UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Paulo Roberto do Carmo Faustino

Avaliação de medidas de similaridade de matrizes *kernel* aplicadas em classificadores de larga margem para seleção de modelos

Juiz de Fora
2019

Paulo Roberto do Carmo Faustino

Avaliação de medidas de similaridade de matrizes *kernel* aplicadas em classificadores de larga margem para seleção de modelos

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Saulo Moraes Villela

Coorientador: Carlos Cristiano Hasenclever Borges

Juiz de Fora

2019

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF com os dados fornecidos pelo(a) autor(a)

do Carmo Faustino, Paulo Roberto.

Avaliação de medidas de similaridade de matrizes *kernel* aplicadas em classificadores de larga margem para seleção de modelos / Paulo Roberto do Carmo Faustino. – 2019.

58 f. : il.

Orientador: Saulo Moraes Villela

Coorientador: Carlos Cristiano Hasenclever Borges

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação, 2019.

1. Seleção de modelos. 2. Seleção de hiperparâmetros. 3. Seleção de características. 4. Medidas de similaridade. 5. Classificadores de larga margem. I. Villela, Saulo Moraes, orient. II. Borges, Carlos Cristiano Hasenclever, coorient. III. Título.

Paulo Roberto do Carmo Faustino

Avaliação de medidas de similaridade de matrizes *kernel* aplicadas em classificadores de larga margem para seleção de modelos

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovada em 20 de março de 2019.

BANCA EXAMINADORA

Prof. D.Sc. Saulo Moraes Villela - Orientador
Universidade Federal de Juiz de Fora

Prof. D.Sc. Carlos Cristiano Hasenclever Borges - Coorientador
Universidade Federal de Juiz de Fora

Prof. D.Sc. Raul Fonseca Neto
Universidade Federal de Juiz de Fora

Prof. D.Sc. Vinicius Layter Xavier
Universidade do Estado do Rio de Janeiro

Aos meus pais pelo apoio incondicional.

AGRADECIMENTOS

Agradeço aos meus pais Ilza e José Roberto, pelo amor, incentivo, e apoio incondicional.

Aos professores Saulo e Carlos Cristiano, pela paciência, tutoria e comprometimento durante toda a orientação.

À minha irmã Ana Paula, pelo incentivo nas horas difíceis, de desânimo e cansaço.

Aos meus amigos, que sempre tiveram uma palavra de conforto nos momentos de ansiedade.

À Universidade Federal de Juiz de Fora, pelo apoio técnico e suporte durante esses anos de pesquisa.

E agradeço ao meu pequeno sobrinho João Pedro, por simplesmente existir, chegando em uma época difícil, iluminando a minha vida e diminuindo meus pesares no fim desta jornada.

"Julgue seu sucesso pelas coisas
que você teve que renunciar
para conseguir"
Dalai Lama

RESUMO

A proposta deste trabalho é investigar o comportamento de medidas de similaridade, como *Kernel Target Alignment* (KTA) e *Feature Space-based Kernel Matrix Evaluation Measure* (FSM), e observar suas interações com um classificador de larga margem, construir um modelo de seleção de modelos, implementando seus componentes separadamente: um modelo de seleção de hiperparâmetros e um modelo de seleção de características. Os métodos KTA e FSM indicam o grau de similaridades entre matrizes *kernel*, retornando um valor de alinhamento. Este alinhamento é utilizando na construção dos modelos de seleção utilizando o método *Simulated Annealing*. São apresentados testes iniciais indicando o desempenho das medidas de similaridade, para a escolha adequada de qual medida será acoplada ao modelo de seleção proposto. Em seguida são descritos, separadamente, os modelos propostos de seleção, bem como seus resultados comparativos.

Palavras-chave: Seleção de modelos. Seleção de hiperparâmetros. Seleção de características. Medidas de similaridade. Classificadores de larga margem.

ABSTRACT

The purpose of this work is to investigate the behavior of similarity measurements, i.e., Kernel Target Alignment (KTA) and Feature Space-based Kernel Matrix Evaluation Measure (FSM) in relation to their correlation with a large margin classifier - support vector machine, in order to propose and implement a model selection method, constructed by means of two steps: a hyper-parameter selection model and a model for feature selection. The KTA and FSM methods indicate the degree of similarity between kernel matrices determined by an alignment measure. This value of alignment is used as reference for a wrapper model selection construction using the simulated anneling as optimizer. Initial tests are depicted to verify the similarity measurements performance in relation to a large margin classifier aiming to identify the better measure to be adopted in the proposed selection model. Following, the described selection model components are tested separately and their results are exhaustively analyzed.

Key-words: Model selection. Hyper-parameter selection. Feature selection. Similarity measures. Large margin classifiers.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

FS	*Feature Space*
FSM	*Feature Space-based Kernel Matrix Evaluation Measure*
KTA	*Kernel Target Alignment*
MS	Medida de Similaridade
NP	Nível de Predição
PUK	*Pearson VII Universal Kernel*
RBF	*Radial Basis Function*
ROC	*Receiver Operating Characteristic*
SA	*Simulated Annealing*
SVM	*Support Vector Machine*

SUMÁRIO

1 INTRODUÇÃO

Métodos *kernel* têm atraído grande interesse por se mostrarem eficientes em classificação não linear. Baseiam-se no uso de funções *kernel* que mapeiam dados não linearmente separáveis para um espaço de dimensão mais alta, ou até mesmo infinita, onde os dados se tornam linearmente separáveis (YOU, 2011). Seleção de modelos tendo como base métodos *kernel* têm se apresentado como uma estratégia interessante em pesquisas da área de aprendizado de máquina. Seleção de modelos é o processo de busca do método mais adequado entre diferentes abordagens de aprendizado de máquina ou de ajuste de diferentes (hiper)parâmetros associados ao modelo utilizado em questão. Considera-se também, como um procedimento de seleção de modelos, a determinação de conjuntos de características mais adequados para uma determinada abordagem.

O desempenho de um classificador, especificamente de um classificador de larga margem, como as Máquinas de Vetores Suporte (*Support Vector Machines* – SVMs), depende diretamente da escolha adequada dos parâmetros. Cada conjunto de parâmetros define um modelo que pode ser aplicado na resolução de um determinado problema de interesse sendo, portanto, um caso específico de seleção de modelos (YOU, 2011). O conhecimento prévio dos dados é fator crucial para a seleção de parâmetros, visto que influencia consideravelmente o ajuste a ser obtido.

Outra questão de bastante relevância em aprendizado de máquina a ser considerada é referente à seleção de características (atributos). Na década de setenta, época em que alguns trabalhos sobre seleção de características foram publicados, bases de dados utilizavam pouco mais de quarenta características (GUYON; ELISSEEFF, 2003). Com o avanço em procedimentos de aquisição de dados, hoje pode-se encontrar bases de dados compostas por centenas de milhares de características. Esta nova perspectiva traz a necessidade de um melhor entendimento dos dados visando identificar situações de irrelevância e redundância das características que compõem a base.

Entre os benefícios da seleção de características, pode-se destacar um melhor entendimento dos dados de maneira geral, uma maior facilidade na visualização, a possibilidade de redução de dimensionalidade e, com isso, a diminuição dos requisitos de armazenamento, com possível melhora no desempenho da previsão, entre outras vantagens (GUYON; ELISSEEFF, 2003).

Apesar de ser extremamente relevante, um processo de seleção de modelos que incorpore uma seleção de parâmetros com uma seleção de características pode requisitar um alto custo computacional, podendo-se tornar inclusive proibitivo, visto que envolve procedimentos de otimização que geralmente solicitam a construção sistemática de modelos de predição para o direcionamento da busca. Soluções eficientes, com menor demanda computacional, são de grande interesse para a real efetivação de seleção de modelos.

Neste trabalho são explorados métodos de medidas entre matrizes *kernel* que têm como objetivo a avaliação do nível de similaridade entre matrizes distintas, assim como entre uma matriz *kernel* e uma matriz de rótulos construída de forma adequada. Esses processos, conhecidos como medidas de similaridade ou alinhamento, possuem um custo computacional consideravelmente baixo, com complexidade $O(n^2)$. Sendo assim, formulações de medição de similaridade se apresentam como uma estratégia de interesse em procedimentos de avaliação de modelos. Desta forma, o objetivo, então, deste trabalho é direcionado para o desenvolvimento de um modelo de seleção tendo como base de avaliação medidas de similaridade.

Inicialmente, é importante uma completa avaliação dos potencial e eficiência das principais medidas de similaridade que se apresentam na literatura desde o trabalho seminal de Cristianini et al. (2002), onde foi introduzida a primeira medida alinhamento, conhecida como *Kernel Target Alignment* (KTA). Esta parte do trabalho é de extrema importância para o entendimento do desempenho das principais medidas de similaridade em situações distintas, respaldando, assim a escolha da mais adequada para o modelo a ser desenvolvido.

Em relação à estratégia de seleção de modelos, o uso de medidas de similaridade viabiliza uma construção do método em fases desacopladas, com a expectativa de desempenho robusto em cada uma das fases. Desta forma, a seleção de modelos é implementada em duas etapas sequenciais: (i) seleção de hiperparâmetros e (ii) seleção de características. Estratégia incomum na literatura, o desacoplamento permite uma simplificação na busca do ajuste de cada etapa, além de fornecer uma referência construtiva para a segunda etapa, utilizando-se do resultado da primeira previamente executada.

A primeira etapa do método proposto tem como objetivo a construção de um modelo de seleção de hiperparâmetros por meio dos alinhamentos, proveniente das medidas de similaridade entre as matrizes *kernel* e uma matriz de rótulos associada da base em questão, para direcionar a seleção do conjunto de parâmetros ótimos de uma função *kernel*. Em sequência, a segunda etapa tem como proposta selecionar características utilizando-se dos valores dos alinhamentos gerados na primeira etapa como referência construtiva para o método de busca, o que, de certa maneira, permite mensurar o impacto das características selecionadas no processo de busca em relação à medida de similaridade adotada. Assim, tem-se como objetivo aumentar a efetividade da classificação, considerando a utilização de métodos baseados em *kernel* no processo de aprendizagem.

Métodos de seleção de modelos, englobando parâmetros e características, são geralmente encontrados na literatura como uma única representação, ou única etapa. Neste trabalho, a seleção de modelos foi separada em duas etapas, seleção de hiperparâmetros e seleção de características, proporcionando uma análise minuciosa em relação ao comportamento de cada uma das etapas.

Finalizando, no contexto de seleção de modelos, é realizado um estudo sobre métodos de alinhamento entre matrizes *kernel* neste trabalho. Porém, também é possível observar outras abordagens para este tipo de seleção que são baseados em *kernel* com propósitos distintos, entre eles a diminuição do tempo computacional ou a otimização na busca de resultados. Em seleção de características, métodos baseados em *kernel* também são utilizados alternativamente, como o trabalho proposto por Kuo et al. (2013), que realiza uma seleção de características baseada em matriz *kernel* para a classificação de imagens.

1.1 ESTRUTURA DO TEXTO

Nos Capítulos 2 e 3 são descritas definições sobre os componentes principais deste trabalho, bem como o funcionamento do SVM, funções *kernel* e definições sobre os métodos de similaridade utilizados na construção do modelo de seleção, assim como suas variantes.

No Capítulo 4 são exibidos os passos de construção do modelo proposto, seu funcionamento e sua interação com os modelos citados nos capítulos anteriores.

O Capítulo 5 descreve os passos iniciais para os testes das medidas de similaridade, representando os resultados em gráficos e tabelas. Além disso, são apresentados, também, os resultados obtidos com o modelo de seleção proposto.

No Capítulo 6 são apresentadas as considerações finais sobre o trabalho e sugestões para trabalhos futuros.

2 CLASSIFICADORES DE LARGA MARGEM

Um classificador de larga margem é uma método de aprendizado supervisionado linear para classificação binária através da construção de um hiperplano separador.

Dado um conjunto de treinamento $Z = \{(x_i, y_i) : i \in \{1, 2, \ldots, n\}\}$, onde $x_i \in \mathbb{R}^d$ representa um vetor com d características ou atributos associado a um rótulo binário $y_i \in \{-1, 1\}$, que define sua classe de pertinência. Considere a existência de uma distribuição de probabilidade $P(x, y)$, não conhecida *apriori*, da qual os dados foram obtidos.

A hipótese indutiva obtida pelo classificador consiste em um processo de aprendizado, utilizando o conjunto (x_i, y_i), que resulte em um mapeamento $x \mapsto y$, de forma que o classificador realize a classificação de um conjunto (x, y) desconhecido, baseando-se no conceito de generalização associada à distribuição de probabilidade representada pelo conjunto de dados no processo de aprendizagem.

2.1 FUNÇÕES *KERNEL*

O conjunto de todas as variáveis, geradas a partir do espaço de entrada, exceto pelas variáveis correspondentes ao conjunto esperado de respostas, é representado pelo termo conhecido como *feature space*, ou espaço de características. Um espaço de características é definido de acordo com a adequada escolha da função *kernel*.

Uma função de mapeamento ϕ é usualmente aplicada em pares de dados visando a obtenção dos termos da chamada matriz *kernel* através do produto escalar das imagens de seus argumentos:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j). \tag{2.1}$$

Diversas funções *kernel* se apresentam na literatura para o mapeamento do produto escalar dos dados no espaço de características. Como exemplo, apresenta-se três das funções mais conhecidas:

$$\begin{aligned} \text{Linear:} \quad & K(x_i, x_j) = \langle x_i, x_j \rangle + c \\ \text{Polinomial:} \quad & K(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^p \\ \text{Gaussiano:} \quad & K(x_i, x_j) = exp\left(-\gamma \left\| x_i - x_j \right\|^2\right), \end{aligned} \tag{2.2}$$

onde c, p e γ são os parâmetros das funções linear, polinomial e gaussiana, representando uma constante de linearidade, o grau do polinômio e a largura da gaussiana, respectivamente.

O objetivo no uso de funções *kernel* é viabilizar que os dados mapeados no espaço de características sejam linearmente separáveis, de forma que o discriminante linear seja não só viável como efetivo. Porém, a escolha não só dos parâmetros bem como da própria função de mapeamento geralmente não é trivial. Assim, ao invés das funções *kernel*

clássicas apresentadas, investiga-se neste trabalho uma função *kernel*, conhecida como uma função universal, que apresenta potencial para o mapeamento do *kernel* linear ao gaussiano, de acordo com o ajuste de seus parâmetros. Esta função é descrita na subseção a seguir.

2.1.1 *Pearson VII Universal Kernel*

Devido ao padrão da distribuição dos tipos de dados serem, geralmente, desconhecidos, a busca por uma função de mapeamento para que se tenha separabilidade linear no espaço de características é feita de maneira experimental, através de procedimentos de tentativa erro, aplicando e validando diversas funções *kernel*, com parâmetros distintos, na busca por um maior nível de separabilidade. Portanto, o processo de seleção do tipo de *kernel* a ser utilizado em conjunto com a seleção dos parâmetros dos mesmos, é um procedimento ineficiente, propenso a imprecisões, além de demandar um alto custo computacional (ÜSTÜN et al., 2006).

Com essa observação sobre a dificuldade de se ter um conhecimento prévio sobre a natureza do mapeamento dos dados no espaço de características, tem-se, portanto, um problema complexo de escolha de funções *kernel* e seus respectivos parâmetros, que devem ser avaliados individualmente. Desta forma, a proposta da utilização da função *Pearson VII Universal Kernel* (PUK) se apresenta como uma alternativa viável pelo seu alto nível de adaptabilidade controlado, basicamente, por seus parâmetros.

A função PUK, descrita por Üstün et al. (2006), tem com principal indicativo a capacidade de ser uma função adaptável de acordo com o ajustes de seus dois parâmetros, ω e σ. Essa propriedade descrita pelos autores, possibilita que a função seja utilizada como um *kernel* genérico que possa ser modelado simulando uma família de *kernels*, entre os quais o *kernel* Linear, Polinomial e Gaussiano (Radial Basis Function – RBF).

A fórmula geral da função PUK é dada da seguinte forma:

$$K(x_i, x_j) = \frac{1}{\left[1 + \left(\frac{2\sqrt{\|x_i - x_j\|^2}\sqrt{2^{\frac{1}{\omega}} - 1}}{\sigma}\right)^2\right]^{\omega}}, \tag{2.3}$$

onde σ e ω correspondem aos parâmetros ajustáveis na função, que controlam a largura e o fator cauda do pico. Por não ter seu uso tão disseminado, é importante avaliar se o *kernel* PUK realmente apresenta, de forma eficiente e prática, o potencial de adaptabilidade de mapeamento indicado, tarefa de interesse deste trabalho.

2.2 MÁQUINAS DE VETORES SUPORTE

Dentre os diversos classificadores de larga margem existentes na literatura, as máquinas de vetores suporte (*Support Vector Machines* – SVMs) destacam-se por ser um

dos métodos mais utilizados devido a seu alto desempenho em tarefas de aprendizado supervisionado (MARSLAND, 2014).

O SVM foi desenvolvido tanto para a resolução de problemas de classificação, como também foi adaptado para a resolução de problemas de regressão (WANG, 2005). O treinamento de um SVM visando a determinação do hiperplano ótimo de separação é realizado através da resolução de um problema de otimização quadrática com restrições.

O SVM foi primeiramente proposto por Boser et al. (1992), onde o autor identifica um bom desempenho na generalização do classificador obtido apenas quando a capacidade da função de classificação é compatível com o tamanho do conjunto de treinamento.

A estratégia básica do SVM é a definição de uma estratégia ótima de separabilidade das classes pelo hiperplano, baseando-se em um conceito de margem diretamente associada as classes. Esse hiperplano é obtido através de um problema de otimização.

Considerando $Z^+ = \{(x_i, y_i) \in Z : y_i = +1\}$ e $Z^- = \{(x_i, y_i) \in Z : y_i = -1\}$, os dados referentes a cada uma das classes, o SVM identifica um hiperplano, referenciado pelo seu vetor normal $w \in \mathbb{R}^d$ e também pela constante $b \in \mathbb{R}$, que resulta na separação ótima dos conjuntos Z^+ e Z^-, sendo a margem determinada pela distância mínima $\gamma \geq 0$ entre os conjuntos Z^+ e Z^- e o hiperplano separador. Assim, o problema resume-se a encontrar (w, b) de forma que seja obedecida a inequação:

$$y_i \left(\langle w, x_i \rangle + b \right) \geq \gamma, \quad \forall \left(x_i, y_i \right) \in Z, \tag{2.4}$$

com a máxima margem possível. Observa-se que a maximização da margem em relação à $\langle w, x \rangle + b = 0$ pode ser obtida através da minimização de $\|w\|$, recorrendo-se então ao seguinte problema de otimização, para todo $i \in \{1, 2, \ldots, n\}$:

$$\begin{array}{ll} \underset{w,b,\xi}{\text{minimizar}} & \frac{1}{2} w^T w + C \sum_i \xi_i \\ \text{com as seguintes restrições} & y_i \left(w^T \phi \left(x_i \right) + b \right) \geq 1 - \xi_i, \ \xi_i \geq 0, \end{array} \tag{2.5}$$

sendo $\phi(.)$ uma função de mapeamento, caso seja aplicada, ξ_i as variáveis de folga e C é uma constante de penalização.

Trata-se de um problema de otimização convexa, ou seja, sem ótimos locais na função objetivo a ser minimizada, com os pontos que são atendidos pelas restrições formando, também, um conjunto convexo. Assim, torna-se viável a introdução de uma função de Lagrange com os multiplicadores de Lagrange associados a viabilidade das restrições. A função Lagrangiana é dada da seguinte forma:

$$L\left(w, b, \alpha \right) = \frac{1}{2} w^T w - \sum_i \alpha_i y_i \left(\langle w, x_i \rangle + b \right) + \sum_i \alpha_i. \tag{2.6}$$

A função Lagrangiana deve ser otimizada em relação à w, b e α, sujeito a $\alpha_i \geq 0$ para todo $i \in \{1, 2, \ldots, n\}$.

Substituindo os parâmetros w e b, a solução é obtida pela maximização das funções duais, sendo assim factível a obtenção da formulação do SVM em função das variáveis α. Sua formulação é dada da seguinte forma:

$$\begin{array}{ll} \text{maximizar} & L(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\ \text{com as seguintes restrições} & \begin{cases} \sum_i \alpha_i y_i = 0, \\ \alpha \geq 0. \end{cases} \end{array} \tag{2.7}$$

Esta formulação, conhecida como dual, tem com principal vantagem a facilidade na introdução de funções de mapeamento *kernel* de forma eficiente, visto que aparece diretamente na função objetivo a ser maximizada o produto interno associado à construção das matrizes *kernel*.

3 MEDIDAS DE SIMILARIDADE PARA MATRIZES *KERNEL*

Entre os processos relacionados a execução de uma função *kernel*, é relevante a avaliação da criação de matrizes *kernel* $\langle K_1, K_2 \rangle$ distintas para realização do mapeamento pelo classificador. As matrizes *kernel* mapeiam as informações dos dados baseando-se no produto interno dos mesmos.

No contexto de matrizes *kernel*, um método de medição de similaridade deve retornar o grau de concordância entre duas matrizes geradas por diferentes funções *kernel*, indicando o quanto elas são similares. Esta mesma medição, também pode ser aplicada entre a matriz *kernel* e uma matriz gerada a partir do vetor de rótulos y. A seguir, apresentam-se algumas medidas de alinhamento e similaridade mais conhecidas e utilizadas da literatura.

3.1 *KERNEL TARGET ALIGNMENT*

O método *Kernel Target Alignment* (KTA) é baseado no cálculo de um grau de similaridade entre diferentes mapeamentos gerados por conjuntos de dados considerando a aplicação de *kernels* distintos (CRISTIANINI et al., 2002). O método determina a criação de um quantificador, que corresponde ao grau de similaridade entre as matrizes *kernel* ou entre uma matriz *kernel* e um conjunto de rótulos, identificando o nível de semelhança entre duas funções *kernel*, como também o nível de semelhança entre o mapeamento gerado por uma matriz *kernel* e um conjunto de rótulos associados.

O KTA é baseado no produto interno de Frobenius entre duas matrizes, também conhecido como produto interno euclidiano (MOAKHER, 2005). O produto interno de Frobenius entre duas matrizes M e N é definido como:

$$\langle M, N \rangle_F = \sum_{i,j} m_{i,j} n_{i,j} = Tr(M, N), \tag{3.1}$$

onde Tr corresponde ao traço da matriz.

Dadas duas matrizes *kernel* geradas por mapeamentos distintos $K_1 : X^2 \to \mathbb{R}^{n \times n}$ e $K_2 : X^2 \to \mathbb{R}^{n \times n}$, o alinhamento conhecido como KTA é definido da seguinte forma:

$$A(K_1, K_2) = \frac{\langle K_1, K_2 \rangle_F}{\sqrt{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F}}. \tag{3.2}$$

O alinhamento pode ser medido também entre uma matriz K e um conjunto de rótulos y. Considerando uma matriz $K : X^2 \to \mathbb{R}^{n \times n}$ e um vetor de rótulos $y \in \{-1, +1\}^n$, o alinhamento é definido na forma:

$$A(K, y\,y^T) = \frac{\left\langle K, y\,y^T \right\rangle_F}{\sqrt{\langle K, K \rangle_F \langle y\,y^T, y\,y^T \rangle_F}} \tag{3.3}$$

Adotando $\hat{K} = y\,y^T$ e uma vez que $\left\langle y\,y^T, y\,y^T \right\rangle_F = n^2$, o alinhamento pode ser representado pela seguinte equação:

$$A(K, \hat{K}) = \frac{\left\langle K, \hat{K} \right\rangle_F}{n\sqrt{\langle K, K \rangle_F}}. \tag{3.4}$$

3.1.1 Matriz *Kernel* Centralizada

A ideia de trabalhar com um alinhamento em matrizes centralizadas, apresentada por Cortes et al. (2012), é similar ao alinhamento KTA proposto por Cristianini et al. (2002).

Sejam os dados $x \in X$ de uma base específica associados a uma determinada função de distribuição. Uma função *kernel* permite o mapeamento dos dados de entrada no espaço de características $\phi : X \to \mathbb{H}$, com $\mathbb{H}$ sendo um espaço de Hilbert de reprodução *(reproducing kernel Hilbert space – RKHS)*. Para a obtenção de uma matriz *kernel* centralizada os dados mapeados devem ser centralizados através da diferença entre seu valor no espaço de características e valor esperado do mapeamento: $\phi - E_x[\phi]$, onde E_x representa o valor esperado de ϕ quando x é estimado de acordo com a distribuição associada. Desta forma, centralizar uma matriz *kernel* $K : X \times X \to \mathbb{R}^{n \times n}$, consiste em centralizar cada uma das características dos dados. Logo, a matriz centralizada K_c relativa a matriz *kernel* K é definida para cada um dos pares de dados x, $x' \in X$ de acordo com:

$$\begin{aligned} K_c(x, x') &= (\phi(x) - E_x[\phi])^T(\phi(x') - E_{x'}[\phi]) \\ &= K(x, x') - E_x[K(x, x')] - E_{x'}[K(x, x')] + E_{x,x'}[K(x, x')]. \end{aligned} \tag{3.5}$$

A extensão do processo de centralização para todos os dados $x \in X$ é dada por (CORTES et al., 2012):

$$K_{c_{ij}} = K_{ij} - \frac{1}{n}\sum_{i=1}^{n} K_{ij} - \frac{1}{n}\sum_{j=1}^{n} K_{ij} - \frac{1}{n^2}\sum_{i,j=1}^{n} K_{ij}, \tag{3.6}$$

ou, em notação matricial:

$$K_c = \left[I - \frac{U}{n}\right] K \left[I - \frac{U}{n}\right], \tag{3.7}$$

com I sendo a matriz identidade e U uma matriz quadrada com todos elementos iguais a unidade.

Feita a centralização da matriz K, o alinhamento entre a matriz *kernel* centralizada e a matriz de rótulos é obtido da seguinte forma:

$$A_c(K, K') = \frac{\left\langle K_c, \hat{K} \right\rangle_F}{\|K_c\|_F \left\|\hat{K}\right\|_F}. \tag{3.8}$$

3.1.2 Matriz *Kernel* Balanceada

Em bases de dados desbalanceadas, visando uma maior eficiência no cálculo do alinhamento, Ramona et al. (2012) descrevem uma forma de balanceamento através do ajuste da matriz construída com os rótulos. Seja n_+ o número de instâncias de uma das classes e n_- a quantidade de instâncias da outra classe, com $n_+ \neq n_-$ e $n = n_+ + n_-$ o número total de instâncias. Neste caso, a matriz de rótulos $\hat{K} = y\,y^T$ é substituída visando compensar o desbalanceamento entre as classes pela matriz:

$$\hat{K}_b = y_b\, y_b^T, \tag{3.9}$$

com $y_{b_i} \in \left\{\frac{1}{n_+}, \frac{-1}{n_-}\right\}$, que gera a nova matriz de rótulos com o seguinte padrão:

$$\hat{K}_b = \frac{1}{n_+\, n_-} = \begin{pmatrix} \frac{n_-}{n_+}U & -U \\ -U & \frac{n_+}{n_-}U \end{pmatrix}, \tag{3.10}$$

novamente, U indica uma matriz com todos elementos iguais a unidade. Neste caso, tem-se $\left\|\hat{K}_b\right\|_F = \frac{n}{n_+\, n_-}$.

Ressalta-se que esta estratégia de alinhamento pode ser estendida para problemas multiclasse, ou seja, com mais de duas classes, através de uma nova definição do *kernel* do rótulo como:

$$\hat{K}_{nc}(x_i, x_j) = \begin{cases} 1, & \text{caso } y_i = y_j \\ \frac{-1}{nc-1}, & \text{caso } y_i \neq y_j \end{cases}, \tag{3.11}$$

onde nc é o número total de classes. A matriz *kernel* é então substituída pela matriz de rótulos $\hat{K}_{nc}$. Considerando como exemplo $nc = 3$, tem-se:

$$\hat{K}_3 = \begin{pmatrix} U & -\frac{1}{2}U & -\frac{1}{2}U \\ -\frac{1}{2}U & U & -\frac{1}{2}U \\ -\frac{1}{2}U & -\frac{1}{2}U & U \end{pmatrix}. \tag{3.12}$$

Neste trabalho o caso multiclasse não será abordado. Porém, tem-se a expectativa que o ajuste da matriz de rótulos $\hat{K}_b$, no caso de bases desbalanceadas, possa trazer um maior nível de precisão no cálculo do alinhamento.

3.2 *FEATURE SPACE-BASED KERNEL MATRIX*

Uma medida de avaliação de matrizes *kernel* foi proposta por Nguyen e Ho (2007), sendo denominada *Feature Space-based Kernel Matrix* (FSM), com o objetivo de superar certas limitações do KTA, de forma a computar eficientemente a relação entre a matriz *kernel* e os rótulos associados. Esta técnica não pode ser caracterizada como uma medida de alinhamento, visto que compara diretamente a matriz *kernel* com a matriz de rótulos. Em seu processo construtivo, são levados em considerações dois componentes: a variância intraclasses e as posições relativas dos centros das classes.

A determinação dos centros das classes é obtida pela média dos dados de cada classe no espaço de características, sendo calculados como:

$$\phi_+ = \frac{\sum_{i=1}^{n_+} \phi(x_i)}{n_+} \quad \text{e} \quad \phi_- = \frac{\sum_{i=n_++1}^{n} \phi(x_i)}{n_-}. \tag{3.13}$$

A medida FSM é definida como a relação entre a variância total (intraclasses) na direção dos centros das classes e a distância entre os centros da classes:

$$FSM(K, y) := \frac{var}{\|\phi_- - \phi_+\|}. \tag{3.14}$$

Para um melhor entendimento do funcionamento da medida, denote $e = \frac{\phi_- - \phi_+}{\|\phi_- - \phi_+\|}$ como um vetor unitário na direção entre os centros. Assim, a variância total intraclasses na direção deste vetor unitário é calculada da seguinte forma:

$$var = [\frac{\sum_{i=1}^{n_+} \langle \phi(x_i) - \phi_+, e \rangle^2}{n_+ - 1}]^{\frac{1}{2}} + [\frac{\sum_{i=n_++1}^{n} \langle \phi(x_i) - \phi_-, e \rangle^2}{n_- - 1}]^{\frac{1}{2}}. \tag{3.15}$$

Denotando o primeiro termo da equação anterior como var_+, tem-se:

$$(n_+ - 1)var_+^2 = \sum_{i=1}^{n_+} \langle \phi(x_i) - \phi_+, e \rangle^2 = \frac{\sum_{i=1}^{n_+} \langle \phi(x_i) - \phi_+, \phi_- - \phi_+ \rangle^2}{(\phi_- - \phi_+)^2} =$$

$$\frac{\sum_{i=1}^{n_+} \left(\phi(x_i)\phi_- + \phi_+^2 - \phi(x_i)\phi_+ - \phi_+\phi_- \right)^2}{(\phi_- - \phi_+)^2}. \tag{3.16}$$

Substituindo $\phi_+ = \frac{\sum_{j=1}^{n_+} \phi(x_j)}{n_+}$ e $\phi_- = \frac{\sum_{j=n_++1}^{n} \phi(x_j)}{n_-}$ na equação anterior, e definindo algumas variáveis auxiliares, para $i = 1 \ldots n_+$:

- $a_i = \phi(x_i)\phi_+ = \frac{\sum_{j=1}^{n_+} \phi(x_i)\phi(x_j)}{n_+} = \frac{\sum_{j=1}^{n_+} k_{ij}}{n_+}$,
- $b_i = \phi(x_i)\phi_- = \frac{\sum_{j=n_++1}^{n} \phi(x_i)\phi(x_j)}{n_-} = \frac{\sum_{j=n_++1}^{n} k_{ij}}{n_-}$.

Para $i = n_+ + 1 \ldots n$:

- $c_i = \phi(x_i)\phi_+ = \frac{\sum_{j=1}^{n_+} \phi(x_i)\phi(x_j)}{n_+} = \frac{\sum_{j=1}^{n_+} k_{ij}}{n_+}$,
- $d_i = \phi(x_i)\phi_- = \frac{\sum_{j=n_++1}^{n} \phi(x_i)\phi(x_j)}{n_-} = \frac{\sum_{j=n_++1}^{n} k_{ij}}{n_-}$.

As variáveis auxiliares a_i, b_i, c_i e d_i, determinadas por Nguyen e Ho (2007), visam simplesmente facilitar a construção final da medida. A seguir, define-se as variáveis:

$A = \frac{\sum_{i=1}^{n_+} a_i}{n_+}$, $B = \frac{\sum_{i=1}^{n_+} b_i}{n_+}$, $C = \frac{\sum_{i=n_++1}^{n} c_i}{n_-}$ e $D = \frac{\sum_{i=n_++1}^{n} d_i}{n_-}$.

Logo, $A = \phi_+\phi_+$, $B = C = \phi_+\phi_-$ e $D = \phi_-\phi_-$.

Sendo assim: $(\phi_- - \phi_+)^2 = A + D - B - C$. Substituindo os termos na Equação (3.16), tem-se:

$$(n_+ - 1)var_+^2 = \frac{\sum_{i=1}^{n_+}(b_i - a_i + A + B)^2}{A + D - B - C}. \tag{3.17}$$

De maneira similar, pode-se usar um cálculo semelhante para o segundo termo na Equação (3.15), obtendo:

$$(n_- - 1)var_-^2 = \frac{\sum_{i=n_++1}^{n}(c_i - d_i + D + C)^2}{A + D - B - C}. \tag{3.18}$$

Desta forma, o cálculo da medida FSM é definido da seguinte maneira:

$$FSM(K, y) = \frac{var_+ + var_-}{\sqrt{A + D - B - C}}. \tag{3.19}$$

De acordo com Nguyen e Ho (2007), a medida FSM apresenta um maior potencial para a avaliação da qualidade de um determinado *kernel* devido a propriedade invariantes em relação à operações lineares como, por exemplo, translação e rotação, tornando-a mais similar a um discriminante linear no que tange a pertinência dos dados mapeados no espaço de características em relação aos rótulos do que o alinhamento KTA. Um dos objetivos do presente trabalho é justamente avaliar as técnicas de similaridade e alinhamento descritas visando um maior entendimento de seus comportamentos e desempenhos.

4 SELEÇÃO DE MODELOS UTILIZANDO MEDIDAS DE ALINHAMENTO

Em se tratando de busca por parâmetros ótimos, é de conhecimento que existem vários métodos que executam esta tarefa, como, por exemplo, a busca em grade (*grid search*). A busca em grade é um procedimento automático de varredura dos parâmetros do modelo visando determinar os valores mais eficientes. Apesar de crucial para o desempenho do classificador, sua aplicação pode demandar alto custo computacional, visto que a implementação é realizada através de uma busca exaustiva em relação a uma grade predefinida (BONESSO, 2013). Alternativas à busca em grade se apresentam, principalmente visando diminuir a demanda computacional necessária. Entre as principais alternativas estão os algoritmos gulosos ou míopes, que são altamente sujeitos a obtenção de ótimos locais como solução e a utilização de procedimentos de otimização global, geralmente baseados em meta-heurísticas como algoritmos evolutivos (REN; BAI, 2010), resfriamento simulado (ARAUJO, 2001), entre outros.

Neste capítulo são apresentadas estratégias baseadas em medidas de alinhamentos, previamente descritas, objetivando o desenvolvimento de:

- Uma estratégia de seleção de hiperparâmetros;
- Uma estratégia de seleção de características.

Máquinas baseadas em *kernel* são métodos bastante eficientes como etapa construtiva de ferramentas de classificação e regressão para aprendizado supervisionado e não supervisionado. Viabilizam, através do mapeamento dos dados no espaço de características, a utilização de métodos lineares de classificação, como o SVM, de forma eficiente, mesmo quando se trata de bases não-linearmente separáveis (MÜLLER et al., 2001).

Desta forma, a avaliação do mapeamento realizado pela matriz *kernel* para classificação ou regressão é crucial. Porém, para que seja efetivo, é necessário que a escolha do *kernel* a ser aplicado seja determinado de maneira eficiente em relação ao mapeamento adequado no espaço de características. Tal tarefa, na prática, é bastante custosa, visto ser necessária a construção de inúmeros classificadores para avaliar, usualmente por meio de técnica de validação cruzada, o melhor *kernel* e a melhor parametrização para a base de dados analisada. Deve-se ressaltar que uma busca exaustiva da parametrização ótima geralmente torna o processo não atrativo ou até mesmo proibitivo.

Algumas formas de avaliação direta da qualidade de um determinado *kernel* perante a uma base de interesse se apresentam na literatura, porém, da mesma forma, com um alto custo de avaliação, geralmente implementadas através de um procedimento de otimização. Buscam indicar a qualidade esperada da aplicação de uma função *kernel* por meio de medidas que não definem um valor específico, mas critérios de regularidade em

certos espaços, como, por exemplo, o RKHS (NGUYEN; HO, 2007). Entre as medidas utilizadas pode-se citar as baseadas em risco de regularização (SCHÖLKOPF et al., 2002), *hiperkernels* (ONG et al., 2005), entre outras.

Uma estratégia que vem se desenvolvendo recentemente são modelos de combinação de *kernels*, onde, por meio de uma combinação linear de diversos tipos de *kernel* busca-se uma melhor representatividade do modelo final em relação aos rótulos da base avaliada (CRISTIANINI et al., 2002). Tais modelos geralmente se utilizam de métodos de otimização na busca da parametrização ótima da combinação linear associada ao mapeamento ótimo, usualmente ajustando o *kernel* alinhado com os rótulos. O custo computacional desta busca tende a tornar a obtenção da solução ótima proibitiva. Assim, técnicas heurísticas de busca permitem a obtenção de alinhamentos subótimos que geralmente apresentam melhor eficiência que o uso de um *kernel* específico. A escolha dos *kernels* que irão participar do processo e seus respectivos parâmetros influenciam expressivamente na qualidade do alinhamento final.

Medidas de similaridade ou alinhamento se apresentam como competitivas principalmente por serem de baixo custo computacional, sendo calculadas com complexidade $O(n^2)$. Utilizando-se de validação cruzada para ajuste de parâmetros ou seleção de características, esta complexidade computacional torna o processo de busca da matriz de alinhamento ótimo viável. Deve-se ressaltar que, apesar da vantagem em relação ao custo computacional, algumas questões relativas ao desempenho das medidas de similaridades de matrizes *kernel* devem ser consideradas.

A seguir, serão apresentadas as estratégias desenvolvidas no trabalho visando, inicialmente, um modelo de seleção de parâmetros usando medidas de alinhamento para a escolha de parâmetros ótimos de um *kernel* para uma base específica. Na continuidade, um modelo de seleção de características, também com seu de desenvolvimento baseando-se em matrizes de alinhamento é apresentado com a expectativa de aumentar a efetividade da classificação binária quando se adota máquinas *kernel* no processo de aprendizagem. Importante ressaltar que é comum encontrar na literatura, técnicas que implementam unicamente a seleção de parâmetros (STAELIN, 2003) ou somente a seleção de características (DASH; LIU, 1997). Quando é proposta a seleção de parâmetros e características em conjunto, geralmente é implementada em um único algoritmo ou etapa (PEROLINI, 2010). Optou-se, aqui, por desenvolver o método de seleção de parâmetros e o de seleção de características desacoplados, porém interagindo entre si, visando um controle eficiente de cada etapa, visto que serão construídos com componentes de máquinas *kernel* menos usuais, sendo, assim, necessária uma avaliação rigorosa de seu comportamento e potencial.

4.1 SELEÇÃO DE HIPERPARÂMETROS

Métodos de seleção de modelos apresentam grande relevância em aprendizagem supervisionada justamente por permitirem, se bem conduzidos, uma melhora expressiva na qualidade da predição e generalização obtida. Em se tratando do uso de máquinas *kernel* na classificação, a seleção torna-se ainda mais relevante devido à importância do mapeamento realizado no espaço de características para que o discriminante linear seja efetivo.

No caso de seleção de modelos em máquinas *kernel*, dois níveis de ajustes são necessários, a saber:

- Escolha do *kernel* adequado;
- Determinação dos parâmetros do *kernel* escolhido.

A busca do *kernel* ótimo para a base avaliada, bem como de seus parâmetros ajustados, torna a tarefa árdua devido à complexidade da busca destes componentes simultaneamente. Desta forma, pode-se optar por promover, em uma primeira etapa, o ajuste do *kernel* ótimo (com parametrização fixa) e, em uma segunda fase, busca-se pelo ajuste dos parâmetros ótimos do *kernel* adotado. Apesar da simplificação do processo de seleção, não se tem a garantia da escolha do melhor *kernel* para a base em questão.

Outra estratégia, visando facilitar o procedimento de seleção, é que se defina o *kernel* a ser utilizado previamente, concentrando os esforços da busca nos parâmetros ótimos do mesmo. Geralmente, o *kernel* gaussiano é adotado devido à sua capacidade de generalização, com seu parâmetro obtido através de um processo de seleção de modelos.

Em relação ao uso de medidas de similaridade para matrizes *kernel* para avaliação da qualidade do alinhamento, estes dois níveis de ajustes também são necessários para que se obtenha o alinhamento mais efetivo para a base.

O modelo desenvolvido neste trabalho tem como objetivo apresentar uma estratégia de seleção de parâmetros para máquinas *kernel* que não sejam necessários os dois níveis de ajustes acima descritos, gerando um procedimento mais simples e efetivo no ajuste dos parâmetros ótimos. A forma de se viabilizar este modelo de seleção para máquinas *kernel* só pode ser alcançada caso seja viável a utilização de um *kernel* que tenha a flexibilidade e adaptabilidade para modelar diversos tipos de *kernel* e, ao mesmo tempo, apresentar um número razoável de parâmetros a serem ajustados. Desta forma, passa a ser viável a seleção do *kernel* adequado e seus parâmetros ótimos em uma única etapa, simplificando o processo de busca necessário.

O *kernel* PUK, descrito na Subseção 2.1.1, de acordo com avaliação de Abakar e Yu (2014), apresenta parâmetros suficientes para diversos tipos de mapeamento comumente utilizados. Possui, em sua formulação construtiva, o potencial para o mapeamento dos

dados de forma linear, polinomial e gaussiana, dependo dos valores determinados para os seus dois parâmetros associados ao mapeamento, ω e σ.

Neste trabalho, foi realizada uma avaliação visando determinar, principalmente, as seguintes questões: (i) qual dos parâmetros ω e σ é mais influente no mapeamento; (ii) quais faixas destes parâmetros estão associadas à representação de cada tipo de *kernel* e (iii) o potencial real do *kernel* PUK para se adaptar a bases em níveis distintos de separabilidade e não separabilidade linear. Este relevante estudo apresenta em suas avaliações e conclusões que, utilizado de forma adequada, ou seja, com os parâmetros ajustados corretamente, o *kernel* PUK pode sim se adaptar na ampla faixa de mapeamento linear e não linear.

É importante ressaltar que, apesar do potencial de adaptação do *kernel* PUK às necessidades de mapeamento da base para a aplicação efetiva de um discriminante linear, o ajuste não é trivial, devendo ser feito com critério e respeitando a influência de correlação dos parâmetros a serem ajustados no mapeamento.

Para o objetivo específico deste trabalho, o *kernel* PUK traz o nível de simplificação adequado na implementação de um modelo de seleção de parâmetros efetivo para máquinas *kernel*. Sua implementação em um processo de seleção pode ser realizada utilizando um classificador linear, como o SVM, por exemplo, para direcionar a busca pelos parâmetros ótimos, geralmente por meio de um procedimento de validação cruzada. O modelo mais utilizado para este tipo de seleção, envolve um otimizador que tem como objetivo maximizar uma função na forma:

$$\max_{\omega_i,\sigma_i} \; f_i(\omega_i, \sigma_i; P_{C_i}), \tag{4.1}$$

onde P_{C_i} é o nível de predição do classificador C_i obtido usando o *kernel* PUK com os parâmetros ω_i e σ_i. O nível de predição adotado depende da base em questão e das necessidades do usuário podendo ser medida F, medida ROC, predição da porcentagem de acertos etc. Apesar de ser a estratégia comumente adotada, o uso de um classificador linear como referência para avaliação dos parâmetros do mapeamento apresenta um alto custo computacional. Além disto, a determinação dos parâmetros é influenciada diretamente pelo discriminante linear utilizado.

O modelo aqui apresentado visa justamente contornar estas duas características de um modelo de seleção de máquinas *kernel* baseado em classificadores. Para isto, o nível de predição determinado pelo classificador associado será substituído por uma medida de alinhamento matricial do *kernel* PUK, ficando a otimização:

$$\max \; f_i(\omega_i, \sigma_i; MS_i), \tag{4.2}$$

onde MS_i é a medida de similaridade ou valor de alinhamento obtido para os parâmetros ω_i e σ_i usados no mapeamento. Este valor de alinhamento pode ser obtido com os diversos

modelos descritos no capítulo anterior. Sua baixa demanda computacional permite que esta etapa seja realizada com um custo muito menor em relação ao uso de um classificador de referência.

Desta forma, um modelo geral de seleção de hiperparâmetros baseado em medida de alinhamento pode ser definido de acordo com o Algoritmo 1.

Algoritmo 1: Seleção de hiperparâmetros baseado em medida de alinhamento

```
1 início
2 |  definição de medida de similaridade, parâmetros iniciais kernel,
  |   otimizador;
3 |  repita
4 |  |  construção da matriz kernel;
5 |  |  cálculo da medida de alinhamento;
6 |  |  cálculo da função objetivo;
7 |  |  atualização dos parâmetros pelo otimizador;
8 |  até convergência;
9 fim
```

Pode-se evitar a necessidade de cálculo de gradientes da função objetivo em relação aos parâmetros do *kernel* PUK adotando-se métodos de otimização de ordem zero como, por exemplo, algoritmos evolutivos.

Apesar das vantagens inerentes de um modelo de seleção utilizando matriz *kernel* e medida de similaridade é importante ressaltar que o valor da medida de similaridade é uma condição suficiente para o bom desempenho do mapeamento mas não necessária. Desta forma, a busca ótima dos parâmetros pelo otimizador pode ser dificultada visto que valores baixos de similaridade não necessariamente indicam um mapeamento de baixa qualidade. Assim, em uma etapa prévia à implementação deste processo de seleção apresentado, é importante que avaliações adicionais sejam realizadas visando um melhor entendimento do comportamento tanto do *kernel* PUK como das medidas de similaridades utilizadas. Em relação ao *kernel* PUK é importante avaliar se os parâmetros ω e σ neste contexto de uso, ou seja, em conjunto com medidas de similaridade, têm flexibilidade e potencial para adaptação aos diversos níveis de separabilidade apresentados em bases distintas, ajustando adequadamente os parâmetros. Outra avaliação relevante diz respeito à qualidade das medidas de similaridade em relação ao erro obtido por um discriminante linear. Na prática, a questão de maior relevância é verificar se existe correlação entre medida de similaridade e erro do classificador para variações tanto do *kernel* quanto dos parâmetros dos mesmos. Estas avaliações serão realizadas em uma primeira fase dos experimentos, oferecendo, assim, subsídios para um melhor entendimento do comportamento do modelo de seleção apresentado perante modelos já estabelecidos, baseados em classificadores de larga margem.

4.2 SELEÇÃO DE CARACTERÍSTICAS

Estratégias de seleção de características são uma das formas mais utilizadas para a redução de dimensionalidade de um base de dados de interesse (DING et al., 2012; JAIN et al., 2000). Um procedimento de seleção de características usualmente busca a determinação de um subconjunto ótimo de atributos da base avaliada baseando-se em alguma medida de referência. Em se tratando de máquinas *kernel*, é bastante importante a definição do espaço onde será definida a medida de referência, ou seja, espaço de entrada das variáveis ou diretamente no espaço de características mapeado pelo *kernel*. Inicialmente, cabe ressaltar que existem três formas básicas construtivas de seleção de características, a saber:

- Seleção em filtro;
- Seleção embutida;
- Seleção encapsulada.

No caso de seleção em filtro, trata-se de um pré-processamento onde medidas intrínsecas em relação aos atributos são utilizadas para caracterização dos mesmos como: (i) relevantes; (ii) irrelevantes e (iii) redundantes. Esta caracterização permite criar critérios para definição do subconjunto ótimo de atributos da base avaliada em relação à métrica adotada. Tratando-se de um procedimento de seleção embutida, busca-se, de forma concomitante com a determinação da hipótese de indução gerada por um classificador predefinido, utilizar alguma medida indicativa da qualidade dos atributos inerente do classificador para viabilizar uma avaliação da qualidade dos atributos perante a generalização e consequentemente, a determinação do subconjunto ótimo, geralmente por meio de um procedimento iterativo. No modelo de seleção encapsulada, um otimizador predefinido direciona a busca do subconjunto ótimo de atributos usando como referência uma função objetivo construída usando uma medida de interesse disponibilizada por um classificador, também predeterminado, para definir a qualidade do subconjunto de atributos avaliado. Neste modelo, é bastante comum o acoplamento na função objetivo de uma parcela associada à penalização das soluções com uma maior cardinalidade, priorizando, assim, subconjuntos que apresentem um menor número de atributos.

De forma geral, os modelos em filtro apresentam um menor custo computacional, justamente por serem implementados como pré-processamento, com os modelos encapsulados demandando um maior custo devido a necessidade de geração de uma hipótese de indução para cada conjunto avaliado. Porém, os modelos encapsulados determinam soluções para o subconjunto de atributos com alto nível de aderência ao classificador de referência, permitindo, assim, um melhor potencial de discriminação e generalização quando se usam os atributos selecionados.

Em se utilizando máquinas *kernel*, a definição da medida associada à qualidade dos atributos deve ser bem determinada. Dado que o classificador linear a ser utilizado será aplicado no espaço de características, uma seleção em filtro, por exemplo, no espaço de entrada pode ser bastante inefetiva gerando inclusive distorções na escolha dos atributos de interesse. Por outro lado, aplicando-se uma matriz *kernel* para mapeamento dos dados no espaço de características, perde-se a representatividade original dos atributos, o que também pode distorcer o resultado final. Uma possível solução é determinar a escolha do subconjunto de atributos no espaço de entrada verificando, a seguir, seu impacto no espaço de características por meio de um classificador linear ou medida de similaridade.

Feita estas considerações, tem-se o interesse em desenvolver um modelo de seleção de características encapsulado, substituindo um classificador linear na avaliação dos subconjuntos selecionados por uma medida de similaridade obtida com a matriz *kernel* referente a tais atributos. Desta forma, consegue-se avaliar o impacto direto dos atributos selecionados no espaço de características, onde será gerada a hipótese de indução.

Neste trabalho, com desenvolvimentos em seleção de parâmetros e seleção características, optou-se pela construção de uma estratégia em duas fases. Em uma primeira etapa, aplica-se a seleção de parâmetros obtendo o *kernel* e sua parametrização adequada, conforme descrito na seção anterior. Definida a seleção de parâmetros, os dados do mapeamento no espaço de características servem de ponto de partida para a implementação do modelo de seleção de características. Entende-se que este desacoplamento não compromete a escolha adequada na primeira etapa, de seleção de parâmetros, principalmente quando se usa medidas de similaridade, visto que um valor alto de similaridade é condição suficiente e não necessária para um bom mapeamento. Ou seja, para a determinação de um *kernel* adequado e sua parametrização o uso de todos os atributos de uma base de interesse serve como limite inferior do valor de similaridade para o mapeamento testado. Além disto, algumas vantagens favorecem a construção de uma estratégia em duas etapas:

- Espaço de busca simplificado em cada etapa desacoplada;
- Uso de parâmetros otimizados na segunda etapa;
- Valor do alinhamento com todos atributos ativos como referência para a inicialização da segunda etapa.

Uma outra vantagem para a construção do modelo é o uso de medidas de similaridade para a busca dos atributos ótimos no espaço de características. A construção da matriz *kernel* e verificação da qualidade do subconjunto a ser avaliado apresenta um custo computacional relativamente baixo, permitindo a construção de modelos de seleção de características encapsulados competitivos em termos de custo computacional.

Para a construção de um modelo de seleção de atributos encapsulado é necessário a definição de uma estratégia determinística para a avaliação dos subconjuntos a serem avaliados ou um otimizador para direcionar a busca ótima deste subconjunto. Os conhecidos modelos determinísticos para frente (*forward*) ou para trás (*backward*) implementam métodos de retirada ou inclusão de atributos a cada iteração, avaliando se devem ser mantidos ou não no subconjunto procurado. Apesar de eficiente, estes métodos tendem a obter ótimos locais, principalmente por não reavaliarem as escolhas previamente feitas. Além disto, este modelo de busca de atributos ótimos tende a ser ineficaz quando associado à medidas de similaridade, visto que tais medidas obrigatoriamente não refletem diretamente a qualidade dos atributos avaliados.

Assim, uma busca baseada em procedimentos estocásticos permite que avaliações sem um fluxo predefinido realizem uma busca mais ampla no que tange a combinações de atributos e seu reflexo na medida de similaridade adotada, principalmente em relação à solução de referência fornecida pelo procedimento de seleção de parâmetros previamente executado. Usualmente, adota-se meta-heurísticas de ordem zero como otimizador para que se tenha uma busca eficiente do subconjunto ótimo de atributos.

Como etapa final da construção do modelo de seleção de características deve-se determinar a função objetivo a ser adotada na otimização. No caso de seleção de características, geralmente constrói-se uma função penalizada do tipo:

$$F_i = f_i + \alpha_p \cdot P_i, \tag{4.3}$$

onde, F_i é a função penalizada da i-ésima solução, f_i o objetivo a ser otimizado, α_p o coeficiente de penalização e P_i a função de penalização da i-ésima solução. A função de penalização, em se tratando de seleção de características, geralmente é construída visando aumentar a prioridade de soluções que apresentem um menor número de atributos. O equilíbrio entre as duas parcelas da função penalizada é determinado pelo coeficiente de penalização α_p, sendo sua escolha crucial para um bom desempenho da otimização.

Desta forma, apresenta-se o pseudocódigo do modelo de seleção de características desenvolvido no Algoritmo 2.

4.3 CONSIDERAÇÕES ADICIONAIS E DETALHAMENTO

Apresenta-se, aqui, algumas considerações adicionais que direcionaram a construção do fluxograma de seleção de parâmetros e características desenvolvidos neste trabalho.

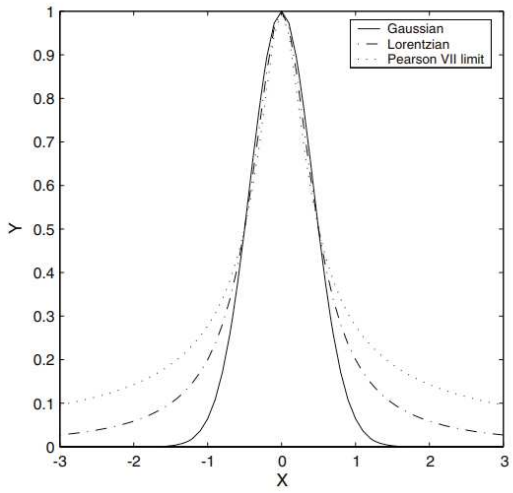
Primeiramente, é importante validar o uso do *kernel* PUK como referência na construção da seleção de modelos. Sua formulação referencia um potencial de adaptação, baseado na escolha adequada de seus parâmetros, de mapeamentos de bases com diversos níveis de não linearidade. Porém, é importante ter um melhor entendimento da influência de cada parâmetro e das suas faixas de utilização. Estudos prévios buscaram avaliar este

Algoritmo 2: Modelo de seleção de características

```
1 início
2 |  definição de medida de similaridade, solução de referência (parâmetros do
  |   kernel), otimizador (estratégia de busca);
3 |  repita
4 |  |  construção da matriz kernel;
5 |  |  cálculo da medida de alinhamento;
6 |  |  cálculo da função objetivo;
7 |  |  atualização do subconjunto de características pelo otimizador;
8 |  até convergência;
9 fim
```

comportamento, inclusive definindo faixas onde os parâmetros fazem um mapeamento linear e faixas de mapeamentos não lineares. Em Abakar e Yu (2014) tem-se o indicativo que numa distribuição de ω considerando a forma de pico do PUK, conforme visto na Figura 1, a região próxima de 0 pode ser comparada à função gaussiana, a região $0,25 - 0,75$ representa a função linear e o intervalo de $0 - 3$ representa uma função sigmoidal.

Figura 1 – Forma de pico da função PUK.



Fonte: Abakar e Yu (2014).

É importante ressaltar que, em conjunto com medidas de similaridade, o *kernel* PUK nunca foi avaliado, sendo necessário verificar se seu potencial de adaptação em relação ao mapeamento de bases no espaço de características pode ser capturado nesta classe de medida. Visando simplificar as avaliações e comparações, a flexibilização da margem no processo de classificação não foi adotada, quando da utilização de classificadores de larga margem.

Uma questão chave do modelo é o desacoplamento das etapas seletivas. É comum encontrar em trabalhos da área métodos somente para seleção de modelos (YOU, 2011) ou

com enfoque somente na seleção de características (DASH; LIU, 1997). Quando o trabalho trata dos dois tipos de seleção, geralmente é implementado com todos os parâmetros envolvidos sendo determinados em uma mesma busca. Por exemplo, Perolini (2010) propõe um algoritmo genético para a busca dos parâmetros do modelo e dos atributos mais representativos.

O modelo hierárquico apresentado em duas etapas tem embasamento principalmente no comportamento intrínseco das medidas de similaridade que são adotadas para avaliar os subconjuntos de atributos. Ou seja, tais medidas seriam menos sujeitas a obter ótimos locais devido ao desacoplamento. Porém, a comprovação tanto teórica quanto prática desta afirmação não é trivial. Se for comparado diretamente com o modelo acoplado, a diferença no espaço de busca entre os dois modelos, que reflete na variação da medida de similaridade com atributos distintos, pode distorcer os resultados obtidos. Porém, um indicativo de funcionamento do modelo está justamente no valor de referência vindo da seleção de parâmetros, onde todos os atributos estão ativos. Ou seja, este valor de referência representa a melhor parametrização quando se consideram todos os atributos. Uma forma de verificar a efetividade do procedimento seria reavaliar a seleção de parâmetros com a solução obtida pela seleção de características no que tange aos valores da medida de similaridade adotada. Mesmo assim, a influência da função de penalização distorce o processo de busca no sentido de inserir na função objetivo um viés para soluções que apresentem um menor número de atributos. Nos experimentos, espera-se que um maior entendimento do comportamento do modelo seja obtido.

Em relação à função objetivo para a seleção de características, sua construção, como dito, deve conter a parcela referente à medida de similaridade e a parcela associada à penalização em relação ao número de características utilizadas, geralmente combinadas aditivamente, conforme a Equação (4.3). Uma grande dificuldade na construção de funções de penalização está no adequado ajuste do parâmetro de penalização. No caso de penalização em modelos de seleção de características, este parâmetro influencia diretamente no processo de busca, tentando equilibrar soluções que tenham boa qualidade com o menor número de atributos possível. Porém, o ajuste de tal parâmetro é bastante complexo, podendo, se mal determinado, distorcer completamente a busca.

A determinação da função de penalização com o modelo desacoplado traz a vantagem de se ter a solução de referência obtida da primeira etapa da seleção. Este valor é crucial para o equilíbrio entre as parcelas que compõem a função objetivo, de forma a evitar que soluções de pior qualidade, porém com baixa cardinalidade no subconjunto de características sejam escolhidas em detrimento de soluções de melhor qualidade em relação à medida de similaridade. Para isto, a construção da função objetivo é feita na forma:

$$F(a_{ativos}) = MS(a_{ativos}) + \alpha_p \cdot MS_r \cdot \frac{\#(\bar{a}_{ativos})}{\#(a)}, \tag{4.4}$$

onde a_{ativos} são os atributos ativos a serem avaliados, MS_r é a medida de similaridade da solução de referência (vinda da seleção de parâmetros), $\#(.)$ é a cardinalidade do conjunto avaliado, $\#(\bar{.})$ a cardinalidade do conjunto complementar e α_p é o parâmetro de penalização utilizado. Este parâmetro, no contexto da função definida na Equação (4.4), apresenta como vantagem, em relação aos parâmetros de penalização padrões, uma maior facilidade em relação à sua determinação, visto que seu valor deve ser fixado na faixa $0 < \alpha_p \leq 1$, simplificando o processo de ajuste da função penalizada a ser otimizada.

É interessante avaliar que, com esta montagem da função objetivo, tem-se a garantia que somente soluções que sejam melhores que a solução de referência em relação à medida de similaridade terão valores melhores para a função de penalização. Desta forma, o equilíbrio entre a medida de similaridade e o número de atributos é substituído por um procedimento hierárquico onde somente soluções com melhor valor de medida de similaridade em relação à solução de referência serão adotados. Outra grande vantagem é a faixa restrita de definição do parâmetro de penalização na composição da função penalizada, o que facilita e simplifica sua aplicação, diminuindo a possibilidade que escolhas inadequadas prejudiquem a busca.

Em relação aos procedimentos de otimização, necessários nas duas etapas, já foi indicado a maior facilidade na utilização de otimizadores que não necessitem do cálculo de gradiente das funções. No caso específico do modelo de seleção de características, houve também o indicativo que um procedimento de busca estocástico poderia ser mais interessante devido ao padrão de desempenho da medida utilizada. Apesar de algoritmos evolutivos, baseados em população de soluções, serem bastante atraentes, principalmente algoritmos genéticos binários no caso de seleção de características pela facilidade na ativação e desativação de atributos, optou-se pela utilização de um algoritmo estocástico não populacional. Tendo a solução de referência como base, é interessante que as variações estocásticas na busca sejam iniciadas por esta referência, com a ativação e desativação estocástica de seus atributos. Um algoritmo que permite tal construção e implementa uma busca global é a meta-heurística de resfriamento simulado (*Simulated Annealing* – SA), cujo pseudocódigo é apresentado no Algoritmo 3.

Sem perda de qualidade na busca, o SA é adotado nas duas etapas do processo, principalmente por promover uma otimização global, minimizando a chance de obtenção de soluções alocadas em ótimos locais, tanto na seleção de parâmetros quanto na seleção de características. Na seleção de parâmetros, a solução inicial para o começo do processo de otimização é randômica. No caso da seleção de características, utiliza-se uma solução inicial determinística, definida pela solução de referência da seleção de parâmetros.

Finalizando, é importante apresentar um fluxograma completo envolvendo as duas etapas de seleção. Apesar das avaliações serem feitas para cada etapa, visando um melhor entendimento do comportamento de cada modelo, o objetivo é a geração de um modelo

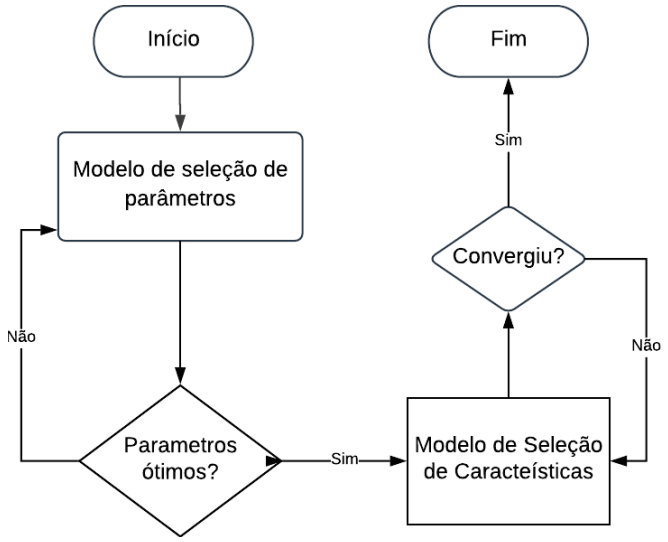
Algoritmo 3: *Simulated Annealing*

Entrada: solução inicial s_0; temperatura inicial T_0; fator de decréscimo na temperatura α; temperatura mínima T_{min};
Saída: solução s^*;

```
início
    s ← s_0;
    T ← T_0;
    enquanto T ≥ T_min faça
        enquanto não houver equilíbrio térmico na temperatura T faça
            selecionar uma solução s' dentro da vizinhança de s;
            Δf ← f(s') − f(s);
            se Δf ≥ 0 então
                s ← s';
            senão
                s ← s' com uma probabilidade p = e^(−Δf/KT);
            fim se
            se f(s) > f(s*) então
                s* ← s;
            fim se
        fim enqto
        T ← αT;
    fim enqto
fim
```

completo, envolvendo, sequencialmente, as duas etapas conforme indicado na Figura 2.

Figura 2 – Fluxograma das etapas de seleção de modelos proposta.



Fonte: Elaborado pelo autor (2019).

No próximo capítulo, experimentos numéricos são realizados visando avaliar a qualidade das medidas de similaridade, o potencial do *kernel* PUK no ajuste do mapeamento

adequado no espaço de características e o desempenho do modelo proposto em suas duas etapas construtivas.

5 EXPERIMENTOS E RESULTADOS

Foram realizados experimentos iniciais com intuito de avaliar o comportamento das medidas de similaridade utilizando o *kernel* PUK, bem como os resultados da classificação considerando sua utilização. Os testes iniciais têm o propósito de identificar relações entre os resultados obtidos pelos métodos de alinhamento com a aplicação de um classificador de larga margem para cada base de dados considerada.

Após os testes iniciais, definida a medida de similaridade com maior nível de eficiência, foram realizados experimentos para avaliar o desempenho do modelo proposto em suas duas etapas, seleção de hiperparâmetros e seleção de características.

5.1 BASES DE DADOS

Foram utilizadas, ao todo, sete bases de dados para os experimentos, todas contidas no repositório de aprendizado de máquina da UCI (BACHE; LICHMAN, 2013). A Tabela 1 mostra as informações de cada base de dados.

Tabela 1 – Informações das bases de dados.

Base	Atributos	Amostras		
		$+1$	-1	Total
Toy	11	6	6	12
Sonar	60	97	111	208
Synthetic	60	300	300	600
Robot LP4	90	24	93	117
Ionosphere	34	225	126	351
Bupa	6	145	200	345
Wine	13	107	71	178

Fonte: Elaborado pelo autor (2019).

5.2 ANÁLISE DOS DADOS

5.2.1 *K-fold*

O método *k-fold* é um método de validação cruzada que particiona o conjunto de dados em k subconjuntos mutuamente exclusivos do mesmo tamanho (WESTON et al., 2003). Um subconjunto entre os k é utilizado como teste, enquanto os outros $k-1$ são utilizados para treinar o modelo, sendo a acurácia final a soma das amostras classificadas corretamente nos subconjuntos de teste dividido pelo total de amostras da base. Tal processo é executado k vezes. O SVM foi executado, para todas as bases, utilizando um

10-10-*fold*, que realiza 10 partições, com sementes diferentes, no conjunto de dados, no intuito de reduzir o viés das partições.

5.2.2 **Geração dos Dados**

Para os testes iniciais, foi executado o classificador SVM em cada base em conjunto com o *kernel* PUK. Também foram executados os métodos de similaridade KTA e FSM, bem como as variações do KTA utilizando o conceito de centralização de matrizes e do conceito de matriz *kernel* balanceada e do FSM utilizando o conceito de centralização de matrizes.

Para cada coleta de resultados, seja para os resultados da execução do KTA, FSM ou execução do SVM, foram gerados 2 milhões de modelos com a variação dos parâmetros do PUK definidos nas faixas: σ de $0,5$ a 50 e ω de $0,05$ a 100.

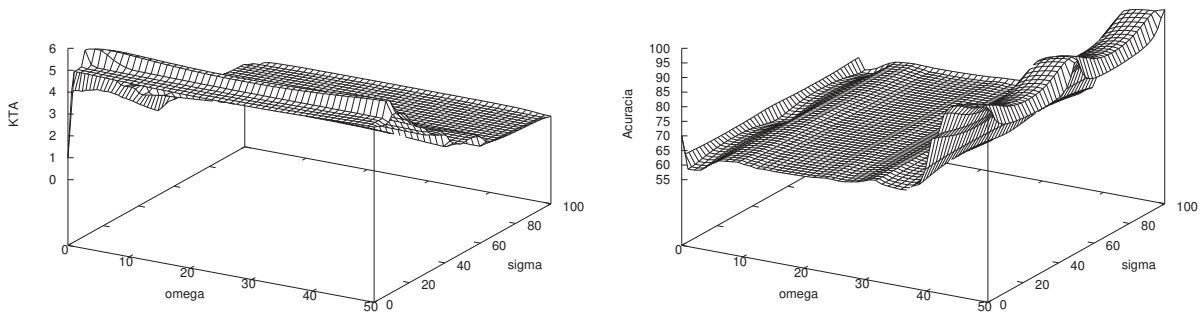
A medida de alinhamento do KTA gera valores entre 0 e 1. Para geração dos gráficos, considerando que, em alguns casos, estes valores eram muito próximos de 0, foi realizada uma padronização no intervalo de $0 \leq KTA \leq 100$. Já para o FSM, que não possui limite de valores, e tem uma relação inversa com o KTA, foi realizada a inversão dos valores, tendo como referência o valor do maior alinhamento.

As subseções seguintes apresentam representações gráficas do comportamento de cada método em comparação com a acurácia do SVM, ambos utilizado a mesma função *kernel*, sendo os gráficos da esquerda relativos à medida de similaridade e os gráficos da direita associados ao uso do classificador SVM.

5.2.3 **KTA Não Centralizado Não Balanceado**

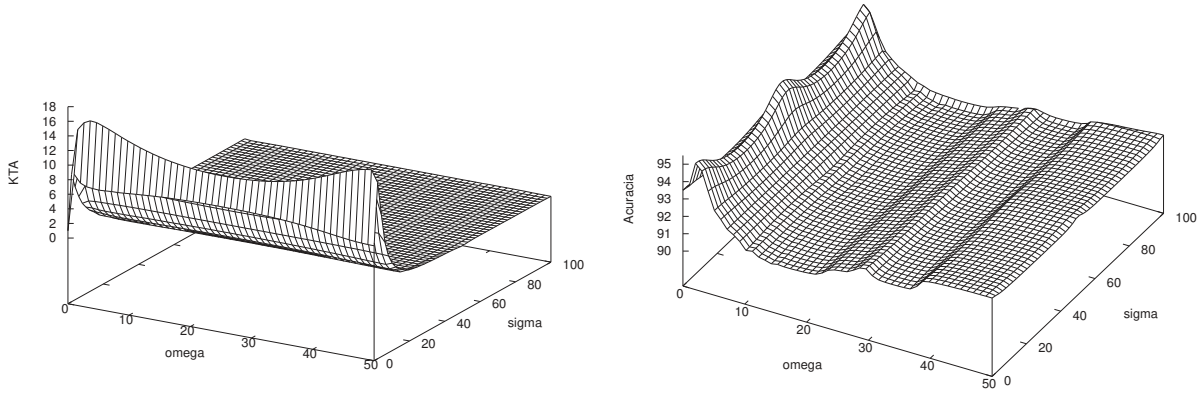
As Figuras 3, 4, 5 e 6 apresentam as representações gráficas referentes ao KTA puro, ou seja, sem centralização da matriz *kernel* e sem balanceamento.

Figura 3 – KTA Não Centralizado Não Balanceado x Acurácia - Bupa.



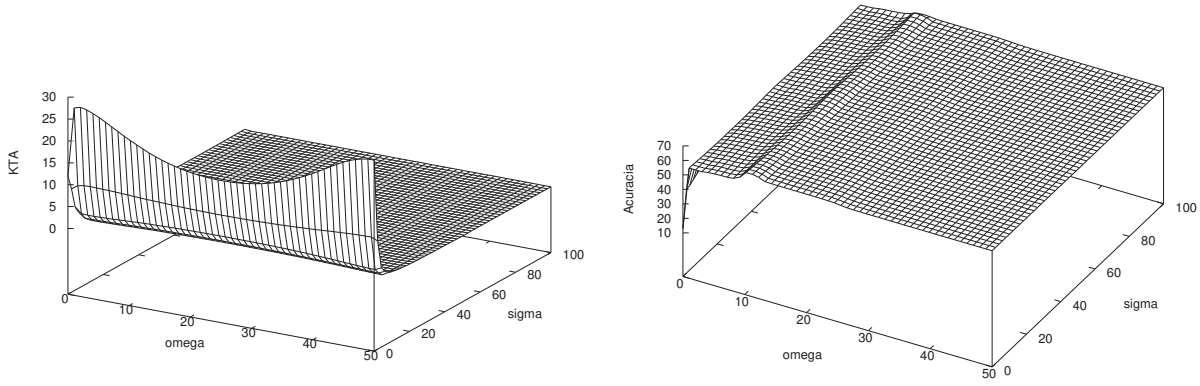
Fonte: Elaborado pelo autor (2019).

Figura 4 – KTA Não Centralizado Não Balanceado x Acurácia - Ionosphere.



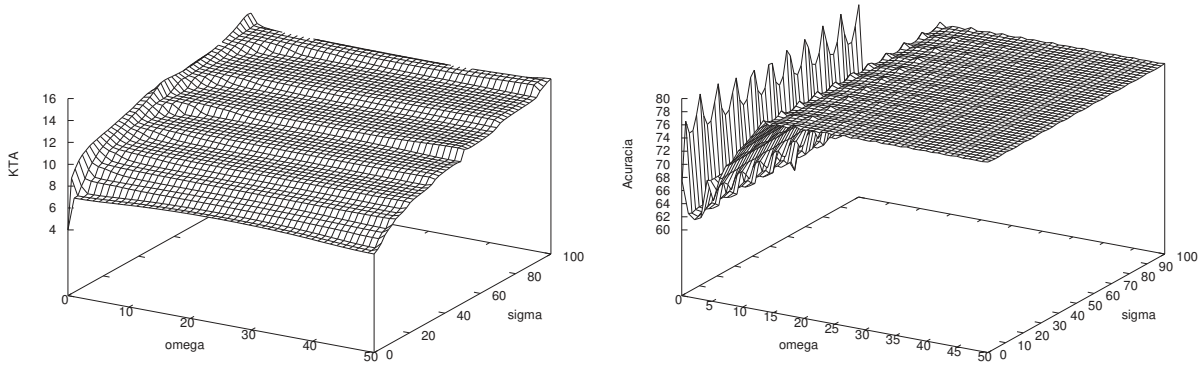
Fonte: Elaborado pelo autor (2019).

Figura 5 – KTA Não Centralizado Não Balanceado x Acurácia - Toy.



Fonte: Elaborado pelo autor (2019).

Figura 6 – KTA Não Centralizado Não Balanceado x Acurácia - Wine.

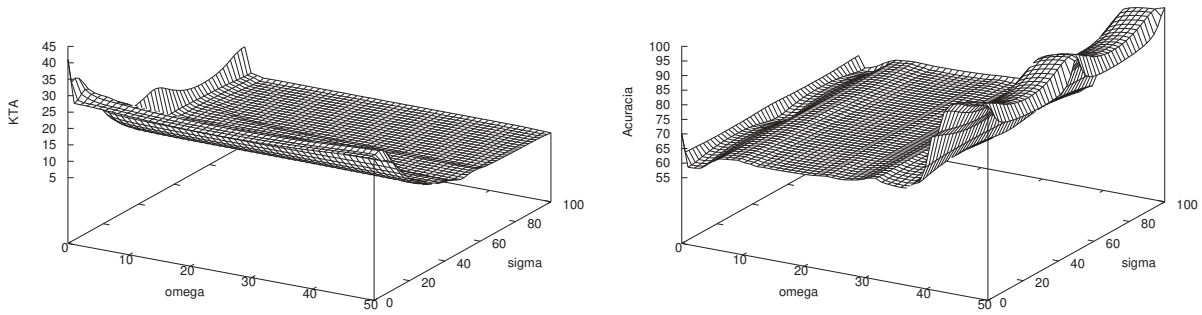


Fonte: Elaborado pelo autor (2019).

5.2.4 **KTA Centralizado Não Balanceado**

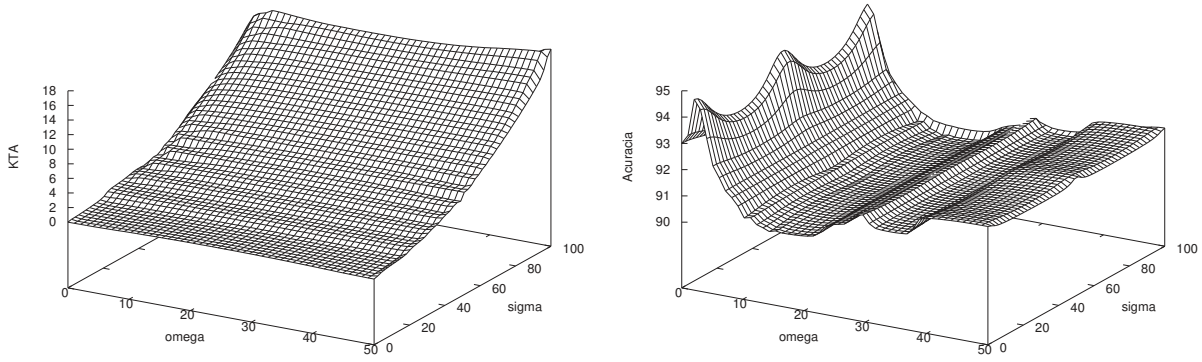
Agora considerando a matriz *kernel* centralizada, porém sem efeituar o balanceamento nas bases, foram observados os resultados apresentados nas Figuras 7, 8, 9 e 10.

Figura 7 – KTA Centralizado Não Balanceado x Acurácia - Bupa.



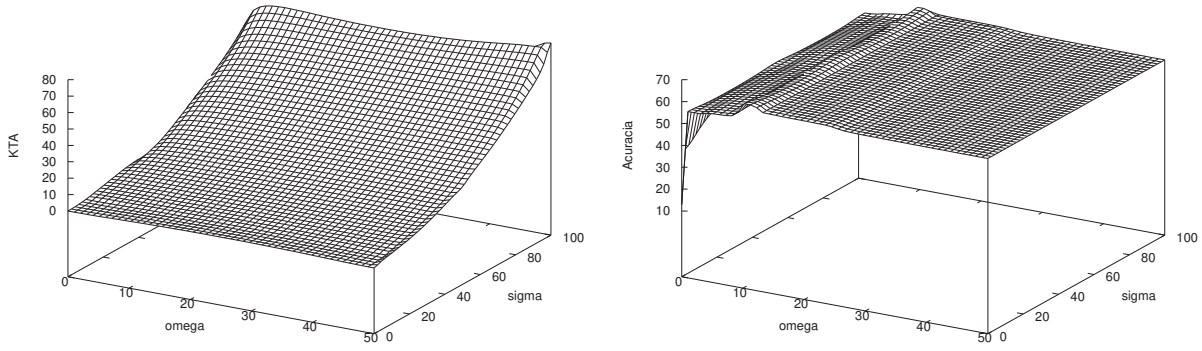
Fonte: Elaborado pelo autor (2019).

Figura 8 – KTA Centralizado Não Balanceado x Acurácia - Ionosphere.



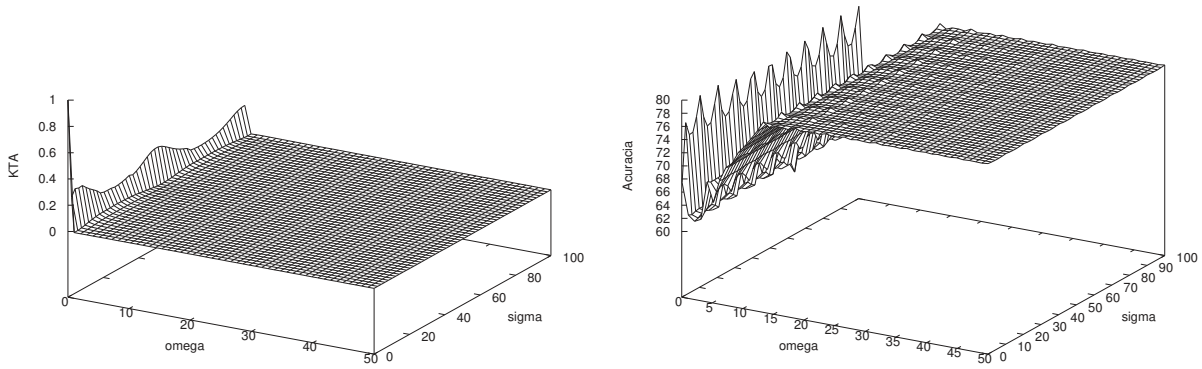
Fonte: Elaborado pelo autor (2019).

Figura 9 – KTA Centralizado Não Balanceado x Acurácia - Toy.



Fonte: Elaborado pelo autor (2019).

Figura 10 – KTA Centralizado Não Balanceado x Acurácia - Wine.

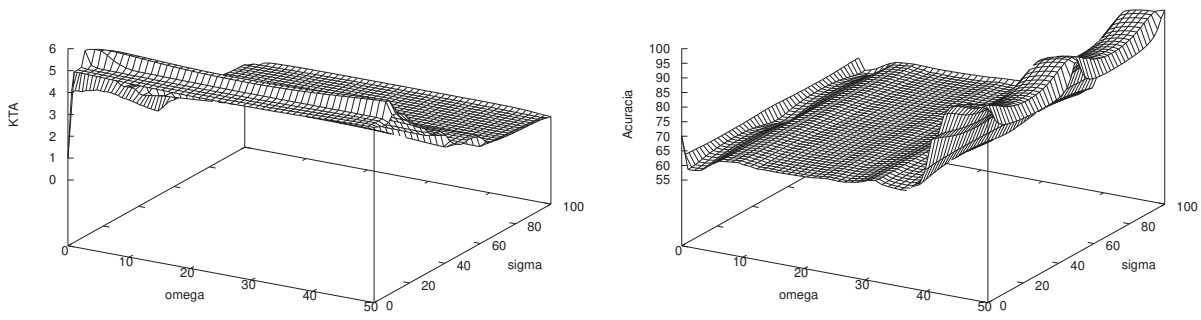


Fonte: Elaborado pelo autor (2019).

5.2.5 **KTA Não Centralizado Balanceado**

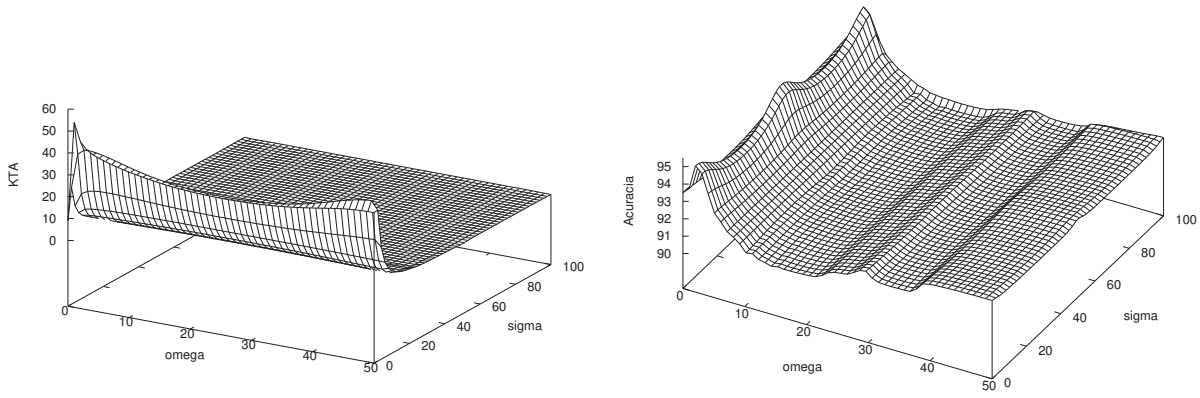
As Figuras 11, 12, 13 e 14 apresentam os resultados considerando a matriz *kernel* não centralizada, efetuando o balanceamento nas bases.

Figura 11 – KTA Não Centralizado Balanceado x Acurácia - Bupa.



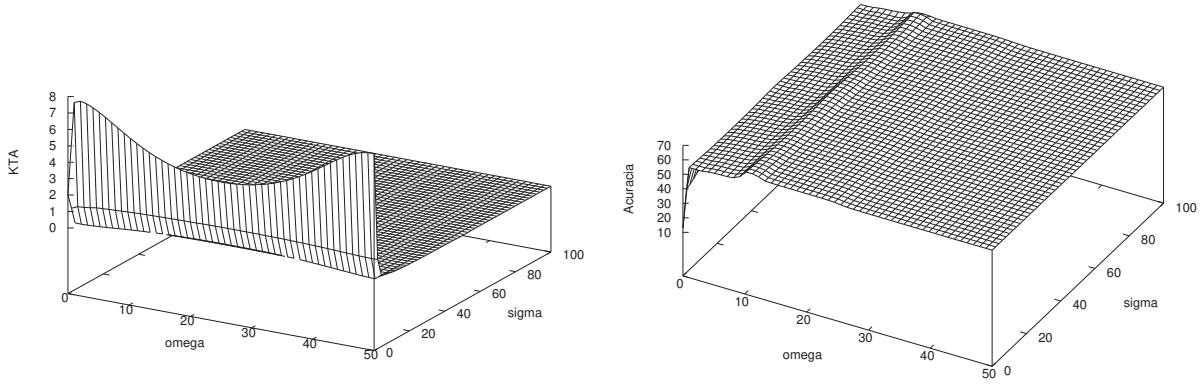
Fonte: Elaborado pelo autor (2019).

Figura 12 – KTA Não Centralizado Balanceado x Acurácia - Ionosphere.



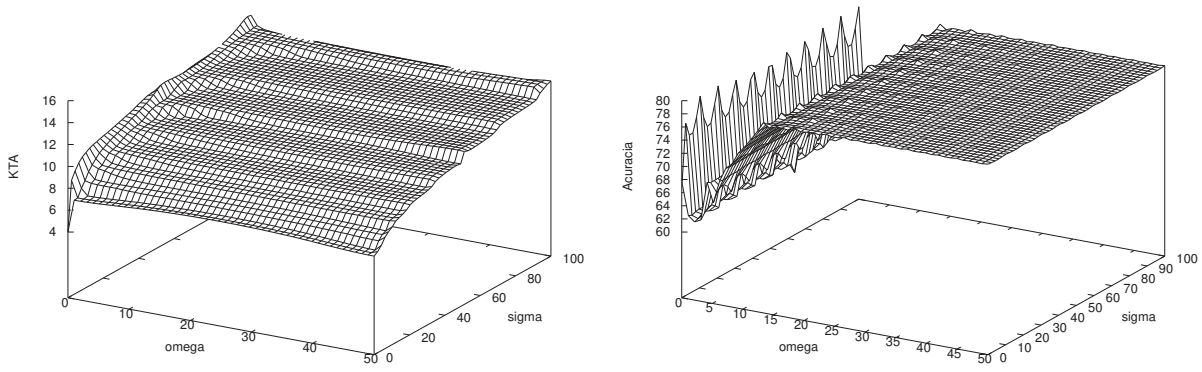
Fonte: Elaborado pelo autor (2019).

Figura 13 – KTA Não Centralizado Balanceado x Acurácia - Toy.



Fonte: Elaborado pelo autor (2019).

Figura 14 – KTA Não Centralizado Balanceado x Acurácia - Wine.



Fonte: Elaborado pelo autor (2019).

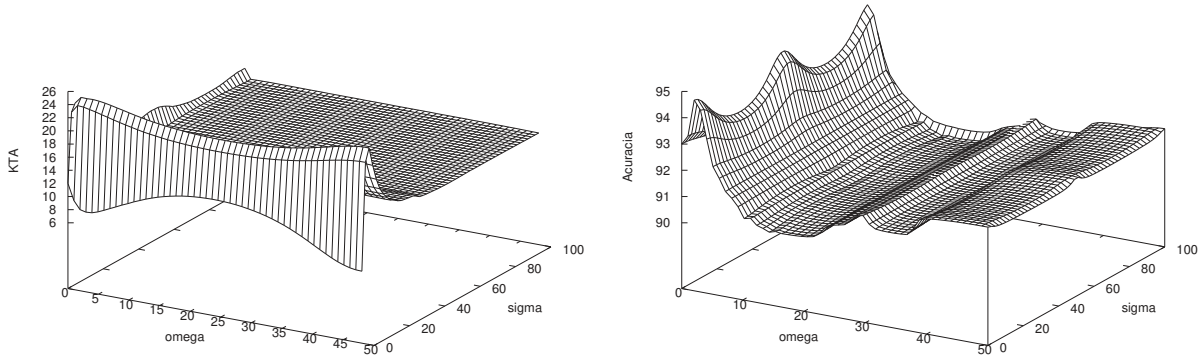
5.2.6 **KTA Centralizado Balanceado**

As Figuras 15, 16, 17 e 18 apresentam os gráficos considerando a matriz *kernel* centralizada, efetuando também o balanceamento nas bases.

Figura 15 – KTA Centralizado Balanceado x Acurácia - Bupa.

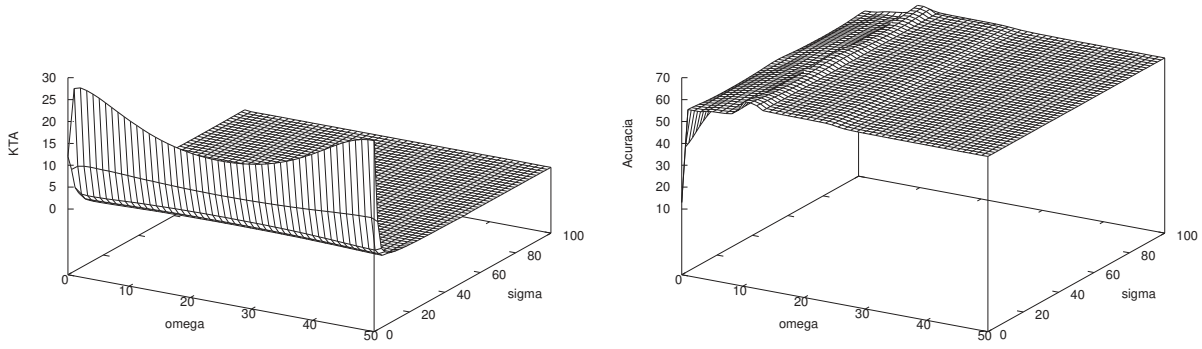
Fonte: Elaborado pelo autor (2019).

Figura 16 – KTA Centralizado Balanceado x Acurácia - Ionosphere.



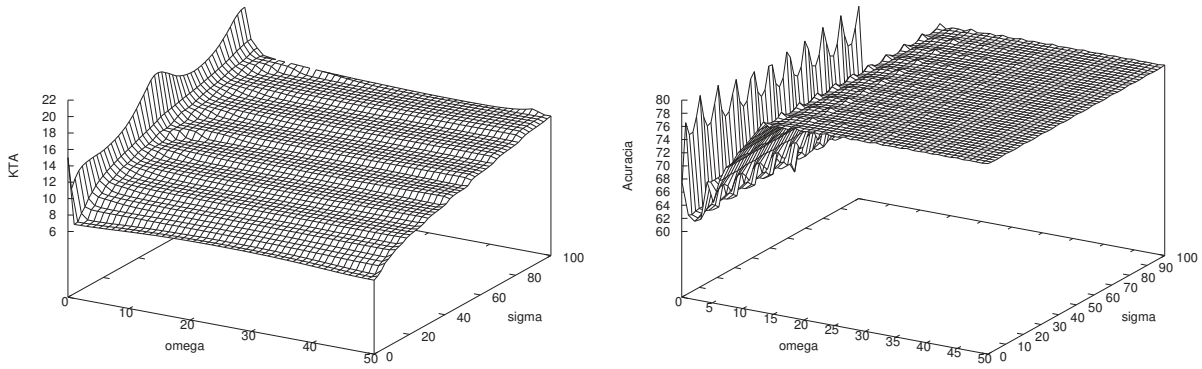
Fonte: Elaborado pelo autor (2019).

Figura 17 – KTA Centralizado Balanceado x Acurácia - Toy.



Fonte: Elaborado pelo autor (2019).

Figura 18 – KTA Centralizado Balanceado x Acurácia - Wine.

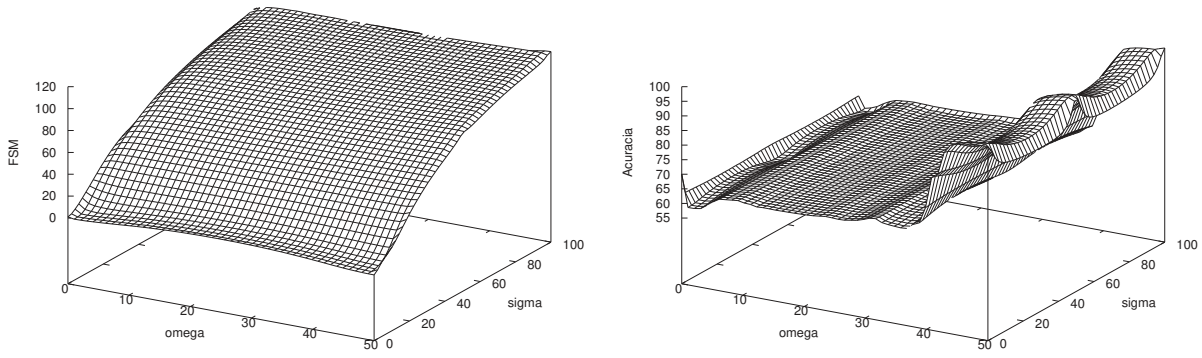


Fonte: Elaborado pelo autor (2019).

5.2.7 **FSM Não Centralizado**

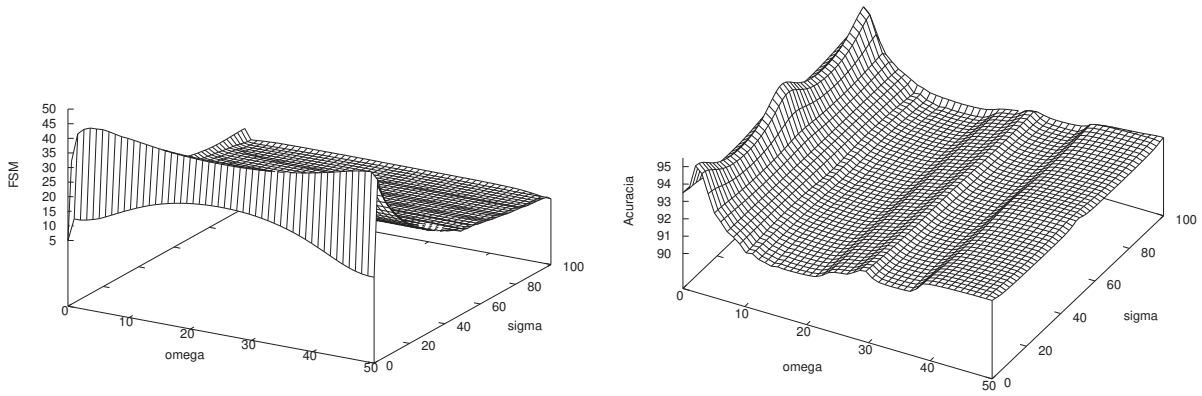
As Figuras 19, 20, 21 e 22 apresentam os resultados do FSM, sem considerar a centralização da matriz *kernel*.

Figura 19 – FSM Não Centralizado x Acurácia - Bupa.



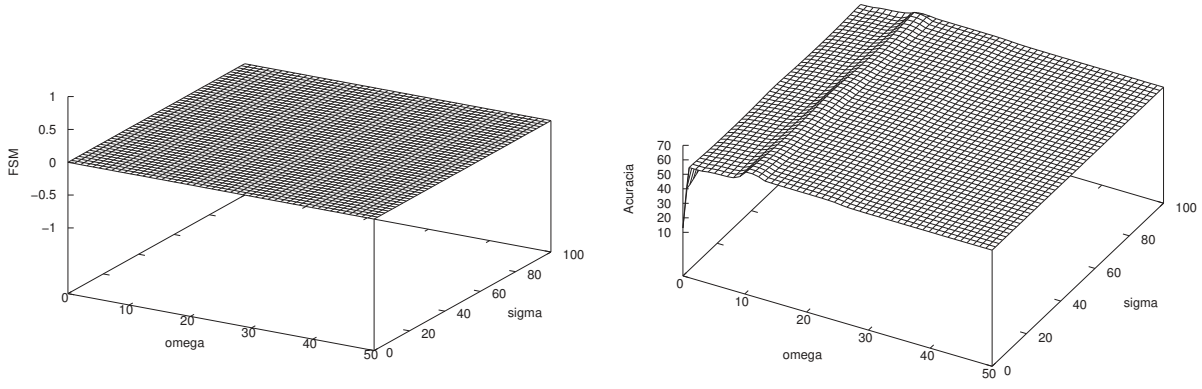
Fonte: Elaborado pelo autor (2019).

Figura 20 – FSM Não Centralizado x Acurácia - Ionosphere.



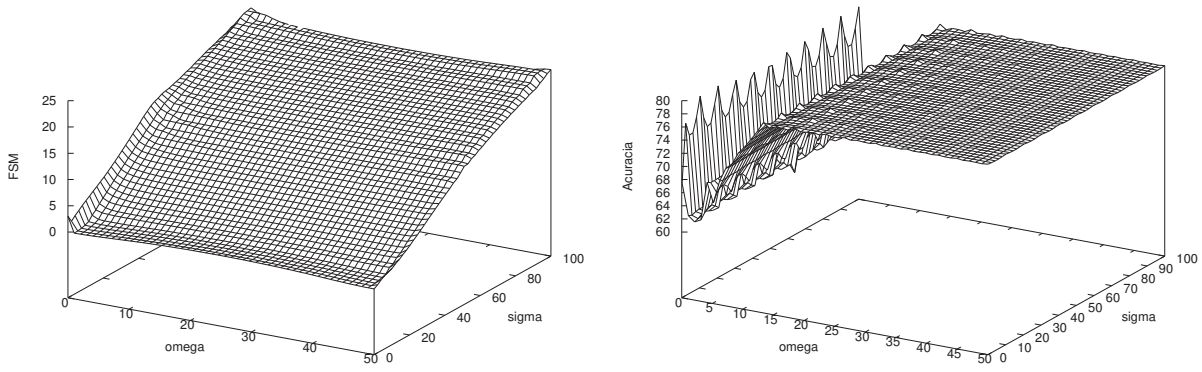
Fonte: Elaborado pelo autor (2019).

Figura 21 – FSM Não Centralizado x Acurácia - Toy.



Fonte: Elaborado pelo autor (2019).

Figura 22 – FSM Não Centralizado x Acurácia - Wine.

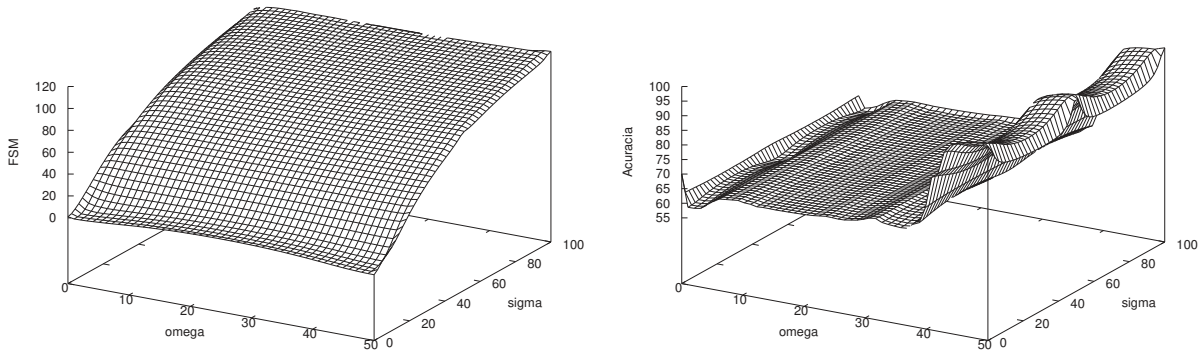


Fonte: Elaborado pelo autor (2019).

5.2.8 **FSM Centralizado**

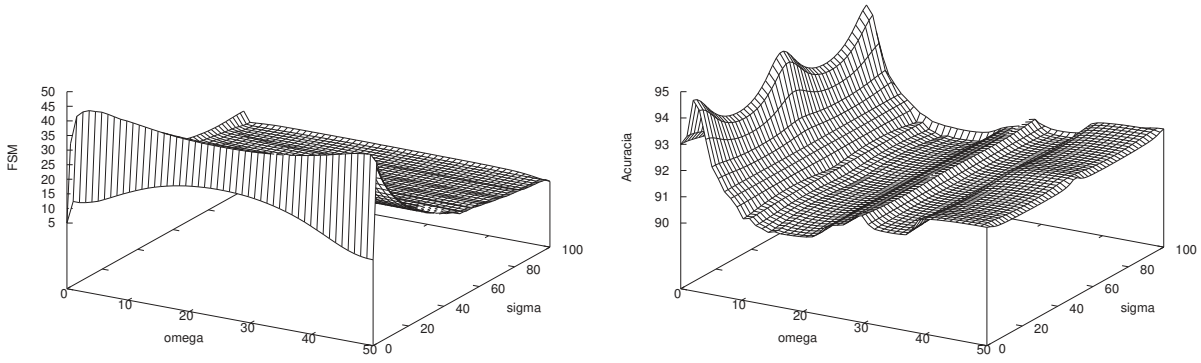
As Figuras 23, 24, 25 e 26 apresentam os resultados do FSM, considerando a matriz *kernel* centralizada.

Figura 23 – FSM Centralizado x Acurácia - Bupa.



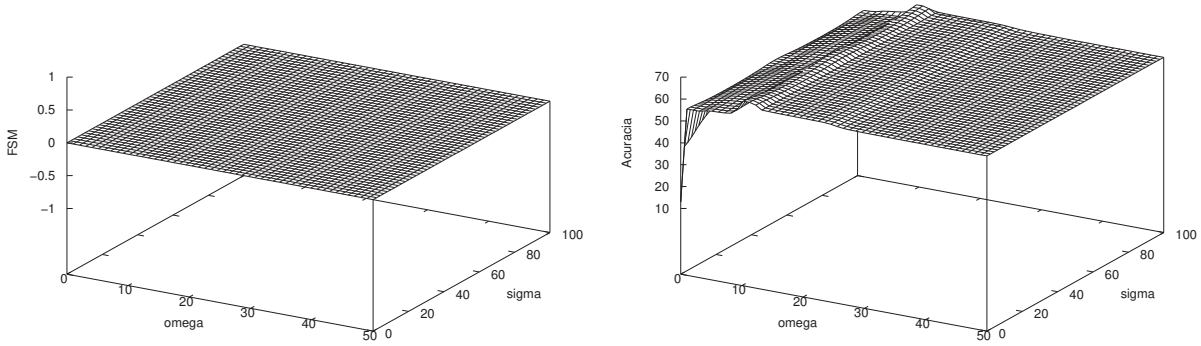
Fonte: Elaborado pelo autor (2019).

Figura 24 – FSM Centralizado x Acurácia - Ionophere.



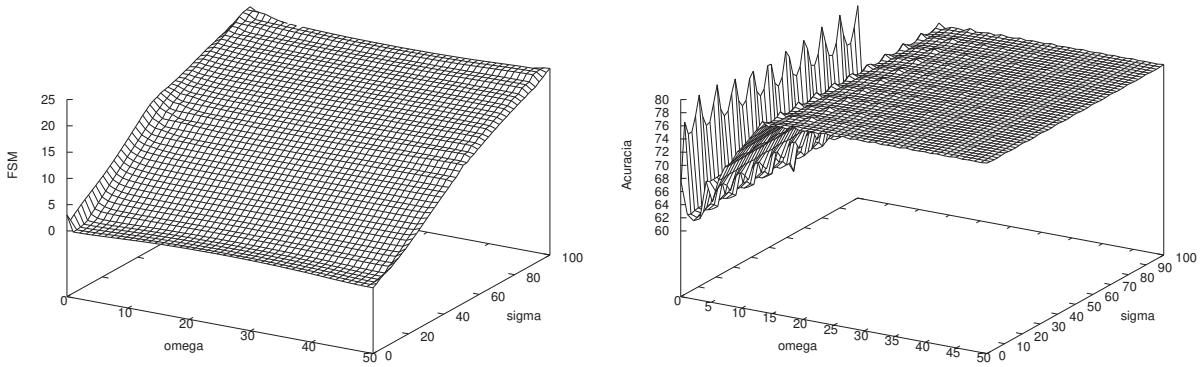
Fonte: Elaborado pelo autor (2019).

Figura 25 – FSM Centralizado x Acurácia - Toy.



Fonte: Elaborado pelo autor (2019).

Figura 26 – FSM Centralizado x Acurácia - Wine.



Fonte: Elaborado pelo autor (2019).

5.3 ANÁLISE DOS RESULTADOS

Após os testes iniciais feitos com as medidas de similaridade avaliadas, foi possível observar o comportamento dos alinhamentos em relação à acurácia do SVM. A escolha da utilização apenas da função PUK é devido ao seu potencial de adaptação ao padrão de *kernels* distintos apenas variando os valores de seus parâmetros, consequentemente, diminuindo o gasto de tempo de execução de cada função separadamente.

Além disso, avaliando os resultados, foi possível observar a suficiência dos métodos de alinhamento associados a um bom desempenho do SVM. Medidas de similaridade, principalmente as baseadas no KTA, apresentam-se como medidas de suficiência, ou seja, nem todo valor alto de acurácia significa uma relação com um valor alto de alinhamento, mas os valores altos do alinhamento estão ligados a um bom desempenho do classificador.

As tabelas a seguir informam os valores médios e os melhores valores para cada medida de similaridade testada, relacionada com a acurácia (Maior Al.) gerada pela mesma combinação de parâmetros da função PUK utilizando o classificador SVM. Além das informações sobre as relações dos alinhamentos com as acurácias, as tabelas apresentam os maiores valores de acurácia (Maior Ac.) encontrados pela busca para cada base, que servem de referência para validar a qualidade dos melhores alinhamentos. O objetivo desta análise inicial é verificar qual medida de similaridade será escolhida para a integração nos métodos propostos para a seleção de modelos utilizando valores de alinhamento. Para o KTA, os valores originais foram mantidos, ou seja, entre 0 e 1.

Na Tabela 2 os dados representam o comportamento do KTA sem alterações (não centralizado e não balanceado). Para cada base testada, nem todos os valores gerados pelo KTA estão correlacionados a valores altos da acurácia, considerando a média, das acurácias geradas pelo SVM.

Tabela 2 – KTA Não Centralizado Não Balanceado - Alinhamento x Acurácia.

Base	Alinhamento		Acurácia		
	Média	Maior	Média	Maior Al.	Maior Ac.
Wine	0,00488912	0,007947	65,81	60,10	89,08
Ionosphere	0,00049033	0,008202	92,29	93,82	95,05
Bupa	0,03041521	0,063107	62,85	59,49	99,56
Toy	0,00177132	0,085784	59,84	60,00	67,00

Fonte: Elaborado pelo autor (2019).

A Tabela 3 apresenta os valores do KTA utilizando o conceito de balanceamento. Na tabela é possível observar uma melhora quanto à análise anterior. Exceto pela base *Wine*, todos os valores estão relacionados a valores acima da média, tendo dois relacionados aos maiores para cada base.

Tabela 3 – KTA Não Centralizado Balanceado - Alinhamento x Acurácia.

Base	Alinhamento		Acurácia		
	Média	Maior	Média	Maior Al.	Maior Ac.
Wine	0,11760737	0,157088	65,81	60,10	89,10
Ionosphere	0,01286048	0,179254	92,29	95,05	95,05
Bupa	0,03026111	0,063084	62,85	70,04	99,56
Toy	0,01092416	0,253001	59,84	67,00	67,00

Fonte: Elaborado pelo autor (2019).

A Tabela 4 apresenta os valores de alinhamento do KTA com o conceito de centralização. Foi observada, na média, uma queda nos valores do KTA em comparativo aos exemplos anteriores. Além disso, os valores da acurácia do SVM correspondentes caíram quase todas em relação ao KTA com balanceamento.

Tabela 4 – KTA Centralizado Não Balanceado - Alinhamento x Acurácia.

Base	Alinhamento		Acurácia		
	Média	Maior	Média	Maior Al.	Maior Ac.
Wine	0,00520600	0,008934	65,81	79,08	89,10
Ionosphere	0,06221412	0,186827	92,29	92,91	95,05
Bupa	0,00119856	0,005739	62,85	70,03	99,56
Toy	0,00809240	0,302009	59,84	57,00	67,00

Fonte: Elaborado pelo autor (2019).

A Tabela 5 informa os dados de medida de similaridade gerados pelo KTA utilizando o conceito de centralização com balanceamento. Nestes dados observa-se valores mais altos que os observados na Tabela 4, que corresponde somente ao conceito de centralização. Porém, com o uso também do balanceamento, é possível observar um aumento nestes valores.

Tabela 5 – KTA Centralizado Balanceado - Alinhamento x Acurácia.

Base	Alinhamento		Acurácia		
	Média	Maior	Média	Maior Al.	Maior Ac.
Wine	0,10696650	0,191747	65,81	89,10	89,10
Ionosphere	0,14773093	0,264996	92,29	95,05	95,05
Bupa	0,05178693	0,070943	62,85	70,04	99,56
Toy	0,00903405	0,036793	59,84	67,00	67,00

Fonte: Elaborado pelo autor (2019).

A Tabela 6 descreve os valores do FSM com os dados gerados considerando tanto

a centralização quanto a não centralização das matrizes *kernel*, que obtiveram os mesmos valores. Para o caso específico do FSM, os melhores valores são os mais baixos para o alinhamento. Portanto, a análise aponta que entre as bases testadas, apenas duas bases apresentam valores realmente baixos. Além de, também, terem sido observada quedas nas acurácias do classificador, correspondentes aos alinhamentos.

Tabela 6 – FSM - Alinhamento x Acurácia.

Base	Alinhamento		Acurácia		
	Média	Menor	Média	Menor Al.	Maior Ac.
Wine	1,09032909	0,000001	65,81	73,44	89,10
Ionosphere	16,16025118	0,088604	92,29	95,05	95,05
Bupa	70,51362621	0,236279	62,85	61,76	99,56
Toy	1,09032909	0,000003	59,84	56,50	67,00

Fonte: Elaborado pelo autor (2019).

É possível notar que os dados gerados para as bases com e sem balanceamento são sempre diferentes. Isto é explicado considerando que as bases testadas não são balanceadas, logo é realizado um balanceamento para cada uma. No KTA utilizando o balanceamento, foi possível observar resultados melhores que o teste realizado sem o balanceamento e, na sequência, o modelo com centralização e balanceamento apresentou os melhores resultados, indicados na Tabela 5, representando o conjunto de melhor relação entre os testes executados. Portanto, a medida de similaridade escolhida para os experimentos de seleção de modelos foi o KTA Centralizado Balanceado.

5.4 SELEÇÃO DE MODELOS

Nesta seção, a estrategia de seleção de modelos em duas etapas é avaliada, principalmente em relação ao uso do *Simulated Anneling* como otimizador. Inicialmente, a etapa de seleção de hiperparâmetros é realizada visando analisar seu desempenho.

5.4.1 **Seleção de Hiperparâmetros**

Nesta subseção são descritos os testes realizados com o *Simulated Anneling* (SA) para a busca de hiperparâmetros utilizando o KTA com balanceamento e com centralização da matriz *kernel*.

Os parâmetros utilizados foram:

- *Simulated Anneling*:
  - Temperatura inicial: 0,01;
  - Temperatura mínima: 0,000001;

  - Decréscimo da temperatura: 0,9;
  - Iterações por temperatura: 100.

- *Kernel* PUK:
  - Valor mínimo para σ: 0,000005;
  - Valor máximo para σ: 10000;
  - Valor inicial para σ: número real aleatório entre valores mínimo e máximo;
  - Valor mínimo para ω: 0,05;
  - Valor máximo para ω: 10000;
  - Valor inicial para ω: número real aleatório entre valores mínimo e máximo.

A cada iteração do algoritmo são realizados os seguintes passos:

1. Sorteio de um número real aleatório entre 0 e 100 – *gap*;
2. Sorteio de um número inteiro aleatório entre 0, 1, 2 e 3 – s:
   - Se $s = 0 \rightarrow \sigma = \sigma + gap$;
   - Se $s = 1 \rightarrow \sigma = \sigma$ - gap;
   - Se $s = 2 \rightarrow \omega = \omega + gap$;
   - Se $s = 3 \rightarrow \omega = \omega$ - gap;

A Tabela 7 apresenta os resultados dos cinco maiores valores de alinhamento, de soluções consideradas distintas, para cada base de dados. Visando a obtenção de soluções distintas em relação ao nível de alinhamento no que tange aos padrões dos parâmetros σ e ω do kernel PUK, adotou-se a estratégia de considerar soluções diferentes somente as que apresentem uma distância, para cada parâmetro, acima de um valor predefinido, no caso 5 unidades (com a exceção de valores menores que 1, que podem diferir na casa decimal). Além dos resultados dos valores de alinhamento, com seus parâmetros, a tabela apresenta a acurácia da respectiva solução usando o SVM.

Após essa primeira etapa, o maior valor de alinhamento serve como base para a utilização na etapa de seleção de características.

5.4.2 **Seleção de Características**

Nesta subseção são descritos os parâmetros utilizados no SA para a seleção de características, após a obtenção do conjunto ótimo de parâmetros da função PUK. Também são apresentados os resultados para cada base, considerando as 4 melhores entropias e a solução inicial com o conjunto total de características.

Os parâmetros utilizados foram:

Tabela 7 – Dados resultantes da seleção de hiperparâmetros.

Base	Alinhamento	σ	ω	Acurácia
Wine	0,35235611	2362,177099	2,265339	95,10 ± 0,82
	0,35228966	2162,050238	7,055884	94,93 ± 0,90
	0,35227232	1951,037538	63,810491	95,10 ± 0,74
	0,35227205	1979,984344	94,910988	95,05 ± 0,85
	0,35226926	1967,906735	9598,110904	94,93 ± 0,81
Sonar	0,11638784	1,355022	4159,266945	88,17 ± 1,03
	0,11073577	75,887331	0,061042	86,69 ± 0,76
	0,07136713	12,106693	6370,076601	86,00 ± 1,65
	0,07007619	34,253975	6124,579608	78,79 ± 1,78
	0,06999024	46,708579	6027,198096	76,38 ± 0,75
Ionospehere	0,26528329	3,717154	9985,451827	93,77 ± 0,59
	0,24137404	3892,254402	0,043896	95,42 ± 0,41
	0,18522846	10,666219	8516,458632	91,62 ± 1,81
	0,14498149	24,109622	8115,750603	91,09 ± 1,40
	0,13693629	41,233558	8441,779839	92,34 ± 0,61
Bupa	0,07094269	9,752500	0,543138	68,42 ± 1,20
	0,06838642	82,652272	0,050000	71,29 ± 0,86
	0,06360160	16,104622	9196,029542	63,07 ± 0,14
	0,06219579	22,247998	2419,818720	61,76 ± 1,48
	0,05649790	36,661885	6177,462081	63,19 ± 3,76
Synthetic	0,63573587	305,459766	5,075233	99,88 ± 0,08
	0,63565395	281,568865	61,168254	99,88 ± 0,08
	0,63564744	276,274911	8969,957579	99,88 ± 0,08
	0,63546650	293,569750	9719,998779	99,85 ± 0,05
	0,63312978	355,998413	9631,104465	99,83 ± 0,00
LP4	0,64919222	125,074007	9675,740837	98,03 ± 0,42
	0,64826201	130,729087	4052,287362	98,03 ± 0,42
	0,64751893	117,795343	3874,312571	98,03 ± 0,42
	0,63244139	152,781762	4452,265999	98,11 ± 0,37
	0,61557705	167,418439	4414,136174	98,11 ± 0,37

Fonte: Elaborado pelo autor (2019).

- SA:
  - Temperatura inicial: 1;
  - Temperatura mínima: 0,0001;
  - Decréscimo da temperatura: 0,9;
  - Iterações por temperatura: 200.
- Função objetivo – Equação (4.4):

  - Peso do valor do alinhamento: 0,8;
  - Peso da cardinalidade: 0,2 * alinhamento inicial (todas as características).

  A cada iteração:

- Sorteio de um número inteiro aleatório entre 1 e d (total de características) – s:
  - Se a característica s está ativada, a mesma é desativada;
  - Se a característica s está desativada, a mesma é ativada.

A Tabela 8 apresenta os resultados com os quatro maiores valores da função objetivo (entropia), além da solução original com todas as características, sendo um único resultado para conjuntos de mesma cardinalidade e considerando no mínimo 2 características por solução. Além disso, a tabela apresenta, também, a respectiva acurácia da solução SVM.

Cada valor de alinhamento e acurácia está representado com o total de características F encontradas pelo modelo. Na prática, quando se remove características em um modelo de seleção, pode-se piorar os resultados, uma vez que há o risco de remoção de alguma informação útil.

Tem-se a expectativa que a seleção de características reflita em um aumento no valor do alinhamento, porém uma consequência do uso de menos características é a possibilidade de diminuição da acurácia do classificador, visto que não a mesma foi utilizada diretamente na otimização. Nas bases testadas, observa-se que, de fato, as suposições iniciais ocorreram, tendo um aumento ao decrementar o número de características. Porém, observa-se, também, que os melhores alinhamentos encontrados estão garantindo um bom valor de acurácia. Nas bases Synthetic e LP4 tem-se quase 100% de acurácia mesmo com a diminuição das características, que aumentam o valor do alinhamento.

Observando a base Wine, foi encontrado um mínimo de 2 características associadas à menor acurácia dos 5 melhores alinhamentos, que já é uma acurácia alta, porém é observado iterações onde até 6 características são encontradas, associadas a valores muito altos do desempenho do classificador.

Com a base Sonar pode-se observar padrões semelhantes à base Wine, a qual, ao decrementar o número de características, ocorre um aumento do alinhamento e uma diminuição na acurácia.

A base Bupa manteve um equilíbrio ao reduzir as características, mesmo aumentando o alinhamento consequente da diminuição de características, a acurácia do classificador se manteve oscilante, porém de forma estável.

Tabela 8 – Dados resultantes da seleção de características

Base	Entropia	Alinhamento	F	Acurácia
Wine	0,28188489	0,35235611	13	95,10 ± 0,82
	0,48964633	0,54429713	3	82,64 ± 0,49
	0,48812909	0,53562450	2	79,76 ± 0,86
	0,46944174	0,52581746	4	90,79 ± 0,49
	0,45643600	0,52311244	6	93,66 ± 0,27
Sonar	0,09311027	0,11638784	60	88,17 ± 1,03
	0,21455072	0,24297103	8	69,13 ± 1,72
	0,21339807	0,24298507	11	69,98 ± 1,40
	0,21313420	0,24314019	12	70,39 ± 1,79
	0,21297859	0,24343062	13	69,95 ± 1,73
Ionosphere	0,21222663	0,26528329	34	93,77 ± 0,59
	0,42966546	0,48246468	6	78,32 ± 1,83
	0,41626659	0,46961732	8	84,67 ± 1,44
	0,41356733	0,46819385	9	83,02 ± 1,41
	0,40787128	0,45717257	7	85,93 ± 1,26
Bupa	0,05675415	0,07094269	13	68,42 ± 1,20
	0,06797464	0,07610047	3	66,23 ± 1,40
	0,06765504	0,07865691	4	65,91 ± 1,12
	0,06581152	0,07930845	5	68,20 ± 1,38
	0,06520745	0,06968553	2	61,39 ± 2,15
Synthetic	0,50858870	0,63573587	60	99,88 ± 0,08
	0,65683847	0,68065642	7	99,27 ± 0,09
	0,65681470	0,68327560	8	99,10 ± 0,14
	0,65544642	0,68421415	9	98,98 ± 0,34
	0,65415932	0,68525417	10	99,33 ± 0,24
LP4	0,51935377	0,64919222	90	98,03 ± 0,42
	0,61731148	0,66344065	30	97,06 ± 0,44
	0,61728035	0,66159842	29	97,06 ± 0,44
	0,61722242	0,66513263	31	96,89 ± 0,45
	0,61640756	0,65870413	28	96,55 ± 0,06

Fonte: Elaborado pelo autor (2019).

6 CONSIDERAÇÕES FINAIS

Esta pesquisa propôs, como objetivo geral, o desenvolvimento de um método de seleção de modelos baseado em medidas de similaridade separado em duas etapas: seleção de hiperparâmetros e seleção de características. O objetivo deste desenvolvimento é, além de superar as limitações dos métodos trabalhados na literatura para seleção de modelos que possuem custo computacional muito alto, gerar valores ótimos de parâmetros para a realização de uma busca por características, utilizando apenas os valores dos alinhamentos gerados pelas medidas de similaridade. Numa primeira etapa, foi estudada a função *kernel* PUK, que funciona como uma função universal podendo, através do ajuste de seus parâmetros, assumir o comportamento de outras funções *kernel*. Por esta característica, a função PUK foi escolhida para a geração das matrizes *kernel*, diminuindo então, o tempo de teste que seria gasto executando diferentes funções *kernel* separadamente.

Numa segunda instância, foram estudados os comportamentos de duas medidas de similaridade, bem como suas variantes, com o objetivo de escolher qual medida seria utilizada para acoplar ao modelo de seleção. Os resultados iniciais apresentados concluíram que a medida de seleção KTA, com centralização e balanceamento, possui as características de manter um valor alto de alinhamento correlacionado a um valor alto de acurácia do classificador. Com esta observação, o método KTA foi utilizado no modelo de seleção, na primeira e segunda etapas para a busca dos melhores alinhamentos, dando sequência para a busca dos melhores subconjuntos de características.

Com os testes realizados com o KTA, pode se dizer que é um método de similaridade suficiente para a seleção de parâmetros e características, uma vez que um conjunto de altos valores obtidos pelo KTA está relacionado a um alto valor de acurácia do SVM. Um dos problemas de o alinhamento ser suficiente é que, apesar de poder aumentar seu valor ao retirar algumas características, pode ocorrer uma diminuição da acurácia. Em outras palavras, nem todo valor ótimo de KTA está relacionado a valores ótimos de acurácia, mas um conjunto dos melhores alinhamentos pode garantir um bom desempenho do classificador.

Dentre as contribuições deste trabalho, atenta-se para a disponibilidade que os testes demonstraram para a utilização do KTA na busca de um conjunto de parâmetros suficientes para a classificação pelo SVM. Pode-se destacar também que o KTA vem sendo utilizado no campo de aprendizado para predição de uma função *kernel* para o contexto de classificação de dados, sendo interessante a ideia da sua utilização como forma alternativa a outras funções nesta área, como a busca por parâmetros ótimos para a classificação supervisionada, utilizada neste trabalho.

Como foi observado, a medida de similaridade testada na construção dos modelos de seleção é uma medida que garante bons resultados para a acurácia do classificador,

selecionando características que asseguram um bom desempenho, porém também foi visto que nem todo bom resultado do KTA está correlacionado a valores ótimos. Portanto, é de valia para pesquisas futuras, um estudo aprofundado do KTA para um maior entendimento da medida de similaridade e suas variantes, para verificar uma integração mais aderente a outros algoritmos baseados em *kernel*. Também é interessante analisar possíveis soluções referentes a diminuição da acurácia em relação ao aumento do alinhamento, identificar comportamentos do KTA na integração ao classificador para obter respostas de algum refinamento do processo para melhores resultados.

O trabalho proposto foi todo montado em etapas, incluindo a obtenção das acurácias do classificador que são associadas aos alinhamentos na seleção de parâmetros e características, como também a obtenção dos resultados de cada solução separadamente. Portanto, é interessante pensar como desenvolvimento futuro, um modelo de seleção embutido, unificando as etapas de seleção de parâmetros e seleção de características, utilizando como referência as medidas de alinhamento.

REFERÊNCIAS

ABAKAR, Khalid A. A.; YU, ChongWen. Performance of svm based on puk kernel in comparison to svm based on rbf kernel in prediction of yarn tenacity. **Indian Journal of Fibre and Textile Research**, v. 39, p. 55–59, 2014.

ARAUJO, Haroldo Alexandre. **Algoritmo Simulated Annealing: uma nova abordagem**. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Santa Catarina, Florianópolis, 2001.

BACHE, K.; LICHMAN, M. **UCI Machine Learning Repository**. 2013. Disponível em: <http://archive.ics.uci.edu/ml>.

BONESSO, Diego. **Estimação dos Parâmetros do Kernel em um Classificador SVM na Classificação de Imagens Hiperespectrais em uma Abordagem Multiclasse**. Dissertação (Mestrado em Sensoriamento Remoto) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

BOSER, Bernhard E; GUYON, Isabelle M; VAPNIK, Vladimir N. A training algorithm for optimal margin classifiers. In: ACM. **Proceedings of the fifth annual workshop on Computational learning theory**. [S.l.], 1992. p. 144–152.

CORTES, Corinna; MOHRI, Mehryar; ROSTAMIZADEH, Afshin. Algorithms for learning kernels based on centered alignment. **Journal of Machine Learning Research**, v. 13, n. Mar, p. 795–828, 2012.

CRISTIANINI, Nello; SHAWE-TAYLOR, John; ELISSEEFF, Andre; KANDOLA, Jaz S. On kernel-target alignment. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2002. p. 367–373.

DASH, Manoranjan; LIU, Huan. Feature selection for classification. **Intelligent data analysis**, Elsevier, v. 1, n. 1-4, p. 131–156, 1997.

DING, Shifei; ZHU, Hong; JIA, Weikuan; SU, Chunyang. A survey on feature extraction for pattern recognition. **Artificial Intelligence Review**, Springer, v. 37, n. 3, p. 169–180, 2012.

GUYON, Isabelle; ELISSEEFF, André. An introduction to variable and feature selection. **Journal of machine learning research**, v. 3, p. 1157–1182, 2003.

JAIN, Anil K; DUIN, Robert P. W.; MAO, Jianchang. Statistical pattern recognition: A review. **IEEE Transactions on pattern analysis and machine intelligence**, Ieee, v. 22, n. 1, p. 4–37, 2000.

KUO, Bor-Chen; HO, Hsin-Hua; LI, Cheng-Hsuan; HUNG, Chih-Cheng; TAUR, Jin-Shiuh. A kernel-based feature selection method for svm with rbf kernel for hyperspectral image classification. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, IEEE, v. 7, n. 1, p. 317–326, 2013.

MARSLAND, Stephen. **Machine learning: an algorithmic perspective**. [S.l.]: CRC press, 2014.

MOAKHER, Maher. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. **SIAM Journal on Matrix Analysis and Applications**, SIAM, v. 26, n. 3, p. 735–747, 2005.

MÜLLER, Klaus-Robert; MIKA, Sebastian; RÄTSCH, Gunnar; TSUDA, Koji; SCHÖLKOPF, Bernhard. An introduction to kernel-based learning algorithms. **IEEE transactions on neural networks**, v. 12, n. 2, 2001.

NGUYEN, Canh Hao; HO, Tu Bao. Kernel matrix evaluation. In: **International Joint Conferences on Artificial Intelligence**. [S.l.: s.n.], 2007. p. 987–992.

ONG, Chorng-Shyong; HUANG, Jih-Jeng; TZENG, Gwo-Hshiung. Model identification of arima family using genetic algorithms. **Applied Mathematics and Computation**, Elsevier, v. 164, n. 3, p. 885–912, 2005.

PEROLINI, A. Genetic algorithms and kernel matrix-based criteria combined approach to perform feature and model selection for support vector machines. **World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering**, v. 4, n. 4, p. 625–634, 2010.

RAMONA, Mathieu; RICHARD, Gaël; DAVID, Bertrand. Multiclass feature selection with kernel gram-matrix-based criteria. **IEEE transactions on neural networks and learning systems**, IEEE, v. 23, n. 10, p. 1611–1623, 2012.

REN, Yuan; BAI, Guangchen. Determination of optimal svm parameters by using ga/pso. **JCP**, v. 5, n. 8, p. 1160–1168, 2010.

SCHÖLKOPF, Bernhard; SMOLA, Alexander J; BACH, Francis et al. **Learning with kernels: support vector machines, regularization, optimization, and beyond**. [S.l.]: MIT press, 2002.

STAELIN, Carl. Parameter selection for support vector machines. **Hewlett-Packard Company, Tech. Rep. HPL-2002-354R1**, 2003.

ÜSTÜN, Bülent; MELSSEN, Willem J; BUYDENS, Lutgarde MC. Facilitating the application of support vector regression by using a universal pearson vii function based kernel. **Chemometrics and Intelligent Laboratory Systems**, Elsevier, v. 81, n. 1, p. 29–40, 2006.

WANG, Lipo. **Support vector machines: theory and applications**. [S.l.]: Springer Science & Business Media, 2005. v. 177.

WESTON, Jason; CHAPELLE, Olivier; ELISSEEFF, Andre; SCHOLKOPF, Bernhard; VAPNIK, Vladimir. Kernel dependency estimation. In: CITESEER. **in Advances in NIPS 15**. [S.l.], 2003.

YOU, Di. **Model Selection in Kernel Methods**. Tese (Doctor of Philosophy in Electrical and Computer Engineering) — The Ohio State University, Columbus, 2011.