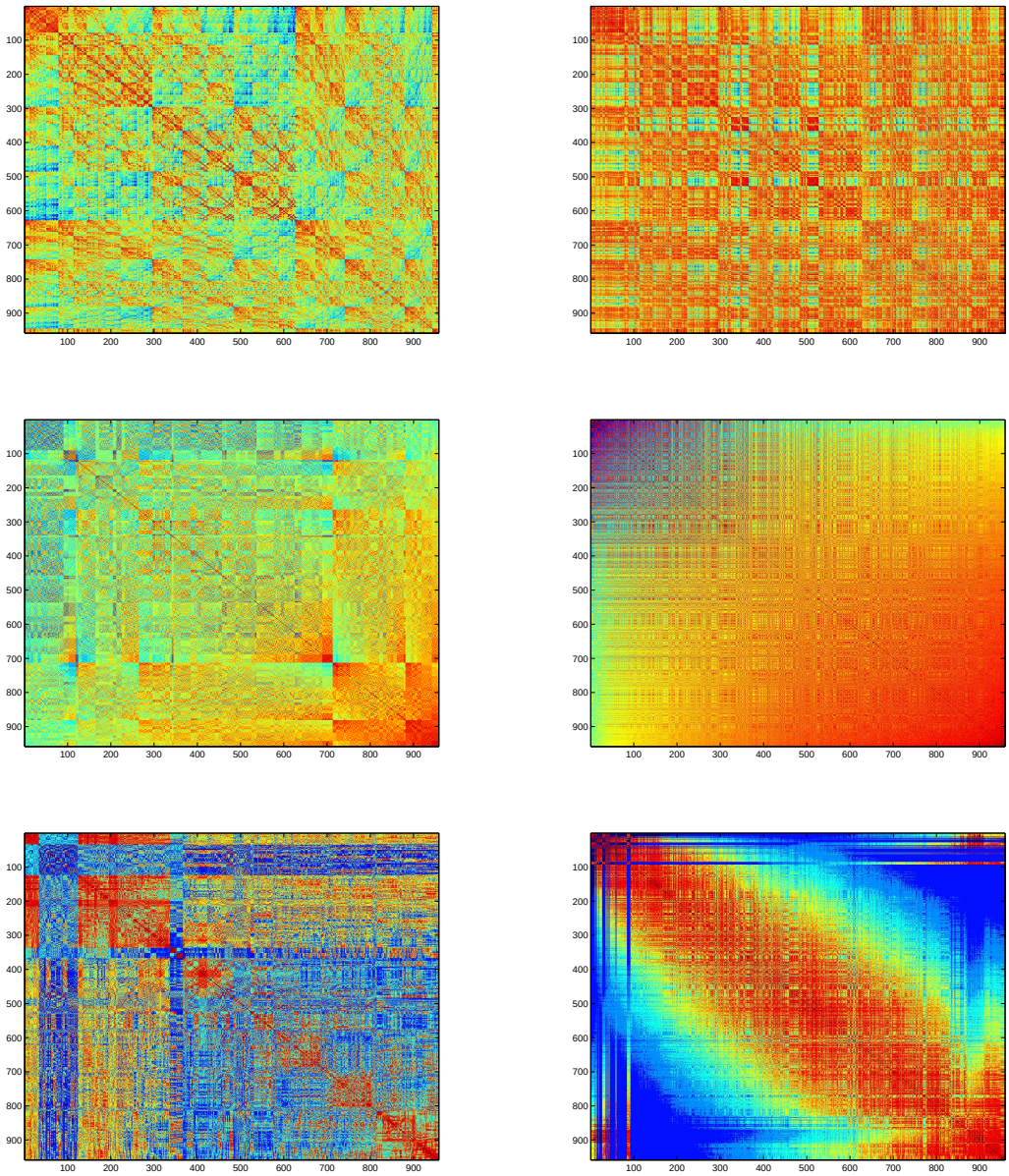

Figura 4.44: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *ttt* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

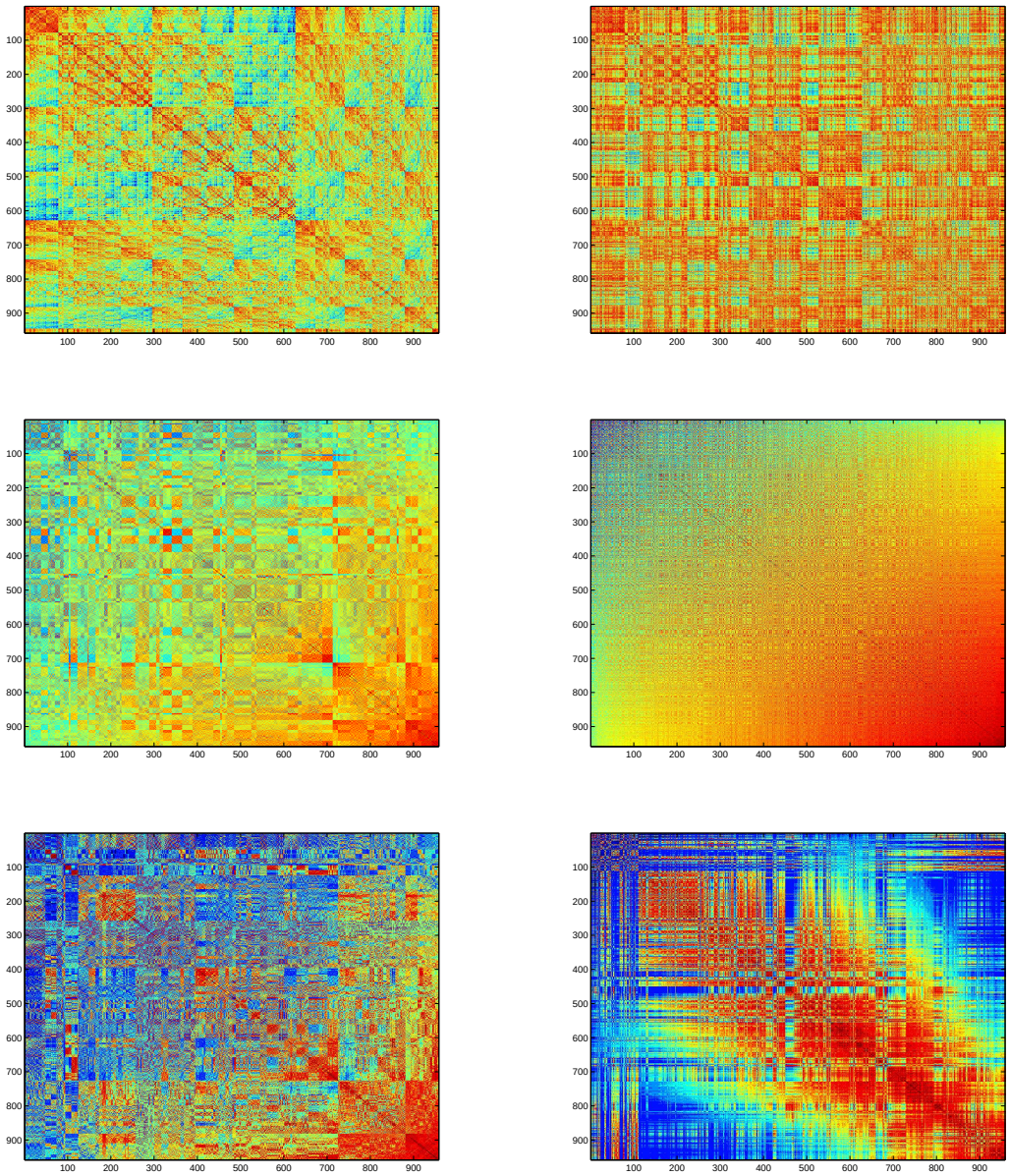

Figura 4.45: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *ttt* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

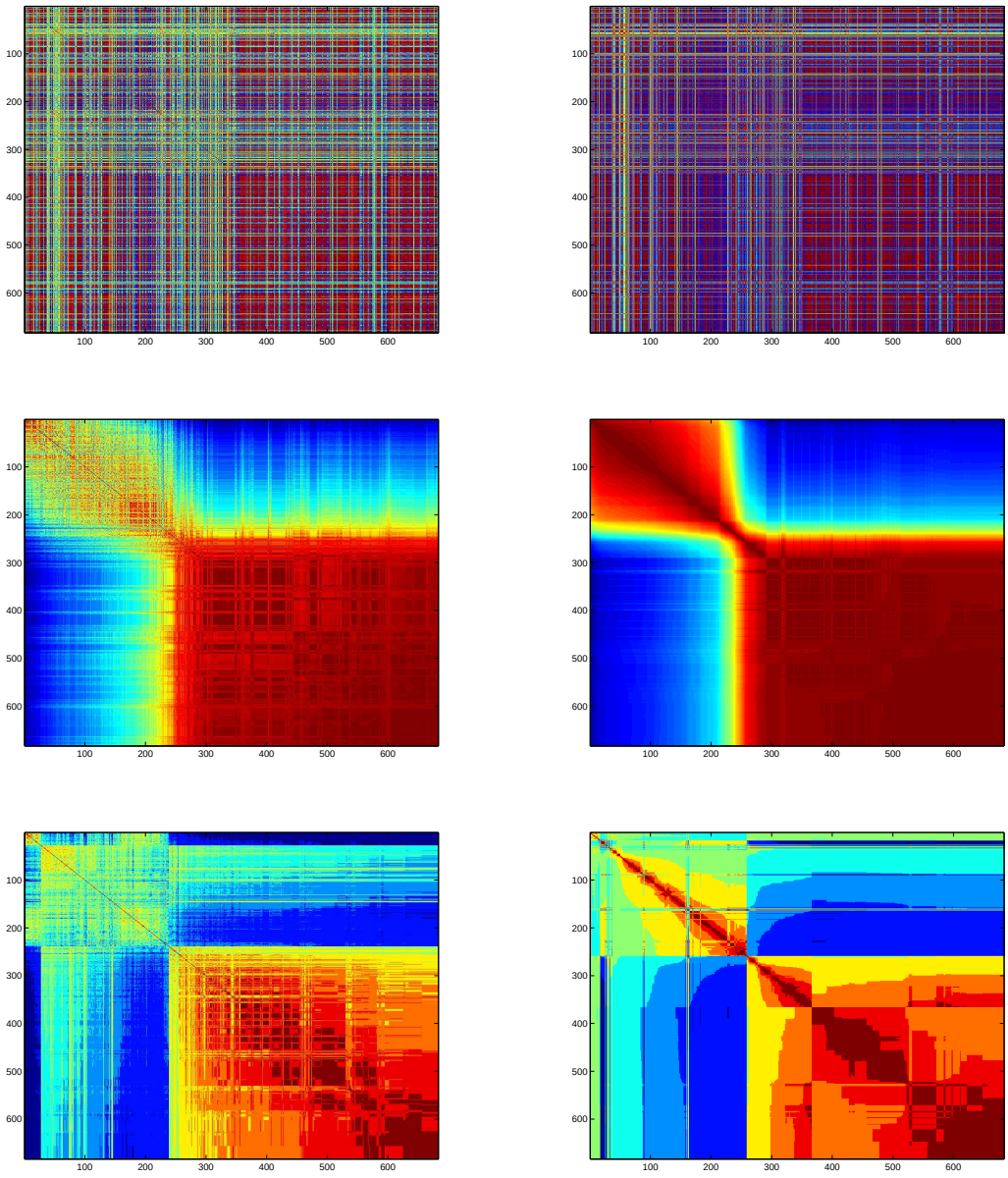

Figura 4.46: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *wbc* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

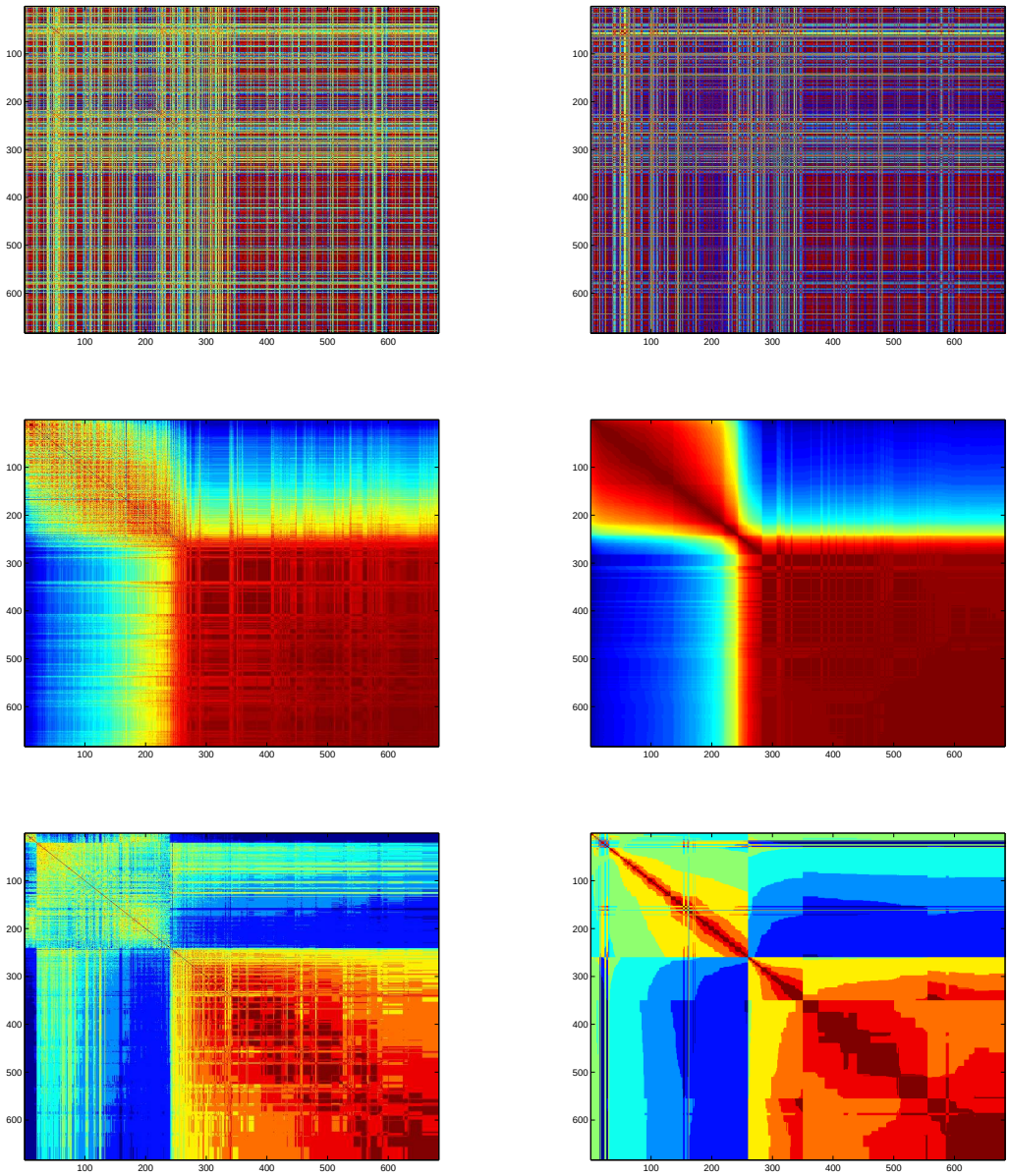

Figura 4.47: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *wbc* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

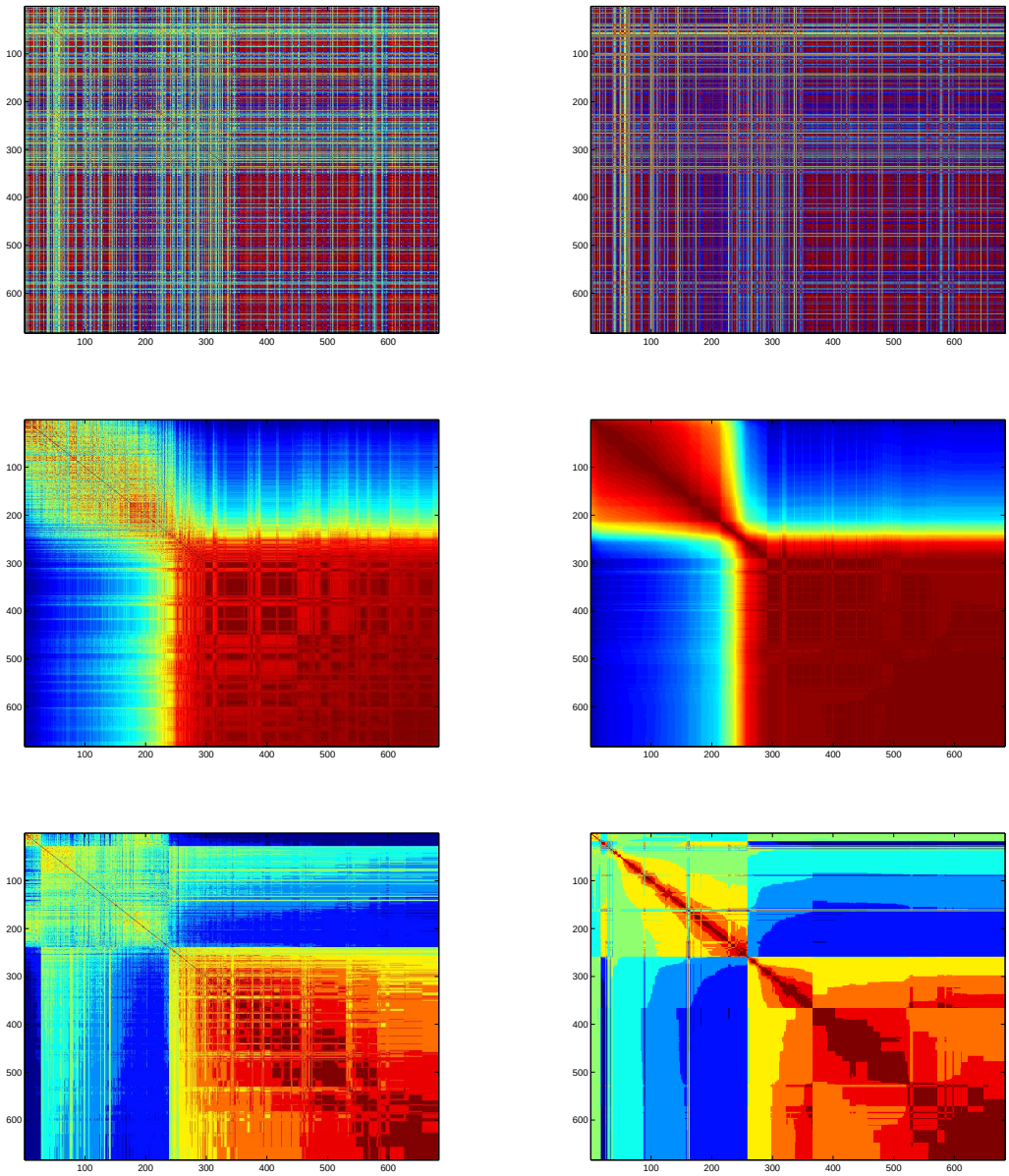

Figura 4.48: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *wbc* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

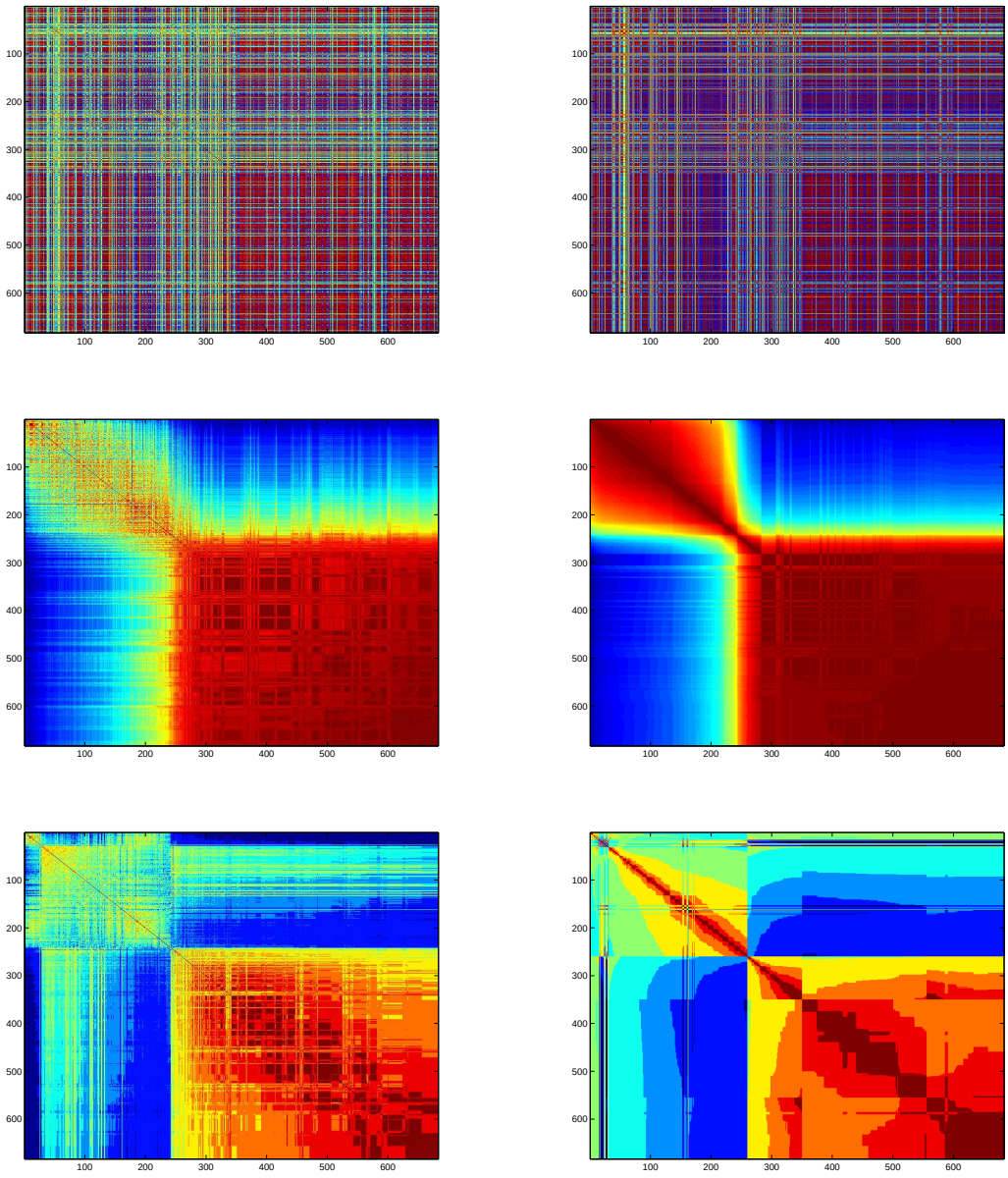

Figura 4.49: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *wbc* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

4.3 Experimento 3

Base img

Antes de avaliar as matrizes ordenadas desta base é preciso destacar que os parâmetros encontrados não são os mais apropriados para os dados sem padronização. Isso porque o número de grupos obtido foi acima de 300, ficando muito além do esperado e sendo da mesma ordem de grandeza do número de amostras dos conjuntos (853 municípios de Minas Gerais). Além disso, são muito baixos os valores apurados para A para os dados não padronizados. Segundo a Tabela 4.4, os valores de A não estão tão próximos da unidade quanto os valores da função de custo para os dados padronizados. Por esse motivo, as Figuras 4.50 e 4.52 revelam que os valores ajustados para σ não possibilitaram a identificação de amostras semelhantes. Seja pelo raio da função gaussiana, seja pelo número de grupos, respectivamente, as matrizes de *kernel* e de proximidade não demonstram similaridade entre as amostras. É possível notar que mesmo nessa condição, o nivelamento por quartis no método Minus realça algumas relações. Contudo, esses resultados devem ser descartados em razão do resultado insatisfatório dos algoritmos de otimização para os dados sem padronização desta base.

Por sua vez, as Figuras 4.51 e 4.53 também não deixam claro o número de conjuntos. Nestas duas figuras, as matrizes de proximidade ordenadas pelo autovetor apontam a existência de dois grupos. Já as matrizes de proximidade ordenadas pelo Minus apresentam quatro e seis grupos. A observação dos mapas na Figura 4.54 permite comparar a quantidade de grupos e como cada um é constituído por municípios geograficamente próximos. Os resultados descartados, referentes ao dados não normalizados, apresentam grande número de partições do conjunto, o que não parece razoável se forem comparados aos mapas disponíveis em [17].

Os mapas em que os municípios estão divididos em quatro e três grupos apresentam grande quantidade de municípios próximos uns dos outros e concentrados em poucas regiões do Estado. A cor atribuída a cada município refere-se ao número (rótulo) do grupo a que o município pertence. Os municípios dos grupos mais baixos recebem cores mais azuladas, enquanto que os de grupos mais elevados, recebem cores mais avermelhadas. Estes mapas com o agrupamento dos municípios em subconjuntos conforme o valor médio obtido para c com dados padronizados, portanto, estão em concordância com o que é de amplo de conhecimento da sociedade com relação à importância econômica, ao tamanho das populações e à qualidade de vida dos municípios mineiros (ver mapas na Figura 4.55 reproduzidos de [12]). A Tabela 4.16

apresenta os valores médios das características de cada um dos grupos de municípios para os agrupamentos em que foram utilizados dados normalizados.

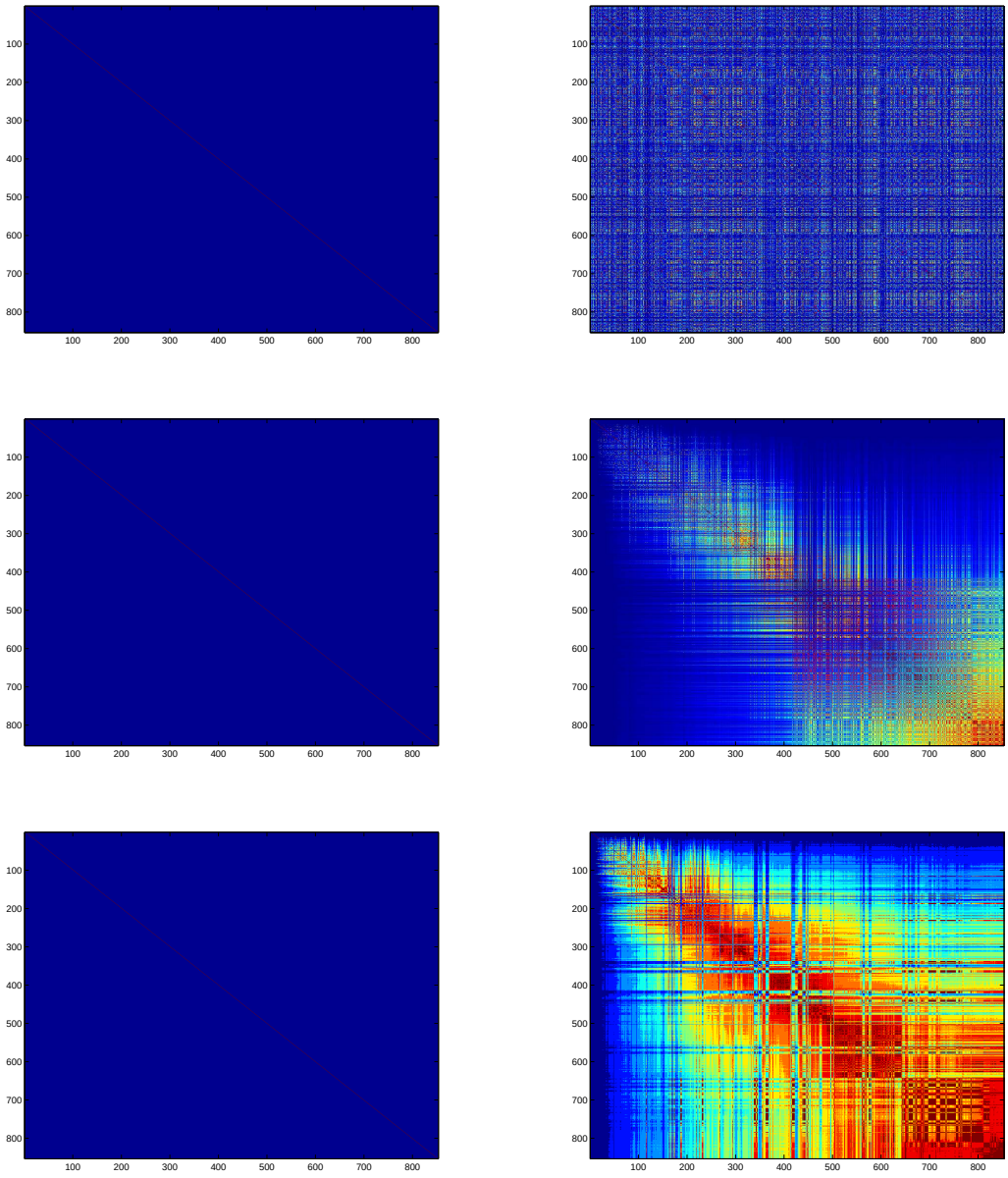

Figura 4.50: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *img* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

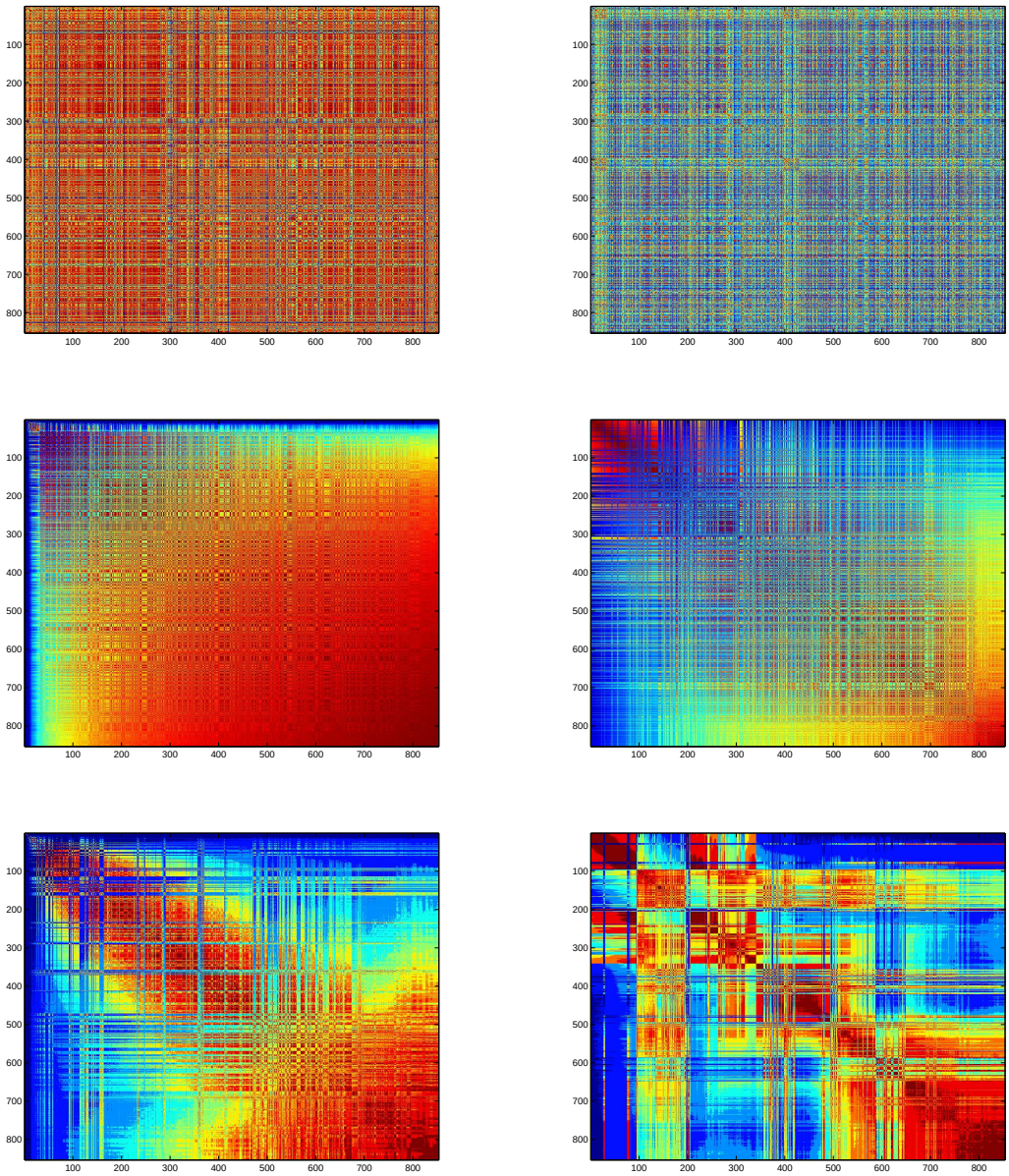

Figura 4.51: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após AG para a base *img* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

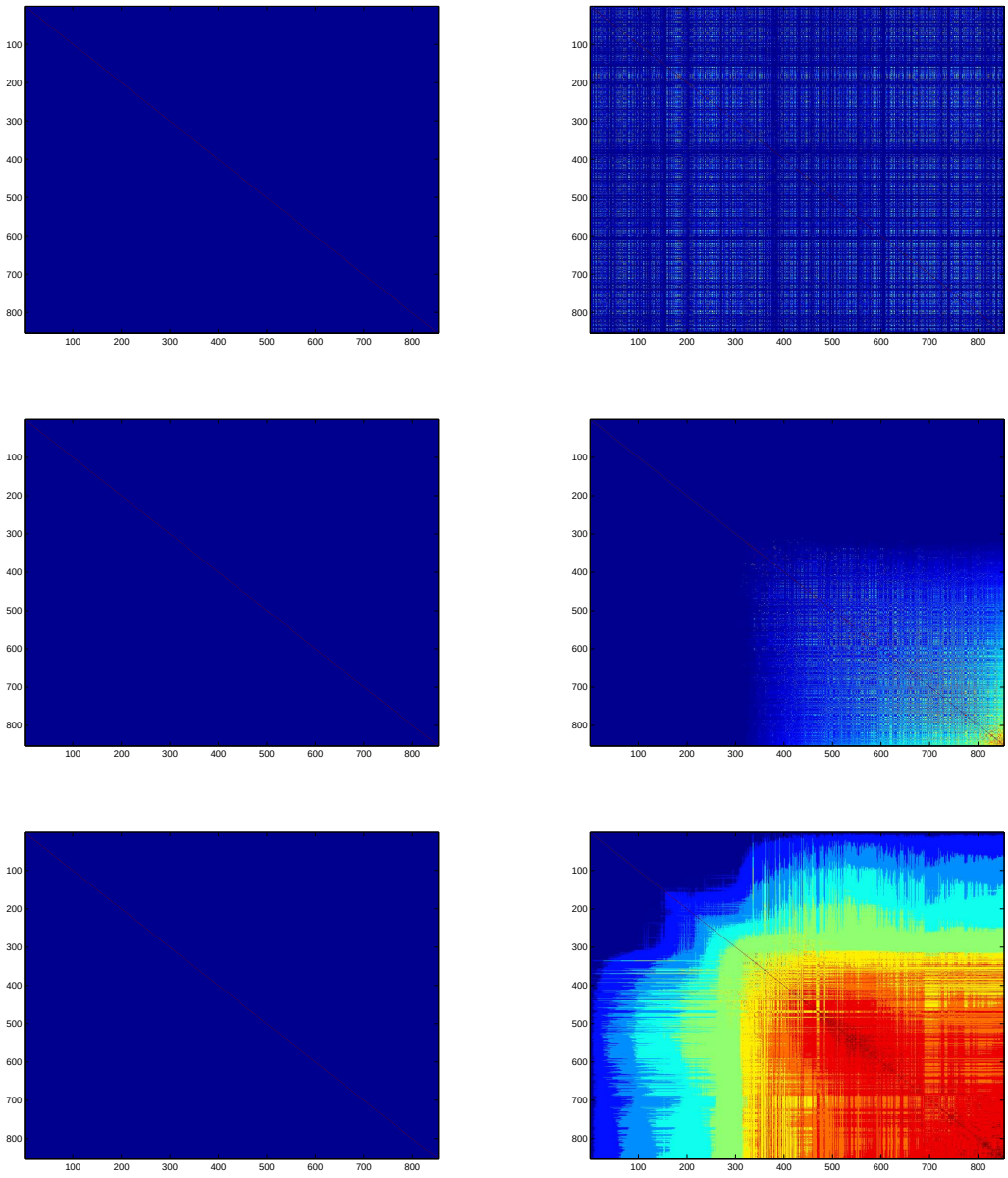

Figura 4.52: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *img* original. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

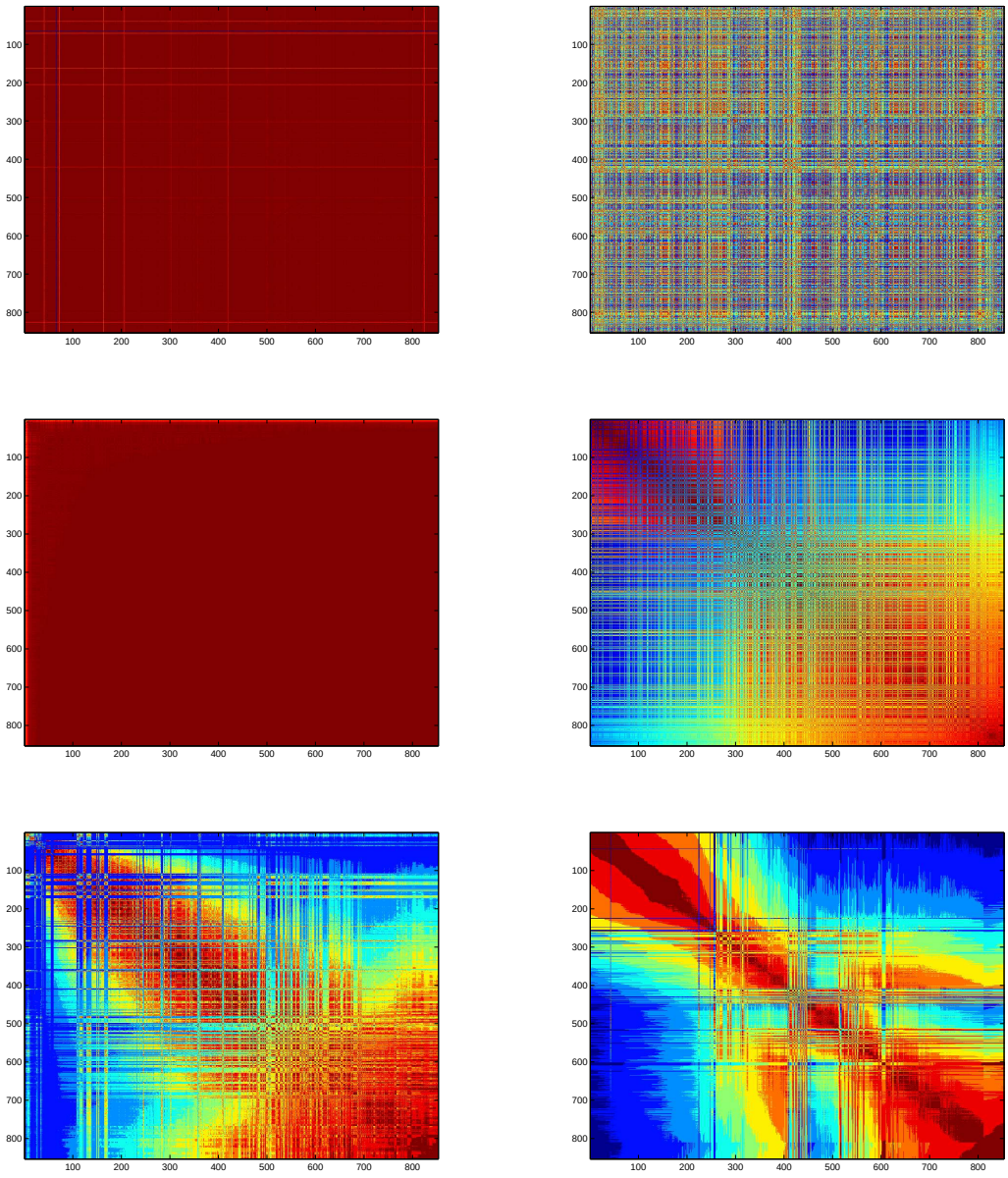

Figura 4.53: Matrizes de *kernel* (na coluna da esquerda) e de proximidade (na coluna da direita) após PSO para a base *img* padronizada. No topo estão as matrizes originais, seguidas ao centro pelas ordenadas pelo autovetor, e abaixo pelas ordenadas por Minus.

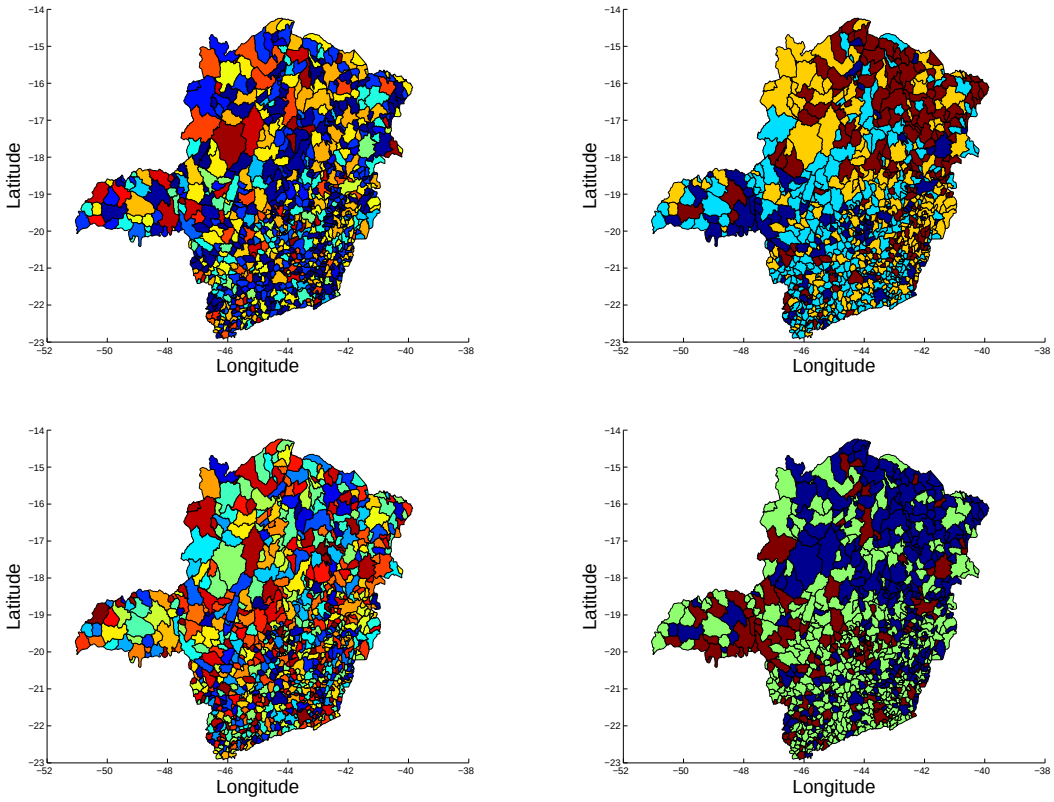

Figura 4.54: Mapas de grupos de municípios de Minas Gerais segundo resultados do AG (acima) e do PSO (abaixo) conforme o uso de dados com valores originais (à esquerda) e normalizados (à direita).

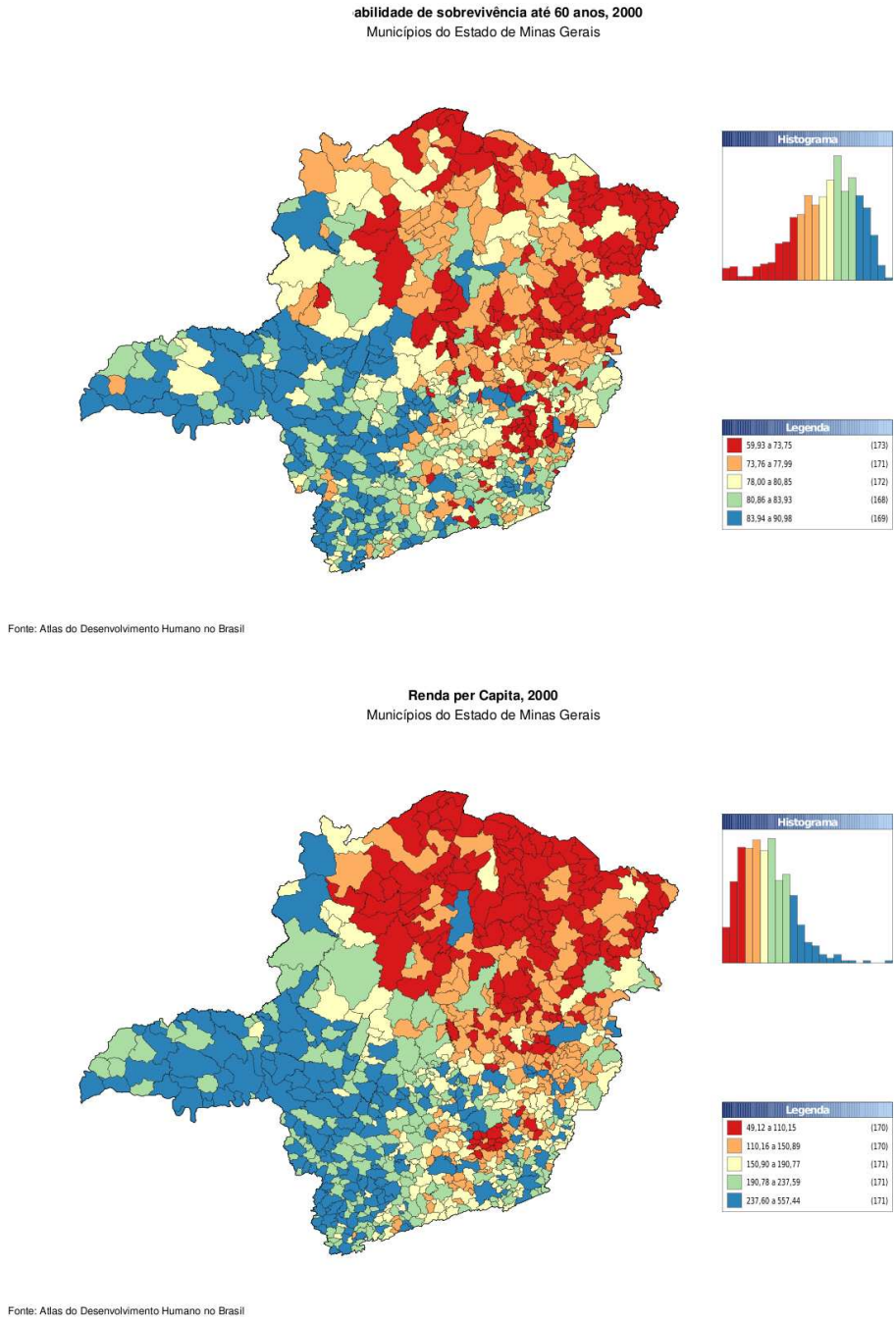

Figura 4.55: Reprodução de mapas de grupos de municípios de Minas Gerais para estudos socioeconômicos. Atlas do Desenvolvimento Humano no Brasil versão 1.0.0 [12].

Tabela 4.16: Comparativo dos agrupamentos para base *img* por AG e por PSO utilizando dados normalizados. Valores médios das características de cada grupo de amostras para uma execução do FCM utilizando o valor de c determinado pela metodologia deste trabalho. PIB e PIB per capita em R$ de 2000(mil).

Método	Grupo	Municípios	PIB per capita	Domicílios	IDH	IMRS	Luz Elétrica	PIB	População
GA	1	94	15720,47	28503,96	0,791	0,679	27984,53	871939,69	103084,08
	2	320	5177,18	3529,04	0,757	0,632	3384,33	50716,29	12679,65
	3	254	3650,16	2438,83	0,699	0,574	2203,03	26287,19	9445,36
	4	185	2676,69	2211,72	0,646	0,480	1719,97	18361,56	9432,32
PSO	1	289	2920,71	2285,76	0,659	0,507	1855,15	20520,92	9497,99
	2	425	4457,20	2794,08	0,738	0,614	2643,44	35247,66	10261,6
	3	139	13081,68	21505,24	0,787	0,671	21023,73	628450,24	77592,76

4.4 Conclusões do capítulo

Teve-se a oportunidade de avaliar os métodos quanto a diferentes bases e procedimentos neste capítulo. Não apenas com a apresentação dos índices de desempenho, mas também com a visualização complementar das matrizes, foi possível estabelecer relações com maior grau de conhecimento e de forma coerente com o objetivo e os resultados. No próximo capítulo, os experimentos têm seus resultados analisados sob um visão mais ampla, a partir da qual são estabelecidas as conclusões deste trabalho.

CAPÍTULO

5

Conclusão

Antes de tecer as conclusões sobre a metodologia proposta, é preciso analisar de uma perspectiva mais ampla os resultados dos experimentos. Portanto, são feitos alguns apontamentos a partir dos quais serão registrados os aspectos principais do trabalho e sugestões para a continuidade deste.

5.1 Considerações sobre os resultados

Em relação aos resultados de [36], percebe-se nos resultados desta dissertação a mudança substancial no valor dos parâmetros ajustados. Acredita-se que essa diferença se deva primordialmente à utilização da validação cruzada. Isso porque essa última altera a constituição dos grupos de treinamento, validação e teste, que passam a contar com partes menores do conjunto total de amostras e porque os parâmetros ajustados resultam da média dos resultados das dez validações e não apenas de uma avaliação do conjunto completo. Como a validação cruzada é o procedimento mais comumente adotado e que busca propiciar maior capacidade de generalização ao classificador obtido, os resultados alcançados neste trabalho são mais alinhados com o objetivo de estabelecer comparações com outros trabalhos.

Nos demais resultados, os valores encontrados para σ em cada uma das bases não são os mesmos registrados no trabalho de referência [23]. Tal diferença pode ser explicada segundo o argumento da inclusão da informação estrutural na matriz de *kernel*. Como exposto anteriormente, o ajuste dos parâmetros tendo como restrição a preservação das relações existentes entre as amostras no espaço de entrada tem como resultado valores significativamente

diferentes na maioria das bases de dados.

Outra explicação para a diferença dos valores encontrados para os parâmetros σ e c é a influência da normalização dos dados. Em alguns casos, tanto o valor encontrado para σ como o de c são muito diferentes quando cada uma das características passa a ter média nula e desvio padrão unitário. Mesmo assim, o desempenho dos classificadores não foi afetado em grande medida na maioria das bases de teste, exceto para a base **pid**. A transformação dos valores das características para o intervalo $[0, 1]$ pode ser também uma alternativa à padronização com média nula e desvio padrão unitário. Ficou claro nos experimentos que a alteração dos dados a serem submetidos às metodologias evita que diferenças de magnitude entre as características interfiram no treinamento e na qualidade da maioria dos classificadores e dos agrupamentos. Portanto, o procedimento a ser adotado deve ser o da normalização prévia dos dados, como destacam [34, 39, 49, 7].

Para várias das bases o número de grupos encontrado é o mesmo do número de classes principalmente nos casos em que os dados são submetidos à normalização na etapa inicial. Por isso, não ficam bem estabelecidas nas matrizes ordenadas as relações entre o número real de funções geradoras e:

- A utilização da validação cruzada ou de todas as amostras para a busca dos parâmetros ótimos;
- A superposição das classes em razão da distribuição das mesmas no espaço de entrada.

A análise dos resultados apontam o fato de que as matrizes de *kernel* ordenadas de conjuntos de classes com pequena ou nenhuma superposição apresentam submatrizes em bloco na diagonal podendo caracterizar bem a estrutura dos dados e o número de funções geradoras. Portanto, a multimodalidade dos conjuntos parece não afetar a identificação visual mais apropriada e próxima da estrutura verdadeira. Dados com essa característica não podem ser identificados corretamente por alguns métodos de agrupamento, o que dá alguma vantagem à metodologia apresentada neste trabalho por ser melhor neste quesito.

O método Minus mostrou-se mais efetivo na obtenção de uma permutação da ordem das amostras para revelar os grupos de dados semelhantes. No entanto, é preciso lembrar que o método altera os valores da matriz original e demanda grande número de cálculos além de muito tempo e memória. Como em alguns casos os resultados do Minus foram aproximados pelos do método do autovetor, seria possível adotar apenas esse último desde que sejam determinadas as condições sob as quais são obtidos resultados tão bons quanto os do primeiro método: tipo da função de similaridade, tipo de constituição

e normalização dos atributos, número de níveis quartis, número máximo de grupos e número mínimo de amostras. Deve-se registrar ainda a impressão inicial de que a visualização do número de grupos nas matrizes é melhor quando as amostras possuem atributos numéricos de valores contínuos e são devidamente normalizadas.

Os critérios de parada adotados nos algoritmos evolucionários foram satisfatórios na maioria dos casos assim como os parâmetros e a codificação escolhidos. Entretanto, acredita-se que a codificação do AG sem o operador de seleção causou a convergência prematura do método, justificando a inferioridade dos seus resultados frente aos alcançados pelo PSO.

As alterações implementadas nos algoritmos utilizados dinamizaram em muito a avaliação da função de alinhamento A, permitindo a sua aplicação em conjuntos muito maiores que os de [36]. Contudo, até que se alcance reduções maiores do esforço computacional e de tempo, permanecem muito custosos os experimentos nas mesmas condições estabelecidas em [23]. Foi observado durante os testes que a execução do algoritmo em computadores com mais de um núcleo pode ter reduzido o tempo total demandado para a aplicação da metodologia em algumas bases. A metodologia aplicada às bases menores de dados utilizou muito menos tempo que o requerido para os testes com as bases maiores. Tal fato não se explica apenas pelo número de cálculos diretamente dependente do número de amostras da base, mas também:

- Pela quantidade de memória física necessária;
- Pela possibilidade de o *MATLAB* alocar dinamicamente mais núcleos do processador para a realização dos processos;
- Pelo fato de que para as bases menores, o número de agrupamentos solicitado ao FCM foi baixo, sobretudo quando os dados foram normalizados.

Por esses argumentos, espera-se que o avanço e o maior acesso a recursos computacionais desse tipo permitam a aplicação de bases maiores aos métodos propostos.

5.2 Conclusões finais

Sob a perspectiva da otimização de função, a transformação e mapeamento dos dados bem como o seu agrupamento foram tratados segundo um objetivo alternativo de modo a preservar no espaço de transformação as relações existentes entre as amostras no espaço de entrada. A sustentação para tal escolha se dá pela capacidade de alinhamento das matrizes de *kernel* e de FPM quando ambas armazenam informação coerente sobre a estrutura das

amostras de um conjunto de dados. O alinhamento A, calculado de acordo com o produto interno de Frobenius segundo [11] entre a matriz de *kernel* e a FPM, é função dos parâmetros σ e c através dos quais pode ser maximizado graças ao ajuste apropriado desses últimos.

Os experimentos demonstram a potencialidade do método em revelar a informação estrutural das amostras, cujo conhecimento a priori é bastante desejado tanto no projeto de classificadores quanto na determinação de partições. Os resultados obtidos em [36] e no presente trabalho permitiram o projeto de classificadores binários do tipo LS-SVM com complexidade aceitável e bom desempenho de generalização na maioria das bases testadas. Portanto, a escolha de A como função objetivo não impede a obtenção de máquinas de aprendizados satisfatórias e ainda permite que essas preservem as relações das amostras utilizadas para o treinamento. Logo, o projeto do *kernel* pode ser obtido do alinhamento explorando as propriedades estruturais dos dados ao invés de ter como base a busca cega e exaustiva no espaço de parâmetros com o único objetivo de estabelecer o mapeamento entrada-saída. Com base nestes apontamentos, o *kernel* resultante é mais representativo dado que o seu projeto se baseia também na estrutura dos dados e também pode ser empregado como função de mapeamento. Ainda com base nesses argumentos, a modelagem dos dados seguindo essa abordagem utiliza conhecimento dos dados mas continua a obter classificadores não-lineares do tipo caixa-preta [1]: sem relação óbvia entre a estrutura e os parâmetros do modelo com as leis da natureza que regem as relações entre as amostras e as classes.

A comparação entre os classificadores obtidos e os de *benchmark* foi prejudicada. Além de os trabalhos de referência não terem fornecido todos os parâmetros necessários, não se estabeleceu neste trabalho massa equivalente de resultados que possibilitasse o mesmo tratamento estatístico para comparar os classificadores resultantes de metologias diferentes.

Algumas partições coerentes foram encontradas quando se tem o máximo alinhamento. Entretanto, não ficam definidas as condições em que a forma e o número de partições encontrados indicam seguramente o agrupamento mais apropriado dos dados. Cabe, portanto, confirmar com maior repetição dos experimentos as sugestões de que *kernels* e FPMs devam ser co-projetados ao invés de serem descritos e ajustados de forma independente. Caso positivo, confirma-se o princípio de que o aprendizado supervisionado e o não supervisionado possuem relação estreita pela interação entre *kernels* e FPMs.

Acredita-se terem sido explorados alguns dos aspectos mais relevantes que se apresentaram no caminho até aqui postas estas conclusões sobre o trabalho desenvolvido. Evidentemente, as fronteiras do problema abordado ficam cada vez mais nebulosas quanto maior é o número de conexões com as demais

áreas da Inteligência Computacional. Há ainda muito a investigar na metodologia apresentada, mas acredita-se ter dado um bom passo inicial ao obter classificadores com acurácia maior que 80% para as bases testadas.

5.3 Trabalhos futuros

Como não poderia ser diferente, ao longo da busca por atingir um objetivo, o contato com outras abordagens enriquece a visão sobre o problema e estabelece novas opções. Permanecendo o desejo por trilhar os caminhos que se apresentam a partir deste ponto, seguem algumas sugestões para o desenvolvimento de novos trabalhos. Dentre as possibilidades imediatas, destacam-se as seguintes:

- Para diminuir o tempo necessário para a realização dos experimentos, os algoritmos merecem ser codificados em linguagens mais eficientes no uso das máquinas disponíveis. A linguagem C disponível na transposição dos *scripts* em *MATLAB* é uma boa candidata para levar adiante as pesquisas nesta área. O conjunto de códigos de [40] para LS-SVM já aponta neste sentido, concentrando aos códigos em C as tarefas mais custosas de ajuste dos modelos.

- Devem ser realizadas mais validações cruzadas em para cada grupo de dados de forma a compatibilizar com os trabalhos de *benchmark* as metodologias de ajuste de parâmetros e, por consequência, os resultados e suas análises estatísticas.

- Pode-se ainda acrescentar uma etapa posterior ao ajuste dos parâmetros objetivando utilizar a matriz de *kernel* ou de proximidade ordenada resultante. A confirmação dessa última como sendo uma matriz que atende às condições de Mercer e a determinação dos vetores suporte diretamente da matriz ordenada poderiam simplificar as etapas seguintes à criação do classificador LS-SVM e SVM [8].

- Alguns dos experimentos realizados para o presente trabalho apontam resultados que merecem ser melhor investigados. A construção e posterior ordenação da matriz de *kernel* forma regiões bem definidas entre grupos de classes em alguns casos. A posterior seleção das amostras presentes na transição desses conjuntos (nas regiões periféricas das submatrizes) apontam ser esses os pontos mais próximos à margem de separação dos conjuntos. Logo, cabe verificar em que condições tal efeito ocorre e se seria possível utilizar tais amostras como vetores suporte dos classificadores.

- Utilizar o grupo de amostras de validação para ajustar o parâmetro γ da LS-SVM;
- Observar o desempenho dos classificadores quando o seu treinamento considera custos diferentes de erros de classificação e o balanceamento das classes no grupo de amostras de treinamento;
- Fixar o número de grupos se o mesmo já tiver sido pré-determinado por um método de agrupamento de consenso [39];
- Realização da varredura em grade como realizada em [23] e em [36];
- Utilizar um método de otimização multiobjetivo para obter o melhor alinhamento e a melhor acurácia ao ajustar os valores do número de grupos e do parâmetro σ do *kernel* Gaussiano. Desta forma, garante-se ao mesmo tempo o desempenho do classificador e a preservação da estrutura dos dados;
- Adaptação da metodologia para a sua utilização na criação de classificadores para dados de múltiplas classes;
- Deve-se investigar se as matrizes de *kernel* e de proximidade devem sofrer alguma transformação ou normalização antes que seja utilizado o método de ordenação por autovetor, como feito com a matriz de afinidade em [47];
- Utilizar as matrizes ajustadas de *kernel* e de proximidade em conjunto com métodos hierárquicos tal como a matriz de similaridade em [45];
- Reduzir a complexidade dos modelos dos classificadores pelos procedimentos de poda de vetores suporte utilizados em [23] e disponíveis em [40];
- Alterar a métrica de distância no método de agrupamento, analisando a variação dos resultados conforme a utilização de métricas semelhantes e diferentes no FCM e no *kernel* em função da constituição e da codificação das bases de dados.

Conforme os resultados alcançados a partir das sugestões supracitadas, pode-se ainda seguir caminhos como os de:

- Testes com outras funções de transformação [8] e métodos de agrupamento também merecem atenção por apresentarem combinações diversas e, em alguns casos, custo inferior na função de objetivo.
- Outras métricas de alinhamento de matrizes também podem ser pesquisadas e aplicadas ao problema em questão para suscitar comparações e a melhoria do processo de ajuste dos parâmetros.

- Reformulação da função de custo estabelecida pode tornar mais rápida sua avaliação e permitir a sua aplicação em bases maiores de dados. Ainda neste sentido, o estabelecimento de um valor máximo possível ou desejável para o número de agrupamentos pode dinamizar em muito a avaliação da função objetivo limitando as demandas de cálculos, memória e tempo do método de agrupamento.

Referências Bibliográficas

[1] Luis Antonio Aguirre, *Introdução à identificação de sistemas: técnicas lineares e não-lineares aplicadas a sistemas reais*, Editora UFMG, Belo Horizonte, 2004.

[2] P. Arabie and L. J. Hubert, *The bond energy algorithm revisited*, IEEE Transactions on Systems, Man and Cybernetics **20** (1990), no. 1, 268–274.

[3] A. Asuncion and D.J. Newman, *UCI machine learning repository*, 2007, http://www.ics.uci.edu/~mlearn/MLRepository.html.

[4] James C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[5] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik, *A training algorithm for optimal margin classifiers*, In: COLT '92: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, ACM Press, 1992, pp. 144–152.

[6] Weiling Cai, Songcan Chen, and Daoqiang Zhang, *Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation*, Pattern Recogn. **40** (2007), no. 3, 825–838.

[7] _____, *Robust fuzzy relational classifier incorporating the soft class labels*, Pattern Recogn. Lett. **28** (2007), no. 16, 2250–2263.

[8] Yihua Chen, Eric K. Garcia, Maya R. Gupta, Ali Rahimi, Luca Cazzanti, and Alexander J. Smola, *Similarity-based classification: Concepts and algorithms*, Journal of Machine Learning Research **10** (2009), 747–776.

[9] Corinna Cortes and Vladimir Vapnik, *Support vector network*, Machine Learning **20** (1995), 273–297.

[10] N. Cristianini and J. Shawe-Taylor, *Support vector machines and other kernel-based learning methods*, Cambridge University Press, 2000.

[11] Nello Cristianini, Jaz Kandola, Andre Elisseeff, and John Shawe-Taylor, *On kernel target alignment*, preprint (2002), available at, 2002.

[12] Programa das Nacoes Unidas para o Desenvolvimento, *Pnud*, 05 2009, http://www.pnud.org.br/idh/.

[13] Instituto Brasileiro de Geografia e Estatística, *Ibge*, 05 2009, http://www.ibge.gov.br.

[14] Governo de Minas, Fudação João Pinheiro, Minas On-line, and PRODEMGE, *Datagerais*, 05 2009, http://www.datagerais.mg.gov.br.

[15] Antonio de Padua Braga, Andre Ponce de Leon F. de Carvalho, and Teresa Bernarda Ludermir, *Redes neurais artificiais: teoria e aplicacoes*, LTC, Rio de Janeiro, 2000.

[16] _____, *Redes neurais artificiais: teoria e aplicacoes*, LTC, 2007.

[17] Instituto de Pesquisa Economica Aplicada (IPEA), *Ipeadata*, 05 2009, http://www.ipeadata.gov.br.

[18] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern classification*, Wiley-Interscience, New York, 10 2000.

[19] J. C. Dunn, *A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters*, Journal of Cybernetics **3** (1973), 32–57.

[20] Eberhart and Yuhui Shi, *Particle swarm optimization: developments, applications and resources*, Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, vol. 1, 2001, http://dx.doi.org/10.1109/CEC.2001.934374, pp. 81–86 vol. 1.

[21] Wesam Elshamy, *Particle swarm optimization simulation (code)*, 2008, http://www.mathworks.com/matlabcentral/fileexchange/11559.

[22] Mario A. T. Figueiredo and Anil K. Jain, *Unsupervised learning of finite mixture models*, IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002), no. 3.

[23] Tony Van Gestel, Johan A. K. Suykens, Bart Baesens, Stijn Viaene, Jan Vanthienen, Guido Dedene, Bart De Moor, and Joos Vandewalle, *Benchmarking least squares support vector machine classifiers*, Mach. Learn. **54** (2004), no. 1, 5–32.

[24] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Kluwer Academic Publishers, Boston, MA, 1989.

[25] Steve R. Gunn, *Support vector machines for classification and regression*, Tech. report, Image Speech and Intelligent Systems Research Group, University of Southampton, 1997.

[26] Peter E. Hart, *The condensed nearest neighbor rule*, IEEE Transactions on Information Theory (1968), 515–516.

[27] Simon Haykin, *Neural networks: A comprehensive foundation*, Prentice Hall, 1998.

[28] In: P.Brusilovsky, M. Grigoriadou, K. Papanikolaou (Eds.): proceedings of Workshop on Personalisation in E-Learning Environments at Individual and Group Level, 11th International Conference on User Modeling (UM2007), *Investigation of group formation using low complexity algorithms*, 06 2007.

[29] A. K. Jain, M. N. Murty, and P. J. Flynn, *Data clustering: A review*, ACM Computing Surveys **31** (September 1999), 264–323.

[30] Jyh-Shing Roger Jang, Chuen-Tsai Sun, and Eiji Mizutani, *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence*, Matlab Curriculum Series, Prentice Hall, 1997.

[31] J. Kennedy and R. Eberhart, *Particle swarm optimization*, Proc. of the IEEE Int. Conf. on Neural Networks (Perth, WA, Australia), vol. 4, IEEE, 11-12 1995, pp. 1942–1948.

[32] John F. Kolen and Tim Hutcheson, *Reducing the time complexity of the fuzzy c-means algorithm*, IEEE Transactions on Fuzzy Systems **10** (2002), no. 2, 263–267.

[33] V. Loia, W. Pedrycz, and S. Senatore, *Proximity fuzzy clustering for web context analysis*, Proceedings of EUSFLAT (2003), 59–62.

[34] Sueli Aparecida Mingoti, *Análise de dados através de métodos de estatística multivariada: uma abordagem aplicada*, Editora UFMG, Belo Horizonte, 2005.

[35] Fundação João Pinheiro, *Fjp*, 05 2009, http://www.fjp.gov.br.

[36] Francisco A. A. Queiroz, Antonio P. Braga, and Witold Pedrycz, *Sorted kernel matrices as cluster validity indexes*, The 13th IFSA World Congress

and the 6th Conference of EUSFLAT, International Fuzzy Systems Association - European Society for Fuzzy Logic and Technology 2009, IFSA, 07 2009.

[37] Reginaldo J. Santos, *Um curso de geometria analítica e álgebra linear*, viii ed., UFMG, Imprensa Universitaria, Belo Horizonte, 07 2009.

[38] Simpósio Brasileiro de Redes Neurais, *Estratégias neurais para treinamento de least squares support vector machines*, Anais do VIII Simpósio Brasileiro de Redes Neurais, 2004.

[39] Alexander Strehl, Joydeep Ghosh, and Claire Cardie, *Cluster ensembles - a knowledge reuse framework for combining multiple partitions*, Journal of Machine Learning Research **3** (2002).

[40] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least squares support vector machines*, World Scientific, Singapore, 2002.

[41] J. A. K. Suykens and J. Vandewalle, *Least squares support vector machine classifiers*, Neural Process. Lett. **9** (1999), no. 3, 293–300.

[42] D. Tsafrir, I. Tsafrir, L. Ein-Dor, O. Zuk, Notterman D. A., and E. Domany, *Sorting points into neighborhoods (spin): data analysis and visualization by ordering distance matrices*, Bioinformatics **21** (2005), no. 10, 2301–2308.

[43] V. N. Vapnik, *An overview of statistical learning theory*, IEEE Transactions on Neural Networks **10** (1999), no. 5, 988–999.

[44] Leo Vohandu, Rein Kuusik, Ants Torim, Eik Aab, and Grete Lind, *Some algorithms for data table (re)ordering using monotone systems*, AIKED'06: Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, World Scientific and Engineering Academy and Society (WSEAS), 2006, pp. 417–422.

[45] Jun Wang, Bei Yu, and Les Gasser, *Classification visualization with shaded similarity matrix*, Tech. report, GSLIS University of Illinois at Urbana-Champaign, 2002.

[46] Ryosuke LA Watanabe, Enrique Morett, and Edgar E. Vallejo, *Inferring modules of functionally interacting proteins using the bond energy algorithm*, BMC Bioinformatics **9** (2008), no. 285.

[47] Yair Weiss, *Segmentation using eigenvectors: a unifying view*, Proceedings IEEE International Conference on Computer Vision (1999), 975–982.

[48] Jieping Ye and Tao Xiong, *Svm versus least squares svm*, The Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 2007) (2007), 640–647.

[49] Guoqiang P. Zhang, *Neural networks for classification: a survey*, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **30** (2000), no. 4, 451–462, http://dx.doi.org/10.1109/5326.897072.