

Trabalho Final de Introdução ao Reconhecimento de Padrões

Gabriel de Araújo Costa Gomes
2022102872
gacg@ufmg.br

Lucas Ribeiro da Silva
2022055564
lucarsilvak@ufmg.br

I. INTRODUÇÃO

Os problemas de Reconhecimento de Padrões podem ser problemas muito comuns nas áreas de engenharia e ciências da computação. Para lidar com esses problemas foram desenvolvidos ao longo do tempo diversos algoritmos e métodos de aprendizado de máquina, cada um com suas peculiaridades em implementações e objetivos.

Este relatório estuda os problemas de classificação binária tendo em vista um classificador **Support Vector Machine** (SVM), um algoritmo que procura encontrar um hiperplano ótimo para separar os dados com a maior margem possível. Em casos de dados que não são linearmente separáveis, há a necessidade de adaptar o SVM para tornar o problema factível. Essa adaptação muitas vezes inclui o **truque do kernel**, que consiste em mapear os dados para um espaço de dimensão mais alta, onde a separação pode tornar-se linear. A utilização do truque do kernel, entretanto, exige a definição de **hiperparâmetros** que tornam o kernel sensível a escolha destes parâmetros. O objetivo desse trabalho é fornecer uma abordagem não-supervisionada para manipular os hiperparâmetros C ou γ da função de kernel do SVM e encontrar um valor para essas constantes que aproxime o classificador do máximo da acurácia evitando os malefícios do **overfitting** e **underfitting** dentro do **espaço de verossimilhanças**, matematicamente definido como:

Sejam C_0 e C_1 os conjuntos das amostras das classes 0 e 1, respectivamente.

Para cada amostra i , definimos as afinidades médias com cada classe como:

$$p_{i0} = \frac{1}{|C_0|} \sum_{j \in C_0} K_{ij}$$

$$p_{i1} = \frac{1}{|C_1|} \sum_{j \in C_1} K_{ij}$$

Onde K_{ij} é o kernel. O vetor de verossimilhança da amostra i é:

$$\mathbf{p}_i = \begin{bmatrix} p_{i0} \\ p_{i1} \end{bmatrix}$$

II. REVISÃO BIBLIOGRÁFICA

Nesta seção, revisamos brevemente alguns artigos que tratam sobre separação de clusters, inclusive no espaço de verossimilhanças em contextos não-supervisionados.

A. Hebbian Learning with Kernel-Based Embedding of Input Data

O artigo de Ushikoshi et al. [1] propõe a projeção dos dados em um **espaço de verossimilhança** por meio de **Kernel Density Estimators** (KDE), visando melhorar a separabilidade linear entre as classes. Para quantificar essa separação, os autores definem uma **métrica combinada** calculada como o produto da **distância euclidiana** entre os vetores médios de verossimilhança de cada classe (V_1 e V_2) e o **cosseno do ângulo** entre esses vetores:

$$\text{Métrica} = \|V_1 - V_2\| \times \cos \theta$$

onde $\cos \theta$ é dado por:

$$\cos \theta = \frac{V_1 \cdot V_2}{\|V_1\| \|V_2\|}$$

Essa métrica será utilizada e combinada com outras no presente relatório para auxiliar na avaliação na escolha dos parâmetros do kernel.

B. Estimating the Number of Clusters in a Data Set via the Gap Statistic

O trabalho de Tibshirani et al. [2] introduz a **Gap Statistic**, uma métrica não supervisionada para estimar o número ideal de agrupamentos em um conjunto de dados. A ideia central é comparar a **dispersão intra-cluster** observada em uma partição k -means com a esperada sob uma distribuição de referência (tipicamente uniforme), por meio da fórmula:

$$\text{Gap}(k) = E^* [\log(W_k^*)] - \log(W_k)$$

onde W_k representa a soma das distâncias quadradas intra-cluster, e a expectativa E^* é estimada por simulações sobre dados gerados aleatoriamente com a mesma distribuição geral dos dados reais.

Esse critério permite identificar o valor de k para o qual a estrutura de clusters é estatisticamente mais significativa em relação ao acaso. Embora originalmente desenvolvido para escolha de k , seu princípio de comparar estruturas observadas com uma referência aleatória é relevante para nosso trabalho,

especialmente na análise da separabilidade de classes no espaço de verossimilhança. Assim, elementos da lógica da Gap Statistic inspiram parte da metodologia usada na seleção dos parâmetros do kernel sigmoid.

III. METODOLOGIA

O código foi desenvolvido na linguagem Python, versão 3.11. Os dados utilizados no experimento foram provenientes principalmente da biblioteca `scikit-learn`. O processo completo para cada dataset seguiu os seguintes passos:

- 1) Normalização e tratamento dos dados via `StandardScaler`.
- 2) Separação dos conjuntos em treinamento e teste.
- 3) Definição da faixa de h observada e iteração para cada valor de h :
 - Cálculo do kernel.
 - Treinamento de um SVM com kernel pré-computado.
 - Avaliação da acurácia no treino e teste.
 - Cálculo do espaço de verossimilhança.
 - Avaliação por métricas não supervisionadas.
- 4) Escolha do melhor h pela métrica definida.
- 5) Reavaliação do SVM no melhor h e melhor bias (pro caso do Sigmoid).

Os datasets utilizados foram os datasets multidimensionais da tabela abaixo:

| Dataset | Amostras | Features |
|---------------|----------|----------|
| breast_cancer | 569 | 30 |
| digits | 1797 | 64 |
| wine | 178 | 13 |
| iris | 150 | 4 |
| diabetes | 442 | 10 |
| spambase | 4601 | 57 |

TABLE I

RESUMO DE DATASETS DA BIBLIOTECA `SCIKIT-LEARN`.

IV. IMPLEMENTAÇÃO

A. Tratamento dos Dados

Para tratamento dos dados, foi utilizada a separação dos dados com **validação cruzada** utilizando-se de conjuntos de treinamento e teste com razão de 70% e 30% respectivamente, e os dados foram configurados de maneira que mesmo datasets que não possuam classificação binária fossem configurados como problemas binários, no caso de `digits` por exemplo, o dígito ser ≥ 5 , enquanto em outras classes o método usado foi comparar uma classe com as demais.

B. SVM e Kernel

O SVM é um método de classificação que busca encontrar um hiperplano ótimo para separar classes de dados. Para dados linearmente separáveis, o problema primal é formulado como:

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \quad (1)$$

sujeito a:

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \quad (2)$$

onde:

- \mathbf{w} é o vetor normal ao hiperplano
- b é o termo de viés (bias)
- C controla o trade-off entre margem e erros
- ξ_i são variáveis de folga

Para problemas não-lineares, o SVM utiliza uma função ϕ para mapear os dados para um espaço de maior dimensão:

$$\mathbf{x}_i \mapsto \phi(\mathbf{x}_i) \quad (3)$$

e as restrições se tornam:

$$y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \quad (4)$$

A solução ótima tem a forma $\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \phi(\mathbf{x}_i)$, levando a funções de decisão que dependem apenas de produtos internos $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$. Em que ao invés de calcular explicitamente $\phi(\mathbf{x}_i)$, definimos uma função de kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \quad (5)$$

Onde a **matriz de kernel** \mathbf{K} com elementos $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. E o problema dual resultante, que depende apenas da matriz \mathbf{K} , é resolvido como:

$$\max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (6)$$

sujeito a:

$$\sum_{i=1}^N \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad \forall i \quad (7)$$

1) **Kernel RBF**: O **Kernel RBF** (Radial Basis Function) é uma das funções de kernel avaliadas neste relatório e é definido como:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

onde γ controla a influência de cada amostra de treino. Neste trabalho, usamos a seguinte relação entre γ e o parâmetro h , que representa a banda de largura do kernel:

$$\gamma = \frac{1}{h^2}$$

Além de h , o SVM também depende do hiperparâmetro de regularização C , que controla o trade-off entre maximizar a margem e minimizar o erro de classificação no conjunto de treino. Neste trabalho, C foi mantido fixo (em $C = 1$), enquanto o parâmetro h foi ajustado dentro de um intervalo que varia de 0,05 a 10, com passos incrementais de 0,05.

2) *Kernel Sigmoid*: O **Kernel Sigmoid**, também conhecido como kernel sigmoidal ou kernel de função tangente hiperbólica, é inspirado nos neurônios artificiais e é definido pela seguinte expressão:

$$K(x_i, x_j) = \tanh(\gamma x_i^\top x_j + \text{coef}_0)$$

Neste caso, γ controla a inclinação da função tangente hiperbólica, enquanto coef_0 atua como um deslocamento (bias) que influencia o centramento da curva do kernel. Assim como no kernel RBF, adotamos a relação:

$$\gamma = \frac{1}{h^2}$$

Em nossos experimentos, fixamos h (e portanto γ) em um valor determinado a partir de observações preliminares, e variamos coef_0 em um intervalo de $[-5, 5]$, com o objetivo de investigar seu impacto na separabilidade dos dados no espaço de verossimilhança.

O kernel sigmoid é menos utilizado que o RBF em muitas aplicações práticas, pois pode não satisfazer todas as condições de positividade definidas (PSD) exigidas por certos teoremas. Ainda assim, sua interpretação como um neurônio com função de ativação o torna interessante em contextos específicos.

Para ambos os kernels, a escolha dos parâmetros foi orientada por uma métrica combinada que leva em conta a separação angular entre classes e a métrica de gap statistic no espaço projetado.

C. Métricas

Na análise de agrupamento e separabilidade entre classes em um espaço de características, a escolha de métricas adequadas é essencial para avaliar a qualidade da separação. Duas métricas utilizadas incluem a **Gap Statistic** e uma **Métrica Combinada baseada em distância e ângulo de centroides**, além de apresentar uma metodologia de combinação e normalização dessas métricas para seleção de parâmetros.

1) *Gap Statistic*: Proposta por Tibshirani, Walther e Hastie [2], é uma abordagem amplamente utilizada para estimar o número ideal de clusters em uma base de dados de forma não supervisionada. A ideia central é comparar a *dispersão intra-cluster* observada com a esperada sob uma distribuição nula de referência (tipicamente uniforme), quantificando o desvio entre elas.

Seja W_k a medida de dispersão dentro dos clusters quando os dados são particionados em k grupos (por exemplo, a soma dos quadrados das distâncias intra-cluster). O valor da Gap Statistic é definido como:

$$\text{Gap}(k) = E^*[\log(W_k^*)] - \log(W_k)$$

onde $E^*[\log(W_k^*)]$ representa a expectativa da dispersão sob a distribuição de referência (obtida por simulações Monte Carlo) e W_k é a dispersão real nos dados.

A intuição é que, se os dados possuem uma estrutura de cluster bem definida, o valor observado de $\log(W_k)$ será substancialmente menor que o valor esperado sob o modelo

nulo. O número ótimo de clusters k^* é escolhido como o menor valor de k tal que:

$$\text{Gap}(k) \geq \text{Gap}(k+1) - s_{k+1}$$

onde s_{k+1} é o desvio padrão estimado da simulação com $k+1$ clusters, ajustado com correção para o número de amostras de referência B :

$$s_k = \sqrt{1 + \frac{1}{B}} \cdot \text{sd}_k$$

Esse critério é conhecido como a “regra do erro-padrão” e é inspirado em práticas semelhantes em poda de árvores de decisão.

Vantagens:

- É aplicável a qualquer algoritmo de clusterização e medida de dissimilaridade.
- Leva em conta o comportamento esperado do modelo nulo.
- É eficaz mesmo para detectar a ausência de estrutura de cluster (i.e., $k = 1$).

2) *Métrica Combinada Baseada em Centroides*: Considerando agora um cenário supervisionado, onde há rótulos binários ($y_i \in \{0, 1\}$), definimos os **centroides** das duas classes no espaço de verossimilhança (\mathbf{p}_i) como:

$$\mathbf{V}_1 = \frac{1}{|\mathcal{I}_0|} \sum_{i \in \mathcal{I}_0} \mathbf{p}_i, \quad \mathbf{V}_2 = \frac{1}{|\mathcal{I}_1|} \sum_{i \in \mathcal{I}_1} \mathbf{p}_i$$

onde:

$$\mathcal{I}_0 = \{i \mid y_i = 0\}, \quad \mathcal{I}_1 = \{i \mid y_i = 1\}$$

As seguintes quantidades são então calculadas:

- **Distância Euclidiana entre os centroides** (Δ):

$$\Delta = \|\mathbf{V}_1 - \mathbf{V}_2\|$$

Essa distância mede a separação absoluta entre as classes no espaço de saída.

- **Cosseno do ângulo entre os centroides**:

$$\cos \theta = \begin{cases} 0, & \text{se } \|\mathbf{V}_1\| = 0 \text{ ou } \|\mathbf{V}_2\| = 0 \\ \frac{\mathbf{V}_1^\top \mathbf{V}_2}{\|\mathbf{V}_1\| \|\mathbf{V}_2\|}, & \text{caso contrário} \end{cases}$$

Isso representa a orientação relativa dos vetores centroides. Valores próximos de 1 indicam alinhamento, enquanto valores próximos de 0 indicam ortogonalidade.

- **Métrica Combinada Final**:

$$U = \Delta \cdot \cos \theta$$

3) *Normalização das Métricas*: Como as escalas de G_h e U_h são distintas, realizamos uma normalização min-max para o intervalo $[0, 1]$, calculada para cada valor testado do parâmetro h :

$$\tilde{S}_h = \frac{G_h - \min_h G_h}{\max_h G_h - \min_h G_h}$$

$$\tilde{U}_h = \frac{U_h - \min_h U_h}{\max_h U_h - \min_h U_h}$$

Esta transformação permite uma comparação justa entre métricas com escalas diferentes.

4) *Combinação das Métricas*: Por fim, definimos uma métrica de decisão combinada G_h , ponderando a importância relativa das duas métricas:

$$T_h = \alpha \tilde{G}_h + (1 - \alpha) \tilde{U}_h$$

No nosso caso, foi adotado $\alpha = 0.75$, definido experimentalmente, privilegiando a contribuição da Gap Statistic em 75%.

Esta combinação busca equilibrar:

- A **distância intracluster** (via Gap Statistic, não supervisionada), buscando o ponto de **máxima acurácia**;
- E a **separabilidade** entre classes (via métrica combinada de centroides, supervisionada, mas utilizada via classificação por K-Médias, então não-supervisionada), evitando o **overfitting** e **underfitting**.

V. EXPERIMENTOS

A. Kernel RBF

Para os experimentos com o Kernel RBF, utilizou-se o **classificador SVM** para fazer a subsequente projeção no **espaço de verossimilhanças** e então, a partir dessa projeção, utiliza-se o **K-Médias** com $K = 2$ para separar o espaço de verossimilhanças em 2 clusters. Então a partir dos clusters previstos pelo K-Médias é calculada tanto o **Gap Statistic** quanto a **métrica apresentada por Ushikoshi et al.** Depois, as métricas não-supervisionadas serão comparadas com a acurácia do modelo para definir o melhor h , por meio de um gráfico 2D.

B. Kernel Sigmoid

Para além dos passos feitos com o Kernel RBF, feitos semelhantemente para o kernel sigmoid, foi utilizada também a variação do $coef_0$ entre -1 e 1, além da variação de γ entre 0 e 10, com passo 0,05. Depois, será possível comparar os dados por meio de um grid.

VI. RESULTADOS

Para os datasets mencionados anteriormente foi rodado cada algoritmo com uma seed definida e gerados os gráficos que relacionam a métrica utilizada a acurácia.

A. Breast Cancer

Para o dataset Breast Cancer, foram obtidos os seguintes resultados:

1) *RBF*: Para a RBF, o h que maximiza o valor da métrica foi de $h = 7.90$, o gráfico que demonstra a acurácia de teste supervisionada em relação a métrica não supervisionada foi:

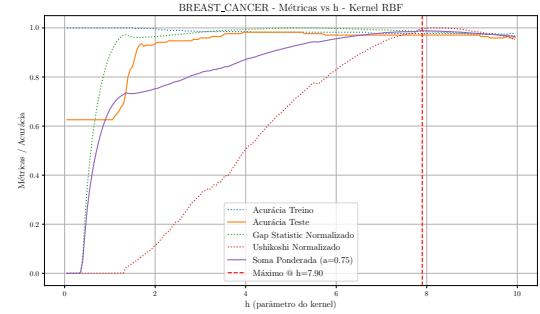


Fig. 1. MÉTRICA x ACURÁCIA

que por sua vez, tem o seguinte espaço de verossimilhança.

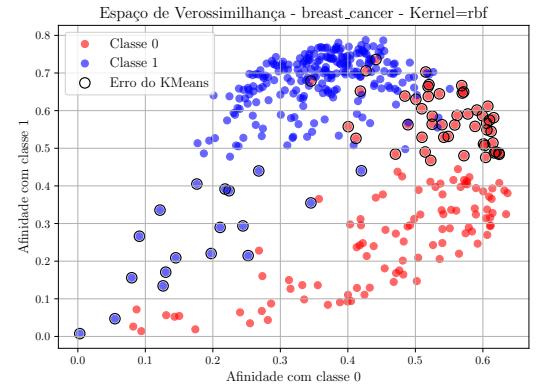


Fig. 2. ESPAÇO DE VEROSSIMILHANÇAS MÁXIMO

2) *Sigmoid*: Os coeficientes obtidos γ e $coef_0$ para o Sigmoid foram:

TABLE II
MELHOR COMBINAÇÃO DE PARÂMETROS PARA O KERNEL SIGMOID

| Parâmetro | Valor |
|--------------------|---------|
| Gamma (γ) | 0.0037 |
| Bias ($Coef_0$) | -1.0000 |
| Métrica (T_h) | 0.9441 |
| Acurácia (Treino) | 0.9422 |
| Acurácia (Teste) | 0.9532 |

A correlação entre a Métrica (T_h) e a Acurácia pode ser visualizada razoavelmente nos grids abaixo:

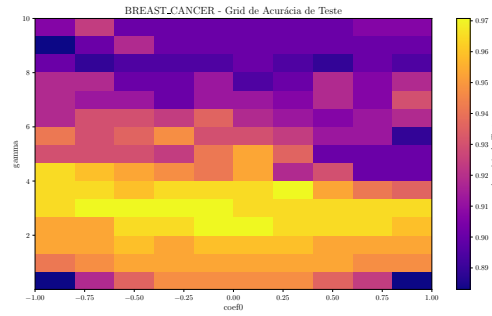


Fig. 3. GRID DE ACURÁCIA

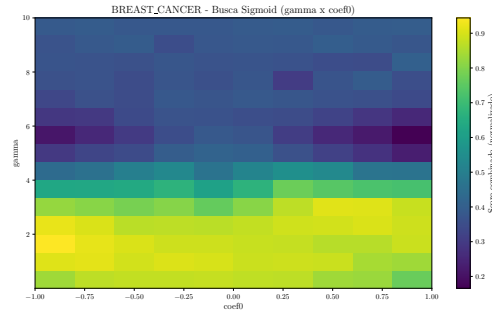


Fig. 4. GRID DA MÉTRICA UTILIZADA

Onde pode-se observar o platô de coloração mais brilhante no canto inferior da imagem.

B. Digits

Para o dataset Digits, foram obtidos os seguintes resultados:

1) *RBF*: Para a RBF, o h que maximiza o valor da métrica foi de $h = 9.95$, o gráfico que demonstra a acurácia de teste supervisionada em relação a métrica não supervisionada foi:

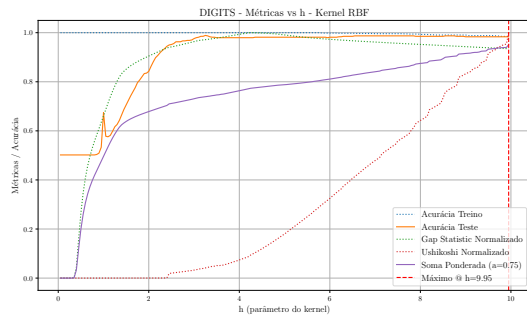


Fig. 5. MÉTRICA x ACURÁCIA

que por sua vez, tem o seguinte espaço de verossimilhança.

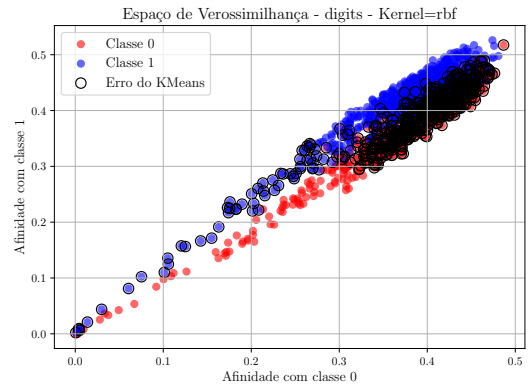


Fig. 6. ESPAÇO DE VEROSSIMILHANÇAS MÁXIMO

2) *Sigmoid*: Os coeficientes obtidos γ e $coef_0$ para o Sigmoid foram:

TABLE III
MELHOR COMBINAÇÃO DE PARÂMETROS PARA O KERNEL SIGMOID

| Parâmetro | Valor |
|--------------------|---------|
| Gamma (γ) | 0.0037 |
| Bias ($Coef_0$) | -1.0000 |
| Métrica (T_h) | 0.8882 |
| Acurácia (Treino) | 0.9236 |
| Acurácia (Teste) | 0.9130 |

A correlação entre a Métrica (T_h) e a Acurácia pode ser visualizada razoavelmente nos grids abaixo:

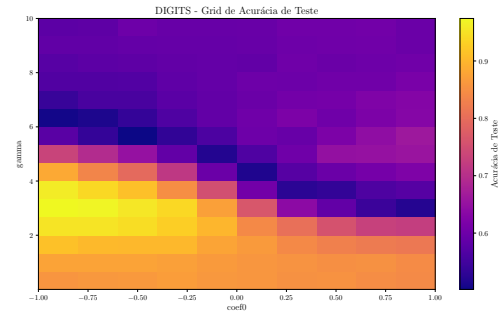


Fig. 7. GRID DE ACURÁCIA

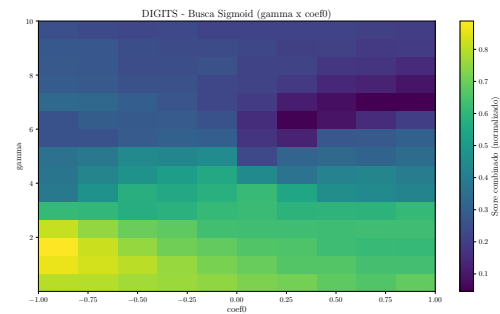


Fig. 8. GRID DA MÉTRICA UTILIZADA

Onde pode-se observar o platô de coloração brilhante em formato senoidal no canto inferior da imagem.

C. Wine

Para o dataset Wine, foram obtidos os seguintes resultados:

1) *RBF*: Para a RBF, o h que maximiza o valor da métrica foi de $h = 5.15$, o gráfico que demonstra a acurácia de teste supervisionada em relação a métrica não supervisionada foi:

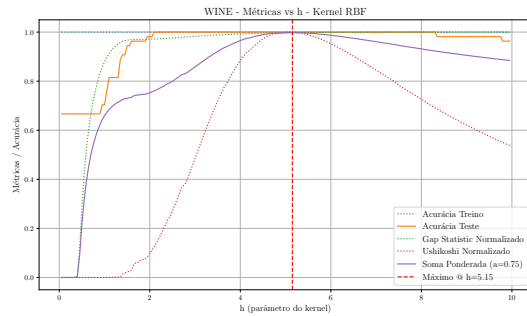


Fig. 9. MÉTRICA x ACURÁCIA

que por sua vez, tem o seguinte espaço de verossimilhança.

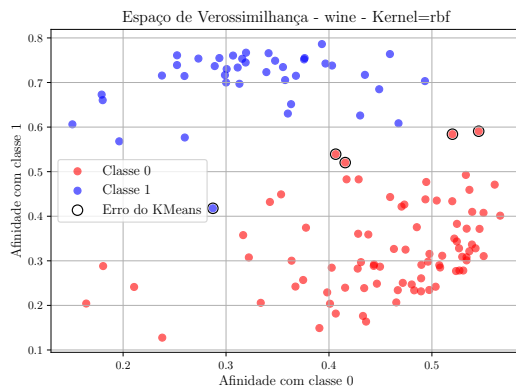


Fig. 10. ESPAÇO DE VEROSSIMILHANÇAS MÁXIMO

2) *Sigmoid*: Os coeficientes obtidos γ e $coef_0$ para o Sigmoid foram:

TABLE IV
MELHOR COMBINAÇÃO DE PARÂMETROS PARA O KERNEL SIGMOID

| Parâmetro | Valor |
|--------------------|---------|
| Gamma (γ) | 0.0518 |
| Bias ($Coef_0$) | -1.0000 |
| Métrica (T_h) | 0.9742 |
| Acurácia (Treino) | 0.9839 |
| Acurácia (Teste) | 1.0000 |

A correlação entre a Métrica (T_h) e a Acurácia pode ser visualizada razoavelmente nos grids abaixo:

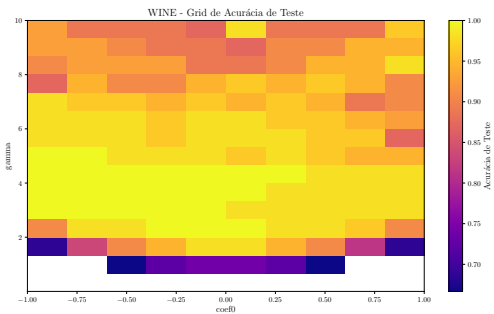


Fig. 11. GRID DE ACURÁCIA

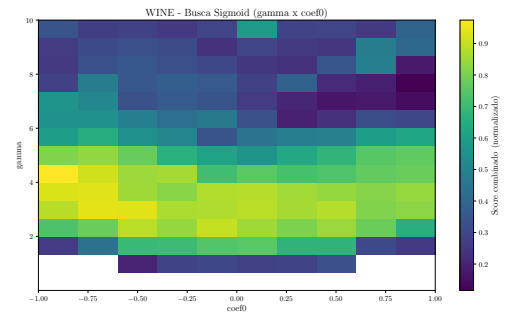


Fig. 12. GRID DA MÉTRICA UTILIZADA

Onde pode-se observar o platô de coloração brilhante no canto inferior da imagem.

D. Outros Datasets

Para os datasets restantes obtiveram-se resultados tão razoáveis quanto os obtidos anteriormente.

1) *RBF*: Gráficos do Kernel RBF:

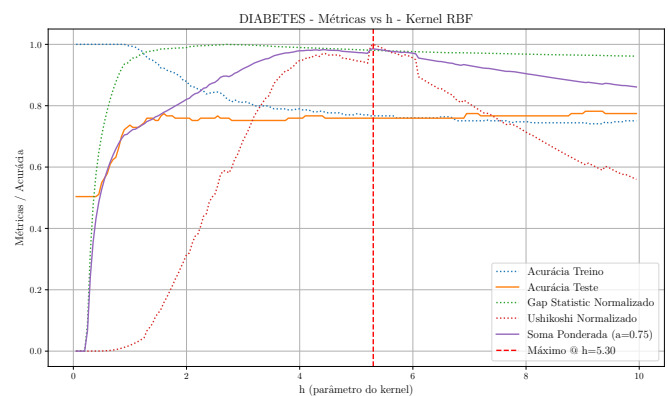


Fig. 13. MÉTRICA X ACURÁCIA (DIABETES)

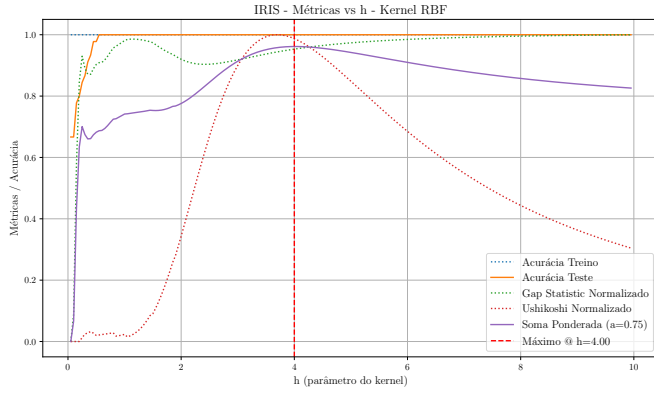


Fig. 14. MÉTRICA x ACURÁCIA (IRIS)

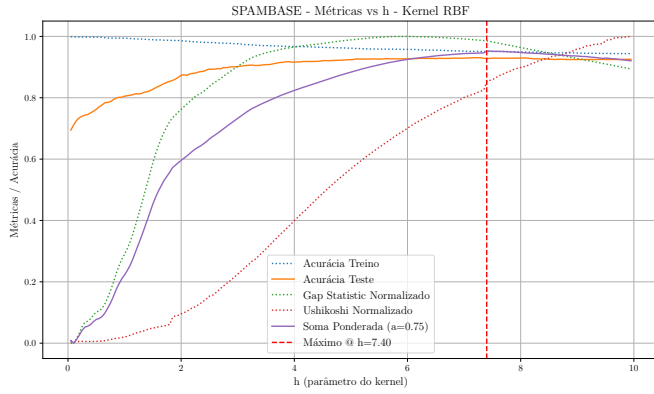


Fig. 15. MÉTRICA x ACURÁCIA (SPAMBASE)

2) Sigmoid: Resultados do Kernel Sigmoid:

TABLE V
MELHOR COMBINAÇÃO DE PARÂMETROS PARA O KERNEL SIGMOID - DIABETES

| Parâmetro | Valor |
|--------------------|--------|
| Gamma (γ) | 0.0010 |
| Bias (Coef_0) | 0.1111 |
| Métrica (T_h) | 0.9187 |
| Acurácia (Treino) | 0.7508 |
| Acurácia (Teste) | 0.6692 |

TABLE VI
MELHOR COMBINAÇÃO DE PARÂMETROS PARA O KERNEL SIGMOID - IRIS

| Parâmetro | Valor |
|--------------------|---------|
| Gamma (γ) | 10.0000 |
| Bias (Coef_0) | 0.3333 |
| Métrica (T_h) | 0.7518 |
| Acurácia (Treino) | 0.9333 |
| Acurácia (Teste) | 0.8889 |

TABLE VII
MELHOR COMBINAÇÃO DE PARÂMETROS PARA O KERNEL SIGMOID - SPAMBASE

| Parâmetro | Valor |
|--------------------|--------|
| Gamma (γ) | 0.0268 |
| Bias (Coef_0) | 1.0000 |
| Métrica (T_h) | 0.9071 |
| Acurácia (Treino) | 0.8360 |
| Acurácia (Teste) | 0.8472 |

E. Discussão

O Kernel RBF obteve um sucesso notável nos resultados em comparação ao Kernel Sigmoid, mais sensível a variações no formato dos dados. Ambos os kernels obtiveram um resultado razoável para a maioria dos casos e certa adaptabilidade além de algumas limitações. O Kernel RBF teve dificuldade ao se adaptar ao teste experimental realizado com espirais enquanto o Kernel Sigmoid teve dificuldades para valores de $|coef_0|$ maiores do que 1.

VII. CONCLUSÃO

O presente trabalho permitiu aplicar os métodos e algoritmos aprendidos em sala de aula durante a disciplina de Introdução ao Reconhecimento de Padrões, permitindo a integração de diversos temas aprendidos em sala de aula como **classificadores, SVM, espaço de verossimilhança e aprendizado não-supervisionado**.

Por fim, a combinação de todos esses elementos com soluções obtidas na literatura de aprendizado de máquina e testes experimentais nos permitiu encontrar uma solução não supervisionada razoavelmente promissora para a escolha dos **hiperparâmetros de kernel RBF e sigmoid** para a previsão da **acurácia** máxima por meio das métricas utilizadas e com resultados experimentais encorajadores nos datasets utilizados.

REFERENCES

- [1] T. A. Ushikoshi, E. J. R. Freitas, M. Menezes, W. J. A. Junior, L. C. B. Torres, and A. P. Braga, "Hebbian Learning with Kernel-Based Embedding of Input Data," *Neural Processing Letters*, vol. 56, no. 6, p. 250, 2024, Springer.
- [2] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001, Wiley Online Library.
- [3] A. P. Braga, "Aprendendo com Exemplos: Princípios de Redes Neurais Artificiais e de Reconhecimento de Padrões" *Disponibilizado via Google Drive durante aulas de Reconhecimento de Padrões, 2025, UFMG*