Breno Lima de Freitas

Aprendiz de Descritores de Mistura Gaussiana

Sorocaba, SP

2017

Breno Lima de Freitas

Aprendiz de Descritores de Mistura Gaussiana

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação (PPGCC-So) da Universidade Federal de São Carlos como parte dos requisitos exigidos para a obtenção do título de Mestre em Ciência da Computação. Linha de pesquisa: Inteligência Artificial e Banco de Dados.

Universidade Federal de São Carlos – UFSCar

Centro de Ciências em Gestão e Tecnologia – CCGT

Programa de Pós-Graduação em Ciência da Computação – PPGCC-So

Orientador: Prof. Dr. Tiago Agostinho de Almeida

Sorocaba, SP

2017

Lima de Freitas, Breno

Aprendiz de Descritores de Mistura Gaussiana / Breno Lima de Freitas. -- 2017.
90 f. : 30 cm.

Dissertação (mestrado)-Universidade Federal de São Carlos, campus Sorocaba, Sorocaba
Orientador: Tiago Agostinho de Almeida
Banca examinadora: Sahudy Montenegro González, Ana Carolina Lorena
Bibliografia

1. Aprendizado de Máquina. 2. Métodos de Classificação. 3. Princípio da Descrição Mais Simples. I. Orientador. II. Universidade Federal de São Carlos. III. Título.

Ficha catalográfica elaborada pelo Programa de Geração Automática da Secretaria Geral de Informática (SIn).
DADOS FORNECIDOS PELO(A) AUTOR(A)

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências em Gestão e Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Breno Lima de Freitas, realizada em 14/12/2017:

Prof. Dr. Tiago Agostinho de Almeida
UFSCar

Profa. Dra. Ana Carolina Lorena
UNIFESP

Profa. Dra. Sahudy Montenegro Gonzalez
UFSCar

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Ana Carolina Lorena e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ao) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.

Prof. Dr. Tiago Agostinho de Almeida

Aos meus pais, que me ensinaram a andar
Aos meus amigos, que me ensinaram a correr
Aos meus professores, que me ensinaram a voar

Agradecimentos

Primeiramente, agradeço imensamente a meus pais por todo apoio, horas, broncas, conselhos, sugestões e preces. Por compreenderem minha falta de compreensão e por me apoiarem nas minhas aventuras que exigem que meus dias tenham 28 horas. É difícil mensurar o impacto de tudo que fizeram na minha vida, mas se enxerguei mais longe, é por que com certeza estavam me levantando. Obrigado por terem "feito minha graduação e meu mestrado" comigo; essa dissertação é fruto de todo empenho que colocaram calçando o caminho que eu sigo.

Agradeço ao meu orientador, Tiago, por compartilhar uma ideia tão interessante e me introduzir à pesquisa em Aprendizado de Máquina. Obrigado por ter aceitado me orientar em uma situação tão peculiar e cheia de incertezas, e por nunca ter duvidado de que eu conseguiria terminá-la, mesmo quando nem mesmo eu acreditava. Agradeço também ao Renato, que me ajudou muito em todas as etapas deste trabalho.

Agradeço à minha querida orientadora de graduação, Cândida, por me mostrar a beleza das pequenas gemas escondidas na matemática e me introduzir à pesquisa científica. Obrigado por ter sido uma pessoa tão compreensiva, carinhosa e ter me ensinado lições valiosas, tanto na vida, quanto na academia, que me permitiram chegar até este momento. Sua energia e paixão pela pesquisa me motivam a sempre procurar o porquê de tudo, e a valorizar as pequenas vitórias da vida. É difícil exprimir em palavras a admiração que tenho pela senhora. Obrigado por tudo.

Agradeço a todos meus professores. Este momento é fruto de todo seu trabalho e dedicação. Agradeço à professora Denise, que me incentivou que eu entrasse na UFSCar. Agradeço às professoras Tiemi e Sahudy, pelas inúmeras risadas, aprendizados, brincadeiras e almoços (espero termos mais). Agradeço aos professores Guimarães e Gustavo por todo conhecimento, histórias e provas memoráveis. Finally, I would like to thank professor Dan for accepting being my advisor back in 2014, and being so welcoming and resourceful; I carry your lessons throughout my life, and I will never forget that "the only thing that separates a skilled person from a beginner, is the time they take to reach the same point".

Agradeço aos meus amigos do *Player 4* por tantas risadas, histórias, reclamações, *board-games*, carros atolados, cantorias, chaves presas dentro do carro, julgamentos e 1-reais que de fato só mandei por *stickers*. Vocês fizeram parte de um momento muito disruptivo e especial da minha vida. Obrigado por sempre estarem dispostos a oferecer um ombro amigo ou a fazer um suco de limão selvagem. Eu guardo com muito carinho todos nossos momentos, e meus "só posso ficar mais trinta minutos" com vocês. Tenho certeza que vamos ter motivos para muito mais *stickers*, aguardo ansioso por eles.

Agradeço aos meus amigos do *FDH* pelas conversas mais filosóficas e/ou sem sentido que já tive, pelos piores memes que só nós entendemos, por todos os *gifs* com os quais eu fico rindo incessantemente e por terem, acima de tudo, sido "meu grupo" na graduação. Fico muito feliz de poder dizer que temos essa amizade há tanto tempo, e ver o quanto eu mudei com vocês. Obrigado por terem sido minha segunda família em diferentes momentos da minha vida, e por terem estado presentes em nossa aventura louca de sair do país pela primeira vez. Obrigado por todo apoio que me deram, especialmente em meus momentos baixos ou quando eu simplesmente acordava de mal-humor. Espero poder falar de "problemas de revista Coquetel" com vocês, de perto, em um futuro próximo.

Agradeço aos meus amigos do *Canindaiá* por todos os momentos bons antes de minha viagem. Por todas as horas de diversão, por todas as horas de não fazer sentido. Dizem que é difícil manter amizades por tanto tempo, acho que somos a prova viva que amizades de verdade não têm fronteiras e não respeitam o relógio. Obrigado pelas diversas horas de "a Bruna já está dormindo" e "é biscoito". Guardo com carinho nossas histórias, vocês não se cansam de me surpreender. Tenho que admitir que minha vinda para o Canadá não teria sido a mesma sem aquela despedida, obrigado por serem tão presentes em vários momentos da minha vida. Espero vocês aqui para fazermos guerra de neve.

Agradeço aos meus mais novos, mas não menos queridos, amigos *Ottawanos*. Obrigado por compartilharem de momentos tão bons comigo. Agradeço por sempre se animarem com minhas loucuras, por todas as risadas que já demos juntos, pelas horas falando de política, problemas de gente grande, e por confiarem em mim. Sou muito grato de ter encontrado cada um de vocês aqui, pelas sextas-feiras (por que é sexta, não é mesmo?), pelas experiências culinárias, por me ignorarem de fim de semana para eu terminar esta dissertação, e por todos os gatos que esperamos estarem sendo bem cuidados por aí. Apesar do pouco tempo juntos, tenho certeza que encontrei grandes amigos que vou levar para toda vida.

Agradeço também a amigos que não fazem parte de nenhum grupo: Carol, Larissa e Vitor. Obrigado por tantos momentos bons, alguns momentos confusos e várias memórias. Dizem que amigos são a família que nós escolhemos; eu tenho a sorte de, não só escolher, como também ter sido escolhido por tantas pessoas especiais. Muito obrigado.

I would also like to thank my ex-lead, Edward. Thank you for being so understanding, kind and resourceful. Thank you for all the support you provided me when I arrived; for the long talks in our 1:1s and for making sure I was following my passion and achieving my dreams. Thank you for being such an inspiring person and for sharing so many valuable lessons with me. Above all, thank you for being more than a lead, a friend.

Finalmente agradeço a UFSCar Sorocaba por permitir que usasse seus recursos para a realização deste trabalho.

"Nanos Gigantum Humeris Insidentes"
(Policraticus, João de Salisbury)

Resumo

Ao longo das últimas décadas, diversos métodos de aprendizado de máquina vêm sendo propostos com o intuito de classificar dados. Entre os modelos candidatos, procura-se selecionar um que se ajuste bem aos dados de treinamento, criando uma hipótese que faça boas predições em amostras não analisadas anteriormente. Um dos maiores desafios é selecionar um modelo, cuja hipótese não seja sobre-ajustada aos dados conhecidos, sendo genérica o suficiente para boas predições futuras. Neste trabalho, é apresentado um método de classificação baseado no princípio da descrição mais simples que efetua uma troca benéfica entre a complexidade do modelo e o ajuste aos dados. O método proposto é multiclasse, incremental e pode ser usado em dados com atributos categóricos, numéricos e contínuos. Experimentos conduzidos em bases reais de diversas características mostraram que o método proposto é estatisticamente equivalente à métodos clássicos na literatura para o cenário *offline* e superior a alguns métodos no cenário de aprendizado incremental. Além disso, o método mostrou-se robusto ao sobre-ajustamento e à normalização dos dados, apresentando características benéficas para um método de classificação que pode ser aplicado nos dias atuais.

Palavras-chaves: Princípio da descrição mais simples. Mistura Gaussiana. Classificação. Aprendizado de máquina.

Abstract

For the last decades, many Machine Learning methods have been proposed aiming categorizing data. Given many tentative models, those methods try to find the one that fits the dataset by building a hypothesis that predicts unseen samples reasonably well. One of the main concerns in that regard is selecting a model that performs well in unseen samples not overfitting on the known data. In this work, we introduce a classification method based on the minimum description length principle, which naturally offers a tradeoff between model complexity and data fit. The proposed method is multiclass, online and is generic in the regard of data representation. The experiments conducted in real datasets with many different characteristics, have shown that the proposed method is statiscally equivalent to the other classical baseline methods in the literature in the offline scenario and it performed better than some when tested in an online scenario. Moreover, the method has proven to be robust to overfitting and data normalization which poses great features a classifier must have in order to deal with large, complex and real-world classification problems.

Key-words: Minimum Description Length Principle. Gaussan Mixture. Classifiers. Machine Learning.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

EDK	Estimatidor de densidade de *kernel*
GMDL	*Gaussian Mixture Descriptors Learner* (Aprendiz de Descritores de Mistura Gaussiana)
GNB	*Gaussian Naïve Bayes* (Bayes Ingênuo Gaussiano)
KDE	*Kernel Density Estimator*
MDL	*Minimum Description Length Principle* (Princípio da Descrição Mais Simples)
MLP	*Multi-Layer Perceptron* (Perceptron multinível)
NB	*Naïve Bayes* (Bayes Ingênuo)
PA	*Passive-Agressive* (Passivo-Agressivo)
RF	*Random Forest* (Floresta Aleatória)
SGD	*Stochastic Gradient Descent* (Gradiente Descendente Estocástico)
SVM	*Support Vector Machines* (Máquinas de vetores de suporte)
kNN	*k-nearest Neighbours* (k-vizinhos mais próximos)
fn	Falso negativos
fp	Falso positivos
oKDE	*Online Kernel Density Estimator*
vp	Verdadeiro positivos

Lista de símbolos

$\mathcal{X}$	Alfabeto finito de símbolos
x^n	Sequência de símbolos $(x_1, x_2, \ldots, x_n)$ onde cada $x_i \in \mathcal{X}$
X^n	Conjunto de sequências de n símbolos em $\mathcal{X}$
$\mathbb{N}$	Conjunto dos número naturais
$\mathcal{C}$	Conjunto de todas as codificações
$\mathcal{L}$	Conjunto de funções de tamanho de código
$\mathcal{M}$	Modelo – conjunto de fontes de probabilidade
$\mathcal{X}^n$	Conjunto de todas as sequências finitas de n símbolos
$P(x)$	Probabilidade de x
$P(\cdot \mid \theta)$	Distribuição de probabilidade parametrizada por θ
$P(\cdot \mid \hat{\theta})$	Distribuição de probabilidade parametrizada pelo θ ótimo
$C(y)$	Codificação de um símbolo y
$I(X)$	Conteúdo de informação (ou surpresa) de uma variável aleatória X
$E_P[X]$	Esperança de uma variável aleatória X distribuida em P
$H[X]$	Entropia de uma variável aleatória X
l_i	Tamanho do código de um símbolo na posição i
$\hat{l}(X)$	Tamanho ótimo do código de um conjunto de sequências de símbolos X
$\hat{L}$	Função de tamanho de código
$\mathcal{R}(P)$	Arrependimento de uma distribuição de probabilidade P
$\mathcal{R}_{max}(P)$	Arrependimento máximo de uma distribuição de probabilidade P
$\mathbb{C}$	Complexidade de um modelo
$\arg\min_{z \in Z}[f(z)]$	Elemento z no conjunto Z que minimiza f
K	Conjunto de todas as classes

n	Número de atributos
m	Número de amostras
$\vec{x}$	Amostra de dado
$\hat{c}$	Classe prevista
c	Classe $c \in K$
c_i	Classe i, tal que $i \in \{1, \ldots, \lvert K \rvert\}$
p'	Função densidade originada pelo oKDE
p	Função densidade ajustada
$\hat{p}$	Função densidade aproximada
μ	Média
σ^2	Variância
σ	Desvio padrão
Σ	Matriz de covariância
$\mathcal{N}(\mu, \Sigma)$	Distribuição Gaussiana parametrizada por $\langle \mu, \Sigma \rangle$
$\bar{v}$	Média de um conjunto $(v_1, \ldots, v_n)$
$J(\cdot)$	Função de custo
∇J	Gradiente da função de custo J
$D(p, d)$	Distância Mahalanobis do ponto p à distribuição d
$M_{2,n}$	Soma das diferenças quadráticas em relação a média até a n-ésima amostra
$M_X(s)$	Função geradora de momento de X
ϕ_{ic}	Abreviação de $\lceil -\log p_{(i,c)}(\vec{x}_i) \rceil$
Φ	Função de Kernel
$\lVert \cdot \rVert_\infty$	Norma suprema
$diag(\cdot)$	Vetor diagonal de uma matriz
A^T	Transposta da matriz ou vetor A

A^{-1}	Matriz inversa de A
$X \sim D$	X é uma variável aleatória definida por D
G	Número de componentes Gaussianas em uma mistura
$O(\cdot)$	Função de limite superior assintótico
L	Função de tamanho de código normalizada
$\hat{S}$	Distância Mahalanobis normalizada
Θ	Vetor de pesos dos atributos
θ_i	Valor i de Θ
δ_{ij}	Delta de Kronecker para os valores i e j
τ	Parâmetro de regularização de $\hat{S}$
β	Valor da densidade a ser considerada no valor máximo de $\hat{S}$
η	Taxa de aprendizado
α	Taxa do momentum
Ω	Valor da densidade a ser considerada nos extremos de $\hat{p}$
$\tilde{\sigma}^2$	Desvio padrão para relaxamento de funções Delta-Dirac
f	Fator de esquecimento
R_i	Soma dos *ranks* do método i
R^+	Soma dos *ranks* onde o segundo algoritmo foi melhor que o primeiro
R^-	Soma dos *ranks* onde o primeiro algoritmo foi melhor que o segundo

Sumário

Prefácio

A categorização automática de dados por meio de métodos de classificação tem sido alvo de grande interesse nas últimas décadas. Por tal motivo, diversos métodos foram propostos em um curto período de tempo. Dentre estes métodos, destacam-se aqueles que, por terem sido utilizados com sucesso, são tidos como tradicionais; a saber: Naïve Bayes – *NB* (DUDA; HART, 1973), máquinas de vetores de suporte – *SVM* (CORTES; VAPNIK, 1995) – *SVM*, e redes neurais artificiais (ROSENBLATT, 1958).

Os métodos de classificação baseiam-se na seleção de um modelo, dada uma hipótese, que tem como objetivo ajustar-se a um conjunto de dados para que possa computar uma saída (classe) para uma amostra nunca observada. Além disso, é desejável que o modelo escolhido tenha uma alta capacidade de *generalização*, isto é, seja capaz de produzir uma saída considerada adequada e consistente à solução do problema, tanto para as amostras utilizadas no conjunto de dados de treinamento, quanto para amostras ainda não observadas.

Dado o grande número de possíveis modelos que podem descrever o conjunto de dados de treinamento, os métodos utilizam critérios de seleção distintos (*e.g.*, probabilidade, otimização, distância) para tentar selecionar o melhor modelo possível. William Occam, filósofo e frade inglês, cunhou o conceito conhecido como *Navalha de Occam*, o qual dita que "entidades não devem ser multiplicadas além do necessário" – uma crítica à filosofia escolástica que tratava a realidade com teorias muito complexas. No contexto de aprendizado de máquina, a Navalha de Occam é utilizada para que na escolha de um modelo, na presença de múltiplas opções que descrevem o mesmo problema, aquele que comprime mais os dados deve ser selecionado (DOMINGOS, 1999).

Rissanen (1978) definiu o Princípio da Descrição mais Simples (*Minimum Description Length – MDL*), enraizando-se nas ideias da complexidade de Komolgorov (KOLMOGOROV, 1963) e formalizando a Navalha de Occam para o problema de seleção de modelos. Tal princípio define que o modelo que melhor se adapta aos dados e possui tamanho de descrição menos complexo, deve ser selecionado. Deste modo, o MDL elege modelos que preservam um equilíbrio favorável entre a capacidade de descrever os dados de treinamento e a complexidade do mesmo. Esta característica evita naturalmente o sobreajuste (*overfitting*) e, portanto, é favorável para um método de classificação. Nos dias atuais, é desejável que métodos de classficação se adaptem bem à mudanças e sejam aptos a escalarem bem os dados. Métodos tradicionais, no geral, falham em um ou nesses dois quesitos, sendo muito restritos às bases treinadas, e podendo ter seu desempenho prejudicado a longo prazo. Deste modo, a teoria do MDL apresenta a vantagem de ser

naturalmente incremental e escalável, além de ser relativamente simples de se implementar.

Objetivos

Esta dissertação de mestrado oferece um método de classificação genérico, multinomial e incremental baseado no Princípio da Descrição mais Simples. O método é uma extensão do método `MDLText` proposto por Silva, Almeida e Yamakami (2016a) para classificação de textos. O método é multiclasse e naturalmente robusto em relação à normalização e ao sobreajuste dos dados. Ele foi testado com bases públicas conhecidas do repositório da UCI (LICHMAN, 2013) e comparado com métodos considerados estado-da-arte em classificação.

Em resumo, esta dissertação procura responder as seguintes questões de pesquisa:

1. É possível estimar de maneira incremental uma função de distribuição de probabilidade de um atributo contínuo, para assim evitar discretização offline?
2. Como podemos adaptar o método apresentado em Almeida (2010) e Silva, Almeida e Yamakami (2016a), Silva, Almeida e Yamakami (2016b) para que seja capaz de processar amostras representadas também por atributos contínuos?

Organização

Este manuscrito foi estruturado da seguinte maneira:

- No Capítulo 1, são apresentados os conceitos e bases teóricas do MDL e Teoria da Informação. Neste capítulo, discute-se também relações entre a Teoria da Informação e Estatística que criam a ponte entre o método proposto e o princípido do MDL.
- No Capítulo 2, é apresentada uma revisão da literatura explorando os diversos campos nos quais o MDL foi utilizado com sucesso e a forma no qual sua essência foi aplicada na solução dos problemas.
- No Capítulo 3, é apresentado o método proposto. Este capítulo é subdivido de maneira a construir o método a partir das pequenas partes que o compõe, apontando suas origens na Teoria da Informação e Estatística, até o método final de forma mais ampla.
- No Capítulo 4, são apresentadas as configurações utilizadas para a avaliação e comparação do método proposto.
- Finalmente, são apresentadas as conclusões e direcionamentos para trabalhos futuros.

1 O Princípio da Descrição Mais Simples

Rissanen (RISSANEN, 1978; RISSANEN, 1983), inspirado no conceito da Navalha de Occam, formalizou o princípio da *descrição mais simples* (do inglês, *Minimum Description Length – MDL*), o qual dita que no problema de seleção de modelos, aquele que possuir menor tamanho de descrição deve ser priorizado. Este princípio, oriundo da Teoria da Informação, delineia que quanto mais se sabe sobre os dados, maior será a regularidade descoberta e, portanto, mais pode-se comprimi-los (GRÜNWALD, 2005).

O MDL possui raízes na complexidade de Kolmogorov (KOLMOGOROV, 1963), definida como o menor tamanho de um programa que imprime uma dada sequência e finaliza. Portanto, quanto menor a complexidade de Kolmogorov mais conhecimento há sobre a sequência sendo codificada. Deste modo, é possível codificar tal sequência mais efetivamente, e este programa deve ser selecionado para representá-la (BARRON; RISSANEN; YU, 1998). Para exemplificar, considere as sequências binárias mostradas na Figura 1.

1. `11011011011011011011011011011011 ... 11011011011011011011011011011011`
2. `11101010100101010001011101001 ... 10100101010010100010110111001 0`

Figura 1 – Sequências binárias originais antes de uma codificação (GRÜNWALD, 2000).

É possível perceber um certo padrão na sequência 1, onde `110` se repete por n vezes, enquanto a sequência 2 aparenta ser totalmente aleatória. Pode-se, então, eleger uma linguagem de programação como *Ruby*, para representar a sequência 1, como mostrado na Figura 2. Nesta representação, é possível notar que houve uma redução do tamanho da sequência. A sequência 2, no entanto, só pode ser representada por um programa que a imprima e termine (GRÜNWALD, 2005); logo, representá-la através de um programa aumentaria sua descrição. A inclusão dos conceitos tanto da Navalha de Occam quanto da complexidade de Kolmogorov, provêem ao Princípio da Descrição mais Simples um equilíbrio benéfico entre a seleção da complexidade do modelo e seu ajuste aos dados, o que evita, portanto, um modelo complexo (GRÜNWALD, 2000).

```
(1..n).each{|x| print '110'}; exit!
```

Figura 2 – Programa em *Ruby* que imprime a sequência binária 1 apresentada na Figura 1.

Este capítulo está estruturado da seguinte forma: a Seção 1.1 apresenta a terminologia e definições utilizadas. Na Seção 1.2, conceitos de codificação são introduzidos e posteriormente explorados na Seção 1.3 que interrelaciona conceitos de probabilidade e Teoria da Informação. Finalmente, a Seção 1.4, aborda uma extensão do MDL.

1.1 Terminologia e Definições

Ao decorrer deste capítulo, alguns símbolos e definições serão utilizados para a derivação do MDL. Esta seção, define-os e introduz conceitos que permitem uma abordagem mais concisa.

Seja $\mathcal{X}$ um alfabeto finito de símbolos. $\mathcal{X}^n$ o conjunto de todas as sequências finitas de n símbolos. A notação x^n é adotada para representar uma sequência de símbolos $(x_1, x_2, \ldots, x_n)$, onde cada $x_i \in \mathcal{X}$. X^n é um conjunto de sequências de n símbolos; onde ficar claro pelo contexto em que é utilizado, se usará X ao invés de X^n. Uma *fonte* de probabilidade P é uma sequência $P^{(1)}, P^{(2)}, \ldots$ em $\mathcal{X}^1, \mathcal{X}^2, \ldots$, tal que $P^{(n)}$ é igual a probabilidade marginal de $P^{(n+1)}$; onde ficar claro pelo contexto em que é utilizado, se usará P ao invés $P^{(n)}(x)$, com $x \in \mathcal{X}^n$. Seja P uma distribuição de probabilidade definida em $\mathcal{X}$, $P(x)$ é chamada de probabilidade de x. Se $\sum_x P(x) < 1$, P é dita *imperfeita*. Se os dados são independentemente e identicamente distribuídos, então, dada uma fonte P, para cada $n, x^n \in \mathcal{X}^n$, segue que $P(x^n) = \prod_i P(x_i)$.

No MDL, procura-se por uma codificação C de um conjunto de dados (símbolos, em uma analogia matemática) D que consiga descrevê-lo de maneira única da menor forma possível. Isto é, procura-se por um C na qual $L_C(D)$ seja a menor possível, sendo L_C uma função que descreve o tamanho dos dados codificados com o auxílio de C. No contexto desta pesquisa, dentre todas as codificações possíveis em $\mathcal{C} : \mathcal{X} \to \mathbb{N}$, são utilizadas aquelas em que $\mathcal{C} : \mathcal{X} \to \bigcup_{n\geq 1}\{0,1\}^n$, ou seja, aquelas descritas na base binária, usando, portanto, o *bit* como medida de informação. Com tal medida de informação, *log* é utilizado para respresentar o logaritmo de base 2.

1.2 Codificações

Uma codificação é uma função que mapeia um conjunto de dados em outro conjunto, usualmente em outro domínio de forma mais compacta (para uma definição de compacto). Não é difícil notar que o número máximo de sequências que podem ser codificadas com n *bits* é 2^n. Apesar de ser interessante conhecer limiares máximos, saber os limitantes de uma codificação é uma informação mais relevante: o maior número de sequências que podem ser codificadas com até n *bits* é dado por $\sum_{i=1}^{n} 2^i < 2^{n+1}$. Segue desta observação que a razão entre sequências que podem ser reduzidas por até k *bits* é definida por $\frac{2^{n-k}}{2^n} = 2^{-k}$. Fica claro que apenas um número pequeno de símbolos pode, portanto, ser reduzido a uma codificação de tamanho razoavelmente pequeno. Note que é possível estabelecer uma relação entre o tamanho máximo do código com uma distribuição de probabilidade no conjunto, *i.e.*, no máximo 2 códigos podem ser codificados com $1 = -\log \frac{1}{2}$ *bits*, 4 com $2 = -\log \frac{1}{4}$ *bits*, e generalizando, n com $-\log \frac{1}{n}$ *bits*. De modo análogo, dada uma probabilidade P, no máximo 2 códigos podem possuir probabilidade $P(z) \geq \frac{1}{2}$, no máximo

4 com $P(z) \geq \frac{1}{4}$, e generalizando, no máximo k podem ter probabilidade $P(z) \geq \frac{1}{k}$.

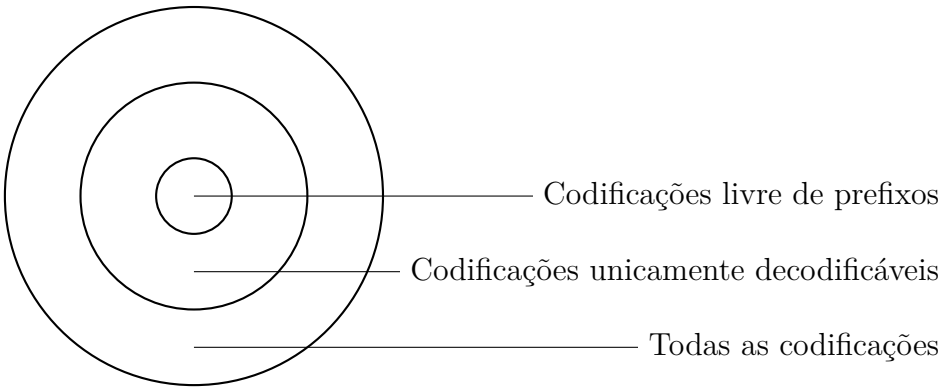


Figura 3 – Relação entre diferentes classes de codificação.

Uma codificação C é dita *livre de prefixos* se nenhum código em C é prefixo de outro código, deste modo a decodificação pode ser feita no mesmo momento que é recebida. Uma codificação C é *unicamente decodificável*, se para todo código x^n, y^n vale que $C(x^n) \neq C(y^n), \forall x^n, y^n \in \mathcal{X}^n$. A Figura 3 mostra graficamente estas relações em $\mathcal{C}$. Pelas relações obtidas entre probabilidade de um código em um conjunto e o tamanho do mesmo, é possível inferir que possa haver uma equivalência entre tamanho de código e probabilidade de uma dada codificação C, dada uma função L_C. Kraft (KRAFT, 1949) descreveu uma relação entre codificação e tamanho de códigos, enunciada a seguir.

Teorema 1.1 (Inequalidade de Kraft) *Existe uma codificação livre de prefixos de aridade D com tamanho de códigos $l_1, \ldots, l_n$ se e somente se $\sum_{i=1}^{n} D^{-l_i} \leq 1$.*

Caso a inequalidade apresentada no Teorema 1.1 seja estritamente menor que um, dizemos que a codificação utilizada possui *redundância*, isto é, há como utilizar menos *bits* para codificar um símbolo; caso seja igual a um, o código C é chamado de *completo*, isto é, $\forall C' \in \mathcal{C}, L_C(x) \leq L_{C'}(x)$; se a inequalidade não for mantida, então C não é unicamente decodificável.

A Proposição 1.1 mostra que a codificação de símbolos em binários onde todos os códigos possuem tamanhos idênticos é redundante.

Proposição 1.1 *Seja $\mathcal{X} := \{0, 1\}$ o alfabeto usado por uma codificação livre de prefixos C. Se $X^m \in \mathcal{X}^m$, então C é redundante.*

Prova: Como C é livre de prefixos, segue do Teorema 1.1 que

$$\begin{aligned} \sum_{i=1}^{|X|} 2^{-m} &\leq 1 \\ 2^{-m \cdot |X|} &\leq 1 \\ &< 1 \end{aligned}$$

□

Pode-se definir o conjunto de funções de tamanho de código $\mathcal{L}$, sob um espaço de símbolos Z, baseado no Teorema 1.1, mostrado pela Equação 1.1.

$$\mathcal{L}_Z := \left\{ L : Z \to [0, \infty) \;\middle|\; \sum_{z \in Z} 2^{-l_z} \leq 1 \right\} \tag{1.1}$$

Formalmente, L_C representa o tamanho esperado dos dados codificados por C. Como C induz uma distribuição de probabilidade P num conjunto de dados, então $L_C := \sum_i x_i \cdot p(x_i)$, onde x_i é um símbolo a ser codificado por C. A *esperança* de uma distribuição de massa P para uma variável aleatória X é definida como $E_P[X] = \sum_{i=1}^{\infty} x_i \cdot p_i$, isto é, o valor esperado dado as probabilidades de um evento ocorrer atrelado a seu respectivo valor. Diz-se, então, que L_C é a esperança de X em P. No MDL, a codificação C não é o fator de maior importância: o tamanho da codificação gerada por C é o valor que importa na codificação de uma mensagem. Deste modo, procura-se por um L que represente a menor codificação dentre todas as possíveis em um espaço $Z \subseteq \mathcal{X}^n$ como a mostrada pela Equação 1.2.

$$L := \arg\min_{L \in \mathcal{L}_Z} E_P[L(Z)] \tag{1.2}$$

1.3 Teoria da Informação

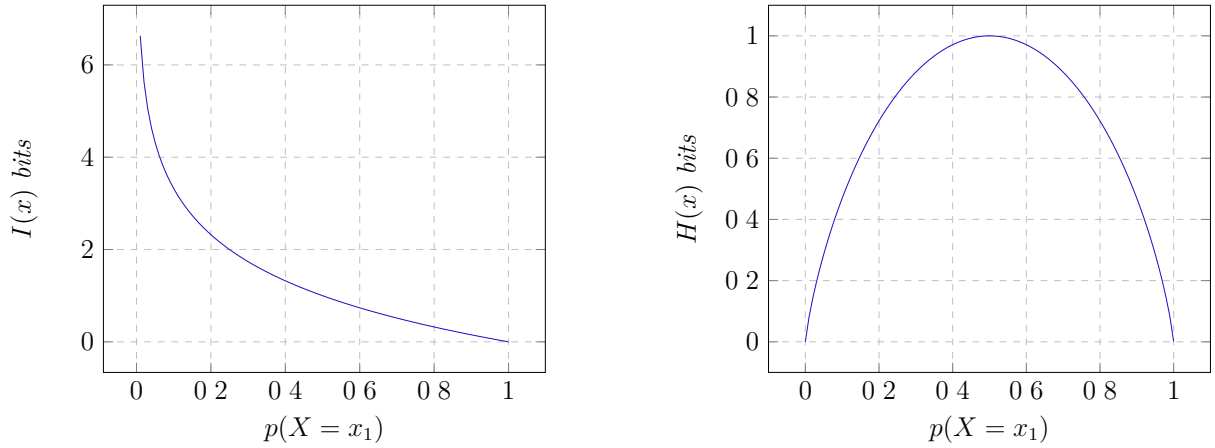
O MDL possui suas raízes na Teoria da Informação (RISSANEN, 1978; RISSANEN, 1983; COVER; THOMAS, 1991), que é intimamente relacionada à Teoria da Probabilidade. A Teoria da Informação lida com a transferência e recepção de mensagens e como compreendê-las eficientemente. Em probabilidade, pode-se estimar a média da distribuição utilizando o Teorema 1.2.

Teorema 1.2 (Lei dos grandes números) *Se Y é uma variável aleatória, em uma distribuição P, então*

$$\lim_{|Y| \to \infty} \frac{1}{|Y|} \sum_{i=1}^{|Y|} Y_i = E_P[Y]$$

A noção de esperança está intimamente relacionada àquela de entropia da Teoria da Informação, portanto, pode-se reescrever o Teorema 1.2 baseando-se na entropia oriunda de uma variável aleatória Y. A *entropia* de uma variável aleatória Y é definida como $H[Y] := \sum_{i=1}^{n} P(y_i) \cdot I(y_i) = -\sum_{i=1}^{n} P(y_i) \log P(y_i)$. I é chamado de *conteúdo de informação* ou *surpresa* e é representado por unidades de informação, como *bits* ou *nats*.

O conteúdo de informação mede a relevância de eventos pouco esperados – um evento com baixa probabilidade de ocorrer possui muita informação – e é formalmente definido como $I(Y) = \log \frac{1}{P(Y)} = -\log P(Y)$. Portanto, pode-se enxergar a entropia como o valor esperado do conteúdo de informação. A relação entre entropia e conteúdo de informação é mostrada na Figura 4. Como mostrado por Cover e Thomas (1991), o equivalente ao Teorema 1.2 é conhecido por Propriedade da Equipartição Assintótica, enunciada a seguir.



(a) Conteúdo de Informação.

(b) Entropia.

Figura 4 – Gráficos do comportamento do conteúdo de informação e entropia de uma variável aleatória X para o símbolo x_1 no alfabeto $\mathcal{X} = \{x_1, x_2\}$.

Teorema 1.3 (Equipartição Assintótica) *Se Y é uma variável aleatória independentemente e identicamente distribuída, então*

$$\lim_{n\to\infty} \frac{1}{n} \log \frac{1}{P(Y_1, Y_2, \ldots, Y_n)} = H[Y].$$

Possui-se, portanto, um guia não muito preciso de qual L procuramos, mas ainda é necessário definir qual o tamanho mínimo ideal para L com base nas informações apresentadas pelo Teorema 1.1. É possível mostrar que o tamanho ótimo é limitado inferiormente por $-\log P(x)$ para uma dada sequência de símbolos x, afirmado pelo Teorema 1.4. Para tal, o método dos multiplicadores de Lagrange é utilizado para encontrar o extremo da função. Tal método baseia-se na otimização de uma função $\Lambda(x, c) = f(x) + \lambda(g(x) - c)$, minimizando $f(x)$ sujeito a $g(x) = c$.

Teorema 1.4 *Seja C uma codificação de uma sequência de símbolos x^n. Seja P a distribuição associada a C e l_i o tamanho da codificação do símbolo x_i. A menor codificação possível para um símbolo x_k é limitada inferiormente por $-\log P(x_k)$.*

Prova: Utilizando o método dos multiplicadores de Lagrange com $f(x, P, l) = \sum_{i=1}^{n} P(x_i) \cdot l_i$ e $g(x, l) = \sum_{i=1}^{n} 2^{-l_i} - 1$, obtido a partir do Teorema 1.1, a derivada é igualada a zero para

encontrar seu extremo.

$$\Lambda(x, P, l) = \sum_{i=1}^{n} P(x_i) \cdot l_i + \lambda \left(\sum_{i=1}^{n} 2^{-l_i} - 1 \right)$$

$$\frac{\partial}{\partial l_k} \Lambda(x, P, l_k) = P(x_k) - \lambda 2^{-l_k} \log 2 \qquad \text{Derivando para cada } k \in \{i, \ldots, n\}$$

$$2^{-l_k} = \frac{P(x_k)}{\lambda} \qquad \text{Igualando a zero}$$

Pelo Teorema 1.1 sabe-se que:

$$\sum_k 2^{-l_k} \leq 1$$

$$\sum_i \frac{P(x_i)}{\lambda} \leq 1$$

$$\frac{1}{\lambda} \leq 1$$

E, portanto:

$$2^{-l_k} = P(x_k) \cdot \frac{1}{\lambda} \leq 1 \cdot P(x_k)$$

$$2^{-l_k} \leq P(x_k)$$

$$\log 2^{-l_k} \leq \log P(x_k)$$

$$-l_k \leq \log P(x_k)$$

$$\hat{l}_k \geq -\log P(x_k)$$

□

Note que uma consequência direta do Teorema 1.4 é que $\hat{L} = E[\hat{l}(X)] = \sum_{x \in X} P(x) \cdot \hat{l}(x) \geq -\sum_{x \in X} P(x) \cdot \log P(x) = H[X]$. Ou seja, o tamanho esperado de uma codificação livre de prefixos é delimitada pela entropia da fonte. No entanto, ainda não se foi decidida a distribuição para tal codificação. Não é difícil notar que se uma distribuição gera uma dada codificação, então sua entropia é a menor dentre qualquer outra distribuição que tente descrever a mesma codificação. Tal propriedade é conhecida como *Inequalidade da informação*, enunciada a seguir.

Proposição 1.2 (Inequalidade da informação) *Se P e Q são distribuições de probabilidade tal que $P \neq Q$, então $E_P[P(Y)] < E_P[Q(Y)]$, para uma variável aleatória Y.*

A partir do Teorema 1.1 pode-se garantir a existência de uma codificação livre de prefixos C, com uma distribuição de massa de probabilidade P associada. Pelo Teorema 1.2, L_C se aproxima do ótimo ao aumentar-se o número de amostras. Finalmente,

a Proposição 1.2 garante que, caso P seja de fato a distribuição que tenta-se aproximar, então $L_C(z) = \lceil -\log P(z) \rceil$ para todo z em um conjunto de sequências de símbolos Z definida por P.

O MDL foi originalmente cunhado como um método de duas partes – também chamado de MDL primitivo (do inglês *crude MDL*), como mostrado pela Equação 1.3. Nesta versão, dado um conjunto de fontes de probabilidade (modelos candidatos) $\mathcal{M}$, o modelo M que deve ser escolhido é aquele que minimiza tal equação.

$$M_{MDL} := \arg\min_{M \in \mathcal{M}} [L(M) + L(D|M)] \tag{1.3}$$

1.4 MDL Refinado

Pelo Teorema 1.4, tem-se uma boa estimativa para $L(D|M)$, uma vez que sabemos que existe uma codificação C, com L_C associado, tal que para todo $x^n \in \mathcal{X}^n, L(x^n) = \lceil -\log P(x^n \mid M) \rceil$, onde $P(\cdot \mid M)$ é a probabilidade associada a uma função de código para $\mathcal{X}^n$. Este método, para a segunda parte da equação é aceitável uma vez que *(i)* a log-semelhança é um método estatístico padrão para a medida de o quão bem ajustado um modelo está à hipótese e *(ii)* se P gera $\mathcal{X}^n$, então L_C é ótima (GRÜNWALD, 2005). No entanto, ainda não é claro como codificar o próprio M. Barron e Cover (1991) propuseram uma maneira de encontrar uma função de densidade estimada $\hat{p}$ para uma distribuição utilizando conceitos da complexidade de Kolmogorov, mostrada pela Equação 1.4.

$$\hat{p} \equiv \min_{\gamma \in \Gamma} L(\gamma) + \log \frac{1}{\prod_i \gamma(Y_1, \ldots, Y_n)} \tag{1.4}$$

Neste método, os autores consideram Γ como um conjunto de funções de densidade de probabilidade candidatas, no qual amostras aleatórias de uma variável Y são aplicadas; e $L(\gamma)$ representa o menor programa que consegue descrever γ, utilizando a complexidade de Kolmogorov como uma noção algorítmica de entropia. Note que ainda não é claro como criar codificações para hipóteses; além disso, não há uma maneira única de escolher uma função de probabilidade para descrever uma dada codificação (GRÜNWALD, 2005).

Rissanen (1983), Barron e Cover (1991), Barron, Rissanen e Yu (1998), Hansen e Yu (2001) deram os primeiros passos para refinar o MDL a caminho do que é chamado de codificação universal (GRÜNWALD, 2005), abandonando o MDL de duas partes em favor de uma descrição mais formal e direta. $\bar{L}_{\mathcal{L}}$ é chamada de *codificação universal* se comprime todo símbolo quase tão bem quanto a codificação em $\mathcal{L}$ que mais comprimiria tal sequência. Idealmente, espera-se encontrar uma codificação que seja universal. Isto é, uma codificação que comprima ao máximo todas as sequências dado o conjunto das

funções de tamanho de código. No entanto, não é possível obter tal função como mostrado no Teorema 1.5.

Teorema 1.5 *Se L é uma função de tamanho de código livre de prefixos associada a uma distribuição não imperfeita $\bar{P}$, então não existe uma função de tamanho de codificação $\hat{L}$ tal que $\hat{L}(X^n) = \min_{L\in\mathcal{L}} L(X^n), \forall X^n \in \mathcal{X}^n$.*

Prova: Suponha por absurdo que $\hat{L}$ exista e comprima ao máximo todas as sequências de um espaço $Z \subseteq \mathcal{X}^n$. Então, existe uma função de probabilidade $\bar{P}$ associada a $\hat{L}$. Esperar-se-ia que $\bar{L}(X^n) = -\log \bar{P}(X^n)$, mas, por definição, para toda sequência $X^n \in \mathcal{X}^n, \bar{L}(X^n) \leq \hat{L}(X^n) = -\log P(X^n \mid \theta(X^n))$. Segue então que:

$$\begin{aligned}
\bar{L}(X^n) &\leq \hat{L}(X^n) \\
-\log \bar{P}(X^n) = \bar{L}(X^n) &\leq \hat{L}(X^n) = -\log P(X^n \mid \theta(X^n)) \qquad \text{(Equação 1.2)} \\
-\log \bar{P}(X^n) &\leq -\log P(X^n \mid \theta(X^n)) \\
\bar{P}(X^n) &\geq P(X^n \mid \theta(X^n)),
\end{aligned}$$

onde $P(\cdot \mid \theta)$ é a probabilidade de um valor parametrizada por um conjunto de atributos θ, isto é, a distribuição definida por P possui parâmetros θ. Uma vez que considera-se o conjunto $\mathcal{L}$ sendo não-vazio, segue que:

$$\sum_{X^n} \bar{P}(X^n) \geq \sum_{X^n} P(X^n, \hat{\theta}(X^n)) = \sum_{X^n} \max_{\theta} P(X^n, \theta)$$

Dado que para quaisquer dois θ^1 e θ^2, tal que $\theta^1 \neq \theta^2$, $P(X^n \mid \theta^1) > P(X^n, \theta^2)$ para algum X^n, tem-se que $\sum_{X^n} \max_\theta P(X^n, \theta) > 1$, uma contradição. □

Uma vez que o Teorema 1.5 mostra que não é possível obter uma codificação universal que seja ótima para todas as codificações, procura-se por uma que seja aproximadamente tão boa quanto a melhor, que comprime todas as sequências possíveis. Isto é, procura-se por $\bar{L}(X^n) \approx \min_{L\in\mathcal{L}} L(X^n)$.

Para mitigar os problemas do MDL primitivo, então, torna-se necessário definir uma medida do quão boa uma distribuição é dado um conjunto de fontes. A partir deste conceito pode-se definir uma versão do MDL conhecida como *MDL refinado* que sana a necessidade de codificar duas partes e encontrar uma codificação para a hipótese. Chama-se de *arrependimento* (do inglês, *regret*) a diferença entre as codificações ótima para uma distribuição e uma P arbitrária. Formalmente, define-se o arrependimento como $\mathcal{R}(\bar{P}) := -\log \bar{P}(x^n) - \min_{P\in\mathcal{M}}\{-\log P(x^n)\}$, dado um conjunto de fontes de probabilidade $\mathcal{M}$. É importante que consiga-se medir o quão bom um modelo universal é; para tal, define-se como *arrependimento máximo* aquele que tem maior diferença dentre todos os símbolos a serem codificados $x^n \in \mathcal{X}^n$, isto é, $\mathcal{R}_{max}(\bar{P}) := \max_{x^n\in\mathcal{X}^n} \mathcal{R}(\bar{P})$.

Portanto, utilizando $\mathcal{R}_{max}$ como medida de ajustamento, busca-se agora pelo $\bar{P}$ que contemple $\min_{\bar{P}} \mathcal{R}_{max}(\bar{P})$.

É possível encontrar tal $\bar{P}$ definindo primeiramente o conceito de *complexidade* de um modelo $\mathcal{M}$ como $\mathbb{C}_n := \log \sum_{x^n \in \mathcal{X}^n} P(x^n \mid \hat{\theta}(x^n))$, onde $P(\cdot \mid \theta)$ é uma distribuição parametrizada por θ e $\hat{\theta}$ é o θ que maximiza a probabilidade (*e.g.*, numa distribuição normal $\theta = \langle \sigma, \mu \rangle$) (GRÜNWALD, 2005). Note que, quanto mais sequências são melhores definidas por P, maior é sua complexidade $\mathbb{C}$. Considere a *probabilidade de Shtarkov* definida como (GRÜNWALD; HARREMOËS, 2009):

$$\bar{P}(x^n) := \frac{\varphi(x^n)}{\int_{\mathcal{X}^n} \varphi(x^n) \quad \partial P(x^n)} \tag{1.5}$$

onde $\varphi(x^n) := \frac{\partial P(x^n \mid \hat{\theta}(x^n))}{\partial \bar{P}}$. Se P é discreta, como no caso que abordamos, e $\mathbb{C}$ finito, então pode-se reescrever a Equação 1.5 como segue.

$$\bar{P}(x^n) := \frac{P(x^n \mid \hat{\theta}(x^n))}{\sum_{y^n \in \mathcal{X}^n} P(y^n \mid \hat{\theta}(y^n))} \tag{1.6}$$

A partir da Equação 1.6, é possível encontrar um único $\bar{P}$. Uma prova para esta afirmação pode ser encontrada em Grünwald (2005, Proposição 2.14). Pelo Teorema 1.5, é sabido que algumas das probabilidades devem ser maior que zero, mas é possível normalizá-las obtendo uma distribuição para $\mathcal{X}^n$ dividindo-a pela soma das probabilidades. $\bar{P}$ é, então, chamada de *probabilidade máxima normalizada*. Deste modo, obtemos o MDL refinado, enunciado a seguir.

Proposição 1.3 *Dado um conjunto de dados D e um conjunto de fontes de probabilidade $\mathcal{M}$, a distribuição $\bar{P}$ que mais se ajusta ao conjunto D é dada pela maximização da probabilidade máxima normalizada de $\mathcal{M}$.*

Como mostrado por Grünwald (2005), maximizar $\bar{P}$ como descrito pela Proposição 1.3 é equivalente a minimizar a Equação 1.7.

$$-\log \bar{P}(D|M) = -\log P(D \mid \hat{\theta}^{(j)}(D)) + \mathbb{C}(M) \tag{1.7}$$

onde M é uma fonte de probabilidade em $\mathcal{M}$ e $P(\cdot \mid M)$ é a distribuição de probabilidade máxima normalizada em $\mathcal{X}^n$ correspondente à fonte M. Note que a primeira parte da soma da Equação 1.7 provê a garantia de não haver sobreajuste dos dados utilizando

a log-semelhança negativa, enquanto a segunda parte garante que o modelo não seja demasiado complexo. Deste modelo surge o *MDL refinado*, mostrado na Equação 1.8.

$$M_{MDL} := \arg\min_{M \in \mathcal{M}} \bar{L}(D|M) \tag{1.8}$$

Codificações universais são computacionalmente inviáveis de serem utilizadas, e portanto, é mais viável obter o próprio modelo e não a codificação (Van Leeuwen; SIEBES, 2008). Por este motivo, grande parte da literatura do MDL é focada no MDL em duas partes, definindo-se uma função de complexidade de modelo.

O MDL está presente em vários trabalhos na literatura, em especial naqueles relacionados diretamente à Teoria da Informação. O Princípio do MDL é, por vezes, interpretado sob uma perspectiva Bayesiana (MACKAY, 2003, Capítulo 28.3) ou comparado com o método *maximum a posteriori* (RUSSELL et al., 1995, Capítulo 20).

1.5 Considerações Finais

Neste capítulo, foram introduzidos os fundamentos elementares do MDL, Teoria da Informação e sua interrelação. O capítulo derivou conceitos que foram utilizados para obter o MDL em duas partes. Foram mostrados os limiares de tamanho de codificação, e a fórmula do MDL foi derivada seguindo os Teorema 1.1, o Teorema 1.2 e a Proposição 1.2, mostrando que com um número crescente de amostras, e uma boa estimativa da distribuição original, é possível utilizar o MDL como um método para estimativa de tamanhos de codificação.

Também, foram explorados os aspectos negativos do MDL em duas partes e foi apresentado o MDL refinado. O MDL refinado apresenta barreiras para implementação no mundo real como mostrado por Van Leeuwen e Siebes (2008), e, deste modo, esta dissertação se baseia na versão em duas partes, assim como outros trabalhos na literatura.

2 Aplicações do MDL em Aprendizado de Máquina

O MDL é um método de seleção de modelos muito versátil e poderoso. A consolidação da Navalha de Occam em um modelo matemático é atrativo para diversos contextos na computação. Desde sua criação, o MDL vem sendo utilizado em diversas pesquisas. Neste capítulo, um panorama da utilização do MDL em aprendizado de máquina é apresentado e organizado em diferentes áreas.

2.1 Árvores de Decisão

Quinlan e Rivest (1989) foram um dos primeiros a descrever uma aplicação prática do conceito do MDL em aprendizado de máquina. Os autores o utilizaram na geração de árvores de decisão, uma vez que uma árvore extremamente complexa, mesmo que com boa acurácia nos dados de treinamento, pode não possuir um bom desempenho em um conjunto de dados reais. Os autores propuseram, então, um método de construção *bottom-up*: nós folha são substituídos por nós de decisão se esta troca não aumenta o custo de transmissão da informação da árvore – baseado no MDL e definido pelos autores. No mesmo âmbito, Kononenko (1998) utilizou o MDL em duas partes para poda de árvores de decisão. O autor propôs a remoção de um nó de decisão e sua substituição por um nó folha, O critério adotado foi: se a descrição de amostras avaliadas por tal nó, caso este fosse folha, fosse menor que a soma da descrição das estruturas das sub-árvores do nó de decisão mais a descrição do subconjunto de amostras que seriam avaliadas pelas mesmas.

2.2 Decisão e Representação de Incerteza

Lam e Bacchus (1994) aplicaram o MDL no treinamento de redes Bayesianas para decisão sob incertezas, descobrindo estruturas na rede sem nenhuma informação sobre as probabilidades subjacentes do modelo. Friedman, Geiger e Goldszmidt (1997a) descreveram como utilizar o MDL para o treinamento de redes Bayesianas. Seu trabalho foi focado em classificação, elegendo a rede bayesiana que minimiza a soma do tamanho de descrição da rede e o tamanho do conjunto de dados codificados por ela.

Hinton e Zemel (1993) utilizaram o MDL juntamente com redes neurais artificiais. Os autores propuseram uma função para treinamento de auto-codificadores que minimiza o tamanho da informação requerida para codificar tanto o codificador quanto a reconstrução dos dados. Os autores comparam o método a dois comunicadores trocando informações

codificadas sob uma distribuição acordada por ambos – no caso, os autores utilizaram a distribuição de Boltzmann para determinação estocástica dos vetores de decodificação. Gregor et al. (2014) trabalharam em um viés muito similar, propondo um treinamento de auto-codificadores auto-regressivos baseado em retro-propagação ao invés de maximização de esperança, escolhendo, dentre os codificadores possíveis, aqueles que diminuem a soma de sua descrição e dos documentos codificados com o mesmo.

2.3 Processamento de Imagens

O MDL foi aplicado com sucesso em problemas relacionados com imagens e seu processamento. Potapov (2012) propôs uma representação diferente do MDL para trabalhar com imagens, nomeado pelos autores de MDL representacional (do inglês, *RMDL - representational minimum description length*). Tal método permite utilizar o MDL no problema de selecionar o critério de decisão em inferência indutiva na segmentação de imagens.

Tataw, Rakthanmanon e Keogh (2013) cunharam um método de clusterização de glifos – representando imagens como letras – que, suportado pelo MDL, obtém resultados positivos ignorando glifos desconhecidos e/ou espúrios na base treinada. Na mesma linha de pesquisa, Li, Mouchère e Viard-Gaudin (2014) propuseram um arcabouço para aprendizado não supervisionado e interativo de classificação de glifos escritos à mão. A abordagem utiliza-se do MDL para determinar a segmentação dos glifos de modo que os sub-grafos, empregados para representação dos mesmos, tenham menor tamanho de descrição.

2.4 Seleção de Atributos

A seleção automática de atributos é um problema clássico na área de aprendizado de máquina. Sheinvald, Dom e Niblack (1990) reportaram um método baseado no MDL que é capaz de eliminar atributos que sejam totalmente irrelevantes. Para isto, os autores utilizaram o critério de seleção de modelos do MDL em vetores de atributos previamente treinados representando a base alvo.

Bosin, Dessì e Pes (2006) utilizaram o MDL para a seleção de atributos de microarranjos em um cenário de alta dimensionalidade e poucas amostras. Como resultado, os autores mostraram que o MDL é equiparável a outros métodos baseados em entropia.

2.5 MDL como Técnica Auxiliar

O MDL também vem sendo amplamente empregado como técnica auxiliar, como no trabalho de Begum et al. (2013), onde os autores derivaram um critério de parada livre de

parâmetros para um aprendizado semi-supervisionado usado em séries temporais. Tataw, Rakthanmanon e Keogh (2013) descreveram um método de segmentação de símbolos em documentos antigos utilizando o MDL, onde foi possível descartar palavras raras e caracteres mal formatados. Miettinen e Vreeken (2014) descreveram uma técnica para a fatorização matricial de elementos binários. Mais recentemente, Hu et al. (2015) utilizaram o MDL para descobrir a dimensionalidade de séries temporais como uma ferramenta de representação de modelos temporais.

2.6 Categorização de Textos

O MDL foi utilizado também na categorização de textos, com aplicação na detecção de *spam*. Um método utilizando compressão dinâmica de Markov com MDL, e outro combinando entropia mínima cruzada com MDL foram propostos por Bratko et al. (2006). Em contraste, uma abordagem utilizando MDL e uma codificação adaptativa de Huffmann foi abordada por Braga e Ladeira (2008), que obtiveram resultados comparáveis àqueles considerados estado-da-arte.

Almeida (2010) e Almeida e Yamakami (2012b), Almeida e Yamakami (2012a), Almeida e Yamakami (2016) propuseram um método para detecção de *spams* baseado no MDL e fatores de confidência, chamado de *MDL-CF*, onde tais fatores representam a relevância dos termos nas amostras textuais para cada classe, o que faz com que termos (*tokens*) mais importantes para a classificação de *spam* e não-*spam* tenham um peso maior. Segundo os autores, tal método obteve resultados superiores àqueles presentes na literatura e amplamente utilizados para a mesma finalidade. Silva, Almeida e Yamakami (2016b) estenderam tal método, cunhando o `MDLText`. Os autores utilizaram a associação de pesos TF-IDF para atributos textuais e modificaram o método apresentado por Almeida (2010), tornando o método multiclasse e aplicável em qualquer problema de categorização de texto. Silva, Almeida e Yamakami (2017) propuseram uma extensão capaz de processar dados contínuos. No entanto, os autores discretizaram os atributos antes do treinamento, fazendo com que o método não pudesse ser treinado de forma incremental. Tal abordagem não foi tão eficiente quanto a utilizada para textos. Esta deficiência, naturalmente levanta a questão da possibilidade de que a discretização tenha sido uma variável problemática para a extensão do método original.

2.7 Considerações Finais

Neste capítulo, foram apresentados aplicações do MDL em diferentes áreas do conhecimento. O MDL mostrou-se uma ferramenta muito versátil, tendo sido utilizado tanto como fator principal da solução, como no caso das contruções de árvores de decisão (QUINLAN; RIVEST, 1989), tanto como uma técnica auxilixar, como quando utilizado

para seleção de critério de parada de um método semi-supervisionado (BEGUM et al., 2013).

Em especial, este capítulo apresentou o trabalho de Silva, Almeida e Yamakami (2017), que propôs uma extenção do trabalho de Almeida (2010), generalizando o método para trabalhar também com dados contínuos. No entanto, a discretização proposta pelos autores mostrou-se pouco efetiva e o desempenho ficou claramente prejudicado. A partir desta observação, uma questão natural que surge é se a normalização dos dados possa ter causado esta queda de desempenho.

3 Aprendiz de Descritores de Mistura Gaussiana

Métodos de classificação tradicionais baseiam-se, em sua maioria, na suposição de que um conjunto de dados está previamente disponível para treinamento de sua hipótese de predição, e que tais dados respeitam algum tipo de normalização prévia. No entanto, é muito comum o cenário onde há um fluxo de dados contínuo e ruidoso, no qual deseja-se que a hipótese original não só tenha um poder de generalização o suficiente para prever amostras novas não-rotuladas, como também se adaptar e melhorar de desempenho enquanto novas amostras são apresentadas. Além disso, métodos como o SVM e redes neurais, possuem um alto custo de treinamento, sofrendo pelo aumento da dimensionalidade mesmo em cenários onde o treinamento ocorre uma única vez (KRISTAN; LEONARDIS, 2014).

O MDL, como abordado no Capítulo 1, delineia a escolha de um modelo de descrição partindo do pressuposto de que quanto mais se conhece sobre um documento, mais é possível comprimi-lo. O MDL também busca por aquele modelo que tem baixa complexidade de descrição. Portanto, ele remonta a escolha de hipóteses em um cenário de aprendizado de máquina, onde procura-se idealmente por uma hipótese simples que descreva bem um conjunto de dados.

Neste capítulo, é apresentada a proposta de um novo método de classificação genérico, multinomial e de aprendizado incremental, sem necessidade de discretização dos dados, baseado no princípio do MDL: o *Aprendiz de Descritores de Mistura Gaussiana* (do inglês, *Gaussian Mixture Descriptors Learner*). Este método tem como foco mitigar as deficiências encontradas na tentativa de extensão proposta por Silva, Almeida e Yamakami (2016b) e dar mais um passo em direção à utilização do MDL como principal componente para classificação.

Este capítulo está estruturado da seguinte forma: a Seção 3.1 aborda a ideia geral do GMDL e seus componentes; as Seções 3.2, 3.3 e 3.4 tratam destes componentes e a base matemática do método. A Seção 3.5 delineia os algoritmos de treinamento e predição do GMDL.

3.1 Motivação

O MDL enuncia que se há dois modelos que descrevem um dado, aquele que o descreve de modo mais compacto deve ser escolhido (RISSANEN, 1978). Isto é, modelos mais simples são preferíveis sobre modelos mais complexos (GRÜNWALD, 2005). O MDL formaliza ideias da complexidade de Komolgorov (KOLMOGOROV, 1963) e da Navalha

de Occam (DOMINGOS, 1999) para o problema de seleção de modelos. Deste modo, elege modelos que preservam um equilíbrio favorável entre a capacidade de ajustar aos dados de treinamento e a complexidade do mesmo, evitando naturalmente o sobreajuste (*overfitting*) e, portanto, contendo características favoráveis para um método de classificação.

Codificações universais são computacionalmente proibitivas e aparecem em número diminuto de trabalhos na literatura (Van Leeuwen; SIEBES, 2008). Na versão em duas partes do MDL, em contrapartida, define-se também o tamanho da complexidade do modelo sendo aplicado, como enunciado a seguir:

$$M_{MDL} := \arg\min_{M \in \mathcal{M}} [L(M) + L(D|M)] \tag{3.1}$$

Pode-se interpretar essa fórmula como a busca pelo modelo M que mais comprima D com o menor modelo M. Como abordado no Capítulo 1, cada modelo M induz uma distribuição de probabilidade P para uma dada codificação de dados. Pelo Teorema 1.4, sabe-se que a menor codificação possível para uma determinada codificação induzida por uma distribuição de probabilidade P é definida por $\lceil -\log P \rceil$.

Idealmente, sabendo de antemão a distribuição de probabilidade real que descreve um conjunto de dados, sob uma perspectiva Bayesiana, seria possível prever com perfeição uma amostra não rotulada. No entanto, em problemas reais, este não é o caso e uma estimativa de tal distribuição é necessária. Procura-se então por alguma estimativa de distribuição de probabilidade que *(i)* evite discretizações, como aquelas apresentadas por Silva, Almeida e Yamakami (2016b) e *(ii)* seja incremental, para que se possa manter a habilidade do método aprender com novas amostras em um fluxo contínuo de dados. Deste modo, seria possível reescrever a Equação 3.1 como um método de classificação baseado em distribuições de probabilidade, onde cada modelo induz uma distribuição para cada uma das classes do problema sendo avaliado.

3.2 Função Densidade

Uma *mistura Gaussiana* é definida como a soma ponderada de G componentes Gaussianas. Cada uma destas componentes possui suas próprias médias e matrizes de covariância. Formalmente, uma mistura Gaussiana $p(X)$, para uma variável aleatória X é definida como segue.

$$p(X) := \sum_{i=1}^{G} w_i \cdot \mathcal{N}(X \mid \mu_i, \Sigma_i), \tag{3.2}$$

sendo que w é um vetor de pesos para cada componente, tal que $\sum_{i=1}^{G} w_i = 1$, com $w_i > 0$. A atribuição de pesos distintos para cada w_i permite uma maior flexibilidade na predição

em uma mistura Gaussiana, uma vez que determinada combinação de média e covariância podem ser mais descritivas que as demais, tendo assim, maior influência na computação do valor da densidade para a variável aleatória.

Um método clássico e paramétrico de se estimar uma função de densidade de probabilidade para um conjunto de dados, é assumir que a distribuição é uma mistura Gaussiana (MCLACHLAN; PEEL, 2004). No entanto, tais técnicas dependem da definição prévia dos parâmetros da mistura (MCLACHLAN; PEEL, 2004; ZIVKOVIC; HEIJDEN, 2004). Estimar tais parâmetros não é uma tarefa trivial. Por exemplo, o uso do número errado de componentes pode fazer com que o método não represente bem a função real (KRISTAN; LEONARDIS; SKOČAJ, 2011).

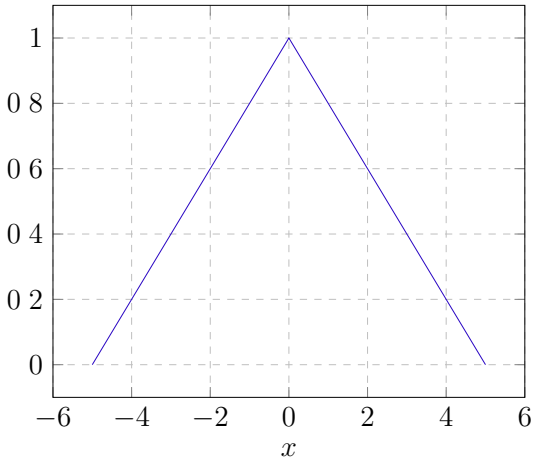
Silverman (1986) cunhou um método não-parametrizado para estimar a função de densidade de probabilidade de variáveis aleatórias chamado de *estimativa de densidade de Kernel* (EDK). Formalmente, dada uma variável aleatória Y e amostras $(Y_1, Y_2, \ldots, Y_n)$ tomadas a partir da distribuição p de Y, procura-se por uma função de distribuição de probabilidade aproximada $\hat{p}$ tal que:

$$\hat{p}(x) := \frac{1}{N} \sum_{i=1}^{N} \Phi_b(Y, Y_i), \tag{3.3}$$

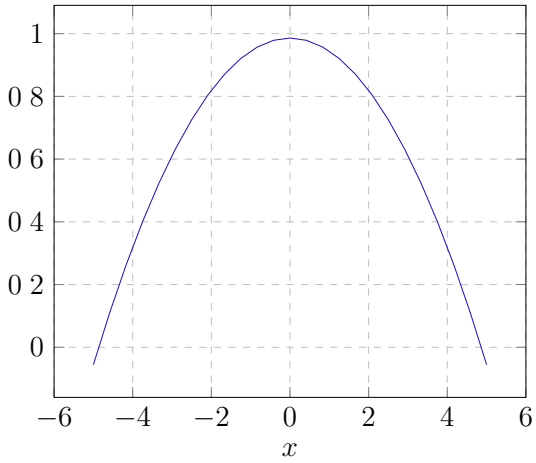
onde Φ é uma função chamada de *kernel*, que integra a um e tem média zero. É utilizado, também, um parâmetro b chamado de *largura de banda*, cujo papel é suavizar a soma dos *kernels* de $\hat{p}$. A seleção do *kernel* e também da largura de banda tem grande influência na estimativa da função (SILVERMAN, 1986). Existem várias funções de *kernel* na literatura, sendo que quatro exemplos clássicos são mostrados na Figura 5. Uma escolha clássica é o *kernel* Gaussiano (Equação 3.4), principalmente pela simplicidade de matemática.

$$\Phi_{\Sigma}^{gauss}(x, \mu) := \frac{1}{\sqrt{2\pi \cdot |\Sigma|^2}} \cdot e^{-\frac{(x-\mu)\cdot(x-\mu)^T}{2\Sigma^{-1}}} \tag{3.4}$$

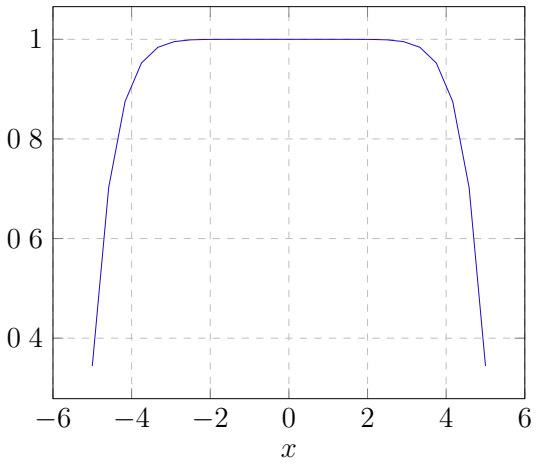
A EDK já foi utilizado na literatura na tentativa de estender métodos Bayesianos de inferência (LANGLEY; JOHN, 1995; LU; YANG; WEBB, 2006), obtendo uma troca entre acurácia de classificação e curva de aprendizado. Além disso, a EDK possui uma das características que se deseja para a distribuição apresentada na Seção 3.1: não necessitar discretizar os dados a priori. Com isto, pode-se estimar distribuições de probabilidade sem sofrer as perdas de informação oriundas de discretizações, além de manter uma estimativa da distribuição real dos dados.



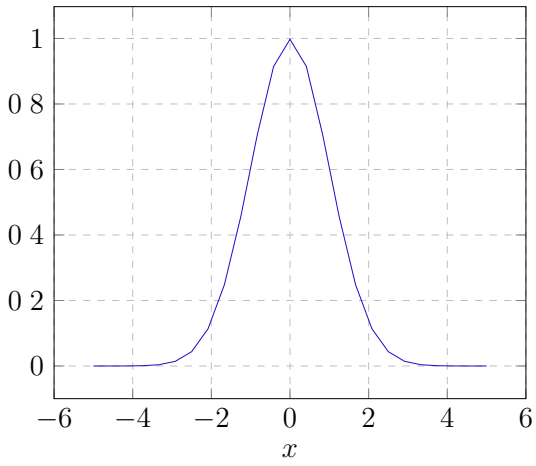
(a) Função de *kernel* triangular.



(b) Função de *kernel* Epanechnikov.



(c) Função de *kernel* tricúbico.



(d) Função de *kernel* Gaussiano.

Figura 5 – Gráficos do comportamento de diferentes funções de *kernel* no mesmo domínio.

3.2.1 Estimativa de Densidade de *Kernel* Online

Uma das grandes dificuldades em se transformar a EDK em um método incremental é que o número de componentes que ele utiliza para a estimativa da distribuição cresce linearmente para cada nova amostra apresentada (KRISTAN; LEONARDIS; SKOČAJ, 2011). Isso por que ele precisa manter informações o suficiente para generalizar para novas amostras sem rebuscar nas já vistas anteriormente (FERREIRA; MATOS; RIBEIRO, 2016). Recentemente, Kristan, Leonardis e Skočaj (2011) propuseram um método de estimativa de densidade de *kernel* totalmente incremental, chamado de *oKDE*. Portanto, esta abordagem é ideal para contemplar a segunda característica que se deseja para uma estimativa de função de distribuição de probabilidade, conforme citado na Seção 3.1.

A ideia principal por trás do oKDE é manter uma *distribuição amostral* não parametrizada. Tais distribuições, são construídas por um conjunto de amostras tratadas como funções Delta-Dirac, que podem ser definidas pela Equação 3.5. Funções Delta-Dirac possuem toda a densidade da distribuição em um único ponto, como mostrado pela

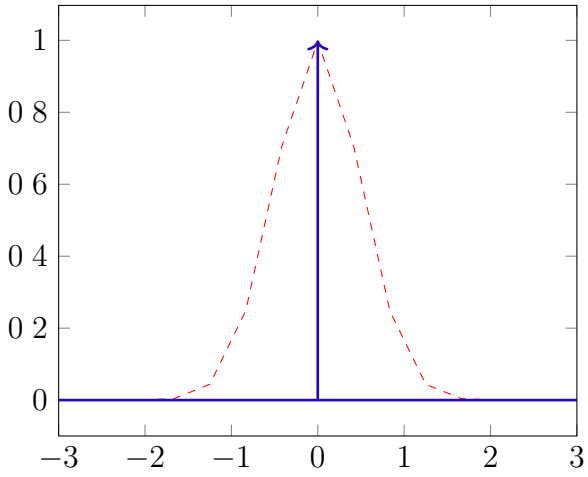


Figura 6 – Gráficos de uma função Delta-Dirac (azul sólido) de $\mu = 0$ e uma distribuição Gaussiana (vermelho cerrilhado) de $\sigma^2 = \frac{1}{2}$ e $\mu = 0$.

Figura 6, fato que as torna interessante no estudo de limitantes estatísticos.

$$p_{DD}(x) := \lim_{\sigma \to 0} \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.5}$$

Cada distribuição amostral é, portanto, um conjunto de funções de densidade infinita em um único ponto. Pode-se considerar uma distribuição amostral como uma mistura Gaussiana, formalmente definida pela Equação 3.6. Neste caso, a largura de banda é dada pela matriz de covariância da distribuição.

$$p_s(x) := \sum_{i=1}^{N} w_i \cdot \Phi_{\Sigma_{si}}(Y, Y_i), \tag{3.6}$$

onde Φ é o *kernel* Gaussiano e w é um vetor de pesos de importância para cada distribuição da mistura. Note que a Equação 3.3 é uma versão específica da Equação 3.6, quando $\Sigma_{si} = b, \forall\ (s, i)$ e $w_i = \frac{1}{N}, \forall\ i$.

Uma vez que manter uma função Delta-Dirac para cada observação não é adequado para o cenário incremental, os autores comprimem o número de componentes utilizando um algoritmo de agrupamento que aproxima uma distribuição Gaussiana dados outros pontos. Esse agrupamento se dá pela redução de um conjunto n-dimensional para um n'-dimensional, tal que $n' < n$, onde procura-se por uma estimativa de distribuição que não ultrapasse uma dada taxa de erro $D_t h$. Em seus experimentos, a taxa de 0,1 foi a que atingiu os melhores resultados.

Durante a estimativa do oKDE, também é utilizado um parâmetro f chamado de *fator de esquecimento*. Tal fator é utilizado para atribuir um peso a amostras antigas, fazendo com que possuam menos impacto em um fluxo de dados – ideal para cenários

onde o fluxo é temporal. Um fator de esquecimento igual a 1, significa que não há peso diferente entre as amostras temporalmente.

O segundo ponto importante na descrição do oKDE é a estimativa da largura de banda ótima. Como abordado na Seção 3.2, a qualidade da estimativa de banda tem impacto direto na qualidade da distribuição aproximada. Pode-se enxergar a estimativa da largura de banda como um problema de otimização onde procura-se minimizar a distância da distribuição aproximada para a distribuição real (LÜTHKE, 2013). Usualmente, neste cenário, utiliza-se a divergência de Kullback-Leibler como medida de distância (KRISTAN; LEONARDIS; SKOČAJ, 2011). No entanto, a distribuição original não está disponível. Para tais cenários, uma abordagem tradicional é utilizar o erro quadrádico médio assintótico integrativo (do inglês, *asymptotic mean integrated squared error* – *AMISE*) (WAND; JONES, 1994), utilizada pelo oKDE para definir a melhor largura de banda.

Ferreira, Matos e Ribeiro (2016) cunharam uma versão do oKDE, chamada de *xokde++*. Este trabalho é uma extensão da proposta original focando em sanar problemas de inconsistência numérica e propor uma abordagem computacional mais robusta em termos de uso de memória e processamento. Nesta abordagem, mantém-se uma distribuição normalizada, isto é, todas as estimativas tem densidade limitada superiormente por um – uma propriedade extremamente útil, uma vez que pode ser mapeada para uma estimativa de probabilidade. Esta dissertação utiliza-se desta implementação. Por simplificação, "oKDE" será utilizado intercambiavelmente.

3.2.2 Predição Incremental de Amostras

Uma vez que o oKDE provê uma maneira não-parametrizada e incremental de estimar uma função de densidade dado um conjunto de dados, naturalmente temos um candidato para os modelos a serem considerados. A Proposição 1.2 garante que nenhuma distribuição aproxima tão bem um conjunto de dados como a distribuição no qual eles são definidos. Portanto, obtendo uma aproximação desta distribuição pelo oKDE, pode-se esperar que com o número crescente de amostras, pelo Teorema 1.2, obtenha-se uma estimativa cada vez mais fidedigna da distribuição real. Deste modo, pode-se definir a função de tamanho de descrição $\hat{L}$ de dados como a soma das descrições de seus atributos quando codificados pela aproximação de suas funções densidades p' para cada classe, formalizada pela Equação 3.7.

$$\hat{L}(\vec{x}|c) := \sum_{i=1}^{n} \lceil -\log p'_{(i,c)}(\vec{x}_i) \rceil, \tag{3.7}$$

tal que $\vec{x}$ é um vetor $(\vec{x}_1, \cdots, \vec{x}_n)$ de atributos e $c \in K$, a classe avaliada. Note que $\hat{L}(\vec{x}|c) \in \mathbb{N}$, e que p' pode tender a infinito no caso onde uma função Delta-Dirac seja

avaliada; e pode ser avaliada a zero, caso não haja amostras o suficiente para medir a densidade em um ponto. Portanto, a Equação 3.7 pode ser reescrita como:

$$\hat{L}(\vec{x}|c) := \sum_{i=1}^{n} \lceil -\log p_{(i,c)}(\vec{x}_i) \rceil \tag{3.8}$$

onde

$$p_{(i,c)} := \begin{cases} 2^{-\Omega} & p'_{(i,c)} \to \infty \ \vee \ p'_{(i,c)} = 0 \\ p'_{(i,c)} & c.c. \end{cases} \tag{3.9}$$

tal que Ω é um meta-parâmetro que funciona como regularizador nos casos onde não há informação o suficiente para computar-se uma probabilidade. Tal meta-parâmetro assemelha-se àquele de mesmo nome utilizado por Silva, Almeida e Yamakami (2017). É possível também alterar o fator de esquecimento f original no oKDE como um meta-parâmetro de p'; mas este será omitido por simplicidade de notação.

Uma vez que $p'_{(\cdot,c)}$ é uma função de densidade aproximada pelo oKDE, ela é uma mistura Gaussiana. Portanto, é composta por um número finito de distribuições que quanto menos relacionadas, maior número de funções Delta-Dirac possuirão e, consequentemente, maior número de componentes. Esta observação torna $|p'_{(\cdot,c)}| = G_c$ (isto é, o número de componentes da mistura Gaussiana para a classe c) um bom candidato para representar a complexidade do modelo, uma vez que quanto maior o número de componentes, maior a descrição do modelo. Temos, então, um candidato para $L(M)$ de um dado modelo M. Portanto, pode-se definir o $GMDL'$ como:

$$GMDL'(\vec{x}, K) := \arg\min_{c \in K} [\hat{L}(\vec{x}|c) + G_c] \tag{3.10}$$

É possível normalizar a Equação 3.10 provendo melhor estabilidade numérica e, convenientemente, uma pseudo-probabilidade quando avaliada como $1 - GMDL(\vec{x})$. Pode-se, finalmente, obter a primeira versão do GMDL:

$$GMDL(\vec{x}) := \begin{bmatrix} GMDL'(\vec{x}, \{c_1\}) \\ GMDL'(\vec{x}, \{c_2\}) \\ \vdots \\ GMDL'(\vec{x}, \{c_k\}) \end{bmatrix} \Bigg/ \sum_{k=1}^{|K|} GMDL'(\vec{x}, \{c_k\}) \tag{3.11}$$

Como apontado por Grünwald (2005), a otimização proposta pela Equação 3.11 é muito similar ao método de estimativa de verossimilhança máxima. Note que, no entanto, esta estimativa está limitada por logaritmos e também por uma normalização entre

classes. Deste modo, pode-se considerar que esta base do GMDL é uma estimativa de verossimilhança máxima normalizada.

3.2.3 Suavização de Funções Delta-Dirac e Distribuições Degeneradas

Originalmente, o método proposto por Kristan, Leonardis e Skočaj (2011) era suscetível a distribuições degeneradas. Uma distribuição degenerada é, em linhas gerais, uma distribuição com baixa variância, formalmente definida a seguir.

Definição 3.1 *Seja $X \sim \mathcal{N}(\mu, \Sigma)$ uma variável aleatória sob uma distribuição p_X. Se $\|diag(\Sigma)\|_\infty \approx 0$, então p_X é uma distribuição degenerada.*

O método refinado por Ferreira, Matos e Ribeiro (2016) tentou mitigar além da estabilidade numérica, este problema em dimensões degeneradas. No entanto, os autores utilizaram o logaritmo da matriz de covariância para suas computações. Utilizar-se de logaritmos é uma técnica amplamente conhecida em Ciência da Computação para evitar problemas multiplicativos, uma vez que $\log(ab) = \log a + \log b$, o que mantém a estabilidade numérica na operação. Porém, $\log 0$ tende a infinito, uma indefinição teórica que faz com que o método seja propenso a erros. A técnica proposta por Ferreira, Matos e Ribeiro (2016) para minimizar o problema de distribuições degeneradas é computar sua decomposição em autovalores e auto-vetores, e analisar seus autovalores procurando por aqueles menores que 10^{-9}, os quais são corrigidos por 1% da média dos autovalores. No entanto, essa técnica é prejudicial quando se tem apenas uma dimensão, uma vez que fica extremamente dependente da ordem em que as amostras são apresentadas. Por exemplo, caso um sequência de amostras de mesmo valor seja apresentada, sua variância seria zero e o método não conseguiria se autocorrigir, dado que o autovalor da matriz de covariância também seria zero.

Para mitigar este problema, um ruído Gaussiano é adicionado à entrada na estimativa de uma densidade de probabilidade. Essa adição pode interferir diretamente na forma de uma distribuição. No entanto, é possível mostrar que a adição de um ruído Gaussiano mantém a suposição que as densidades podem ser expressas por uma mistura Gaussiana.

O *momento* de uma variável aleatória X é definido como sua esperança em sua n-ésima potência. Por exemplo, o momento um representa a média de X, enquanto seu momento dois representa sua variância. É possível definir uma função de distribuição de probabilidade com base em seu momento; tal função é chamada de *função geradora de momento*, definida a seguir.

Definição 3.2 *Seja X uma variável aleatória sob uma distribuição p_X, a função geradora de momento define unicamente p_X e é definida como $M_X(s) = E[e^{sX}]$.*

Com base no conceito de função geradora de momento, pode-se mostrar que a adição de um ruído Gaussiano não altera a propriedade de que a mistura oriunda do oKDE é, também, Gaussiana. Tal afirmação é provada pelo Teorema 3.1.

Teorema 3.1 *Se* $X \sim \mathcal{N}(\mu_X, \sigma_X^2)$ *e* $Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$ *são duas variáveis aleatórias independentes, então* $(X + Y) \sim \mathcal{N}(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2)$.

Prova: Pela Definição 3.2, a função geradora de momento da distribuição normal $N \sim \mathcal{N}(\mu, \sigma^2)$ é dada por:

$$\begin{aligned} M_N(s) &= E[e^{sN}] \\ &= \int_{-\infty}^{\infty} e^{sN} \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(N-\mu)^2}{2\sigma^2}} dN \\ &= e^{s\mu + \frac{s^2\sigma^2}{2}} \end{aligned}$$

Portanto, tem-se que as função geradoras de momento de X e Y são, respectivamente,

$$M_X(s) = e^{s\mu_X + \frac{s^2\sigma_X^2}{2}}$$

e

$$M_Y(s) = e^{s\mu_Y + \frac{s^2\sigma_Y^2}{2}}.$$

Uma vez que X e Y são independentes, pode-se definir a distribuição obtida por sua soma como a multiplicação de suas funções geradoras de momento. Isto é,

$$\begin{aligned} M_{(X+Y)}(s) &= e^{s\mu_X + \frac{s^2\sigma_X^2}{2}} \cdot e^{s\mu_Y + \frac{s^2\sigma_Y^2}{2}} \\ &= e^{s(\mu_X+\mu_Y) + \frac{s^2(\sigma_X^2+\sigma_Y^2)}{2}} \end{aligned}$$

Não é difícil de notar que $M_{(X+Y)}$ é uma função geradora de momento de uma distribuição normal com média $(\mu_X + \mu_Y)$ e desvio padrão $(\sigma_X^2 + \sigma_Y^2)$. □

Portanto, a adição de ruído $\mathcal{N}(0, \tilde{\sigma}^2)$ mantém o centro da distribuição e altera apenas seu desvio padrão. A primeira característica é muito útil, uma vez que, por exemplo, para funções Delta-Dirac não altera o fato que o ponto central de tal função é onde há o maior acúmulo de densidade. No entanto, alterar o desvio padrão também altera a densidade dos 68% dos dados em volta da média. Deste modo, o ruído é usado seletivamente quando a variância de um atributo em uma dada classe ficaria menor do que o limiar de 10^{-9} definido por Ferreira, Matos e Ribeiro (2016). Este processo, dado o meta-parâmetro $\tilde{\sigma}^2$, pode ser definido por

$$p'_{(i,c)} := \begin{cases} \vec{x}_i & \sigma_{ic}^2 \geq 10^{-9} \\ \vec{x}_i + \sum_{n \sim \mathcal{N}(0,\tilde{\sigma}^2)} n, \sigma_{ic}^2 > 10^{-9} & c.c. \end{cases}.$$

A computação incremental da variância é feita por uma aproximação baseada no acúmulo da média. Este método foi criado por Welford (WELFORD, 1962) e posteriormente refinado por Ling (1974) e Chan, Golub e LeVeque (1983). Ele se baseia em um fluxo contínuo de dados de uma variável aleatória. No caso específico do GMDL, esse fluxo é dado pelas amostras para cada atributo i da classe c. O cálculo é definido pela seguinte recorrência:

$$
\begin{aligned}
M_{2,n_{ic}} &= M_{2,n_{ic}-1} + (x_{n_{ic}} - \bar{x}_{n_{ic}-1}) \cdot (x_{n_{ic}} - \bar{x}_{n_{ic}}) \\
\bar{x}_{n_{ic}} &= \bar{x}_{n_{ic}-1} + \frac{x_{n_{ic}} - \bar{x}_{n_{ic}-1}}{n_{ic}} \\
\sigma^2_{ic} &= \frac{M_{2,n_{ic}}}{n_{ic}}
\end{aligned}
$$

onde $M_{2,n_{ic}}$ é a soma das diferenças quadráticas em relação a média até a n_{ic}-ésima amostra. Deste modo, guarda-se os valores da quantidade de amostras n_{ic}, $M_{2,n_{ic}}$ e $\bar{x}_{n_{ic}}$ para cada atributo i de cada classe c, para então computar σ^2_{ic} a fim de determinar quando aplicar o ruído.

3.3 Ponderação da Importância dos Atributos

Seleção de atributos é uma técnica clássica da área de aprendizado de máquina. Ela é usada, por vezes, para se combater a *maldição da dimensionalidade*. Tal fenômeno teve sua primeira aparição na literatura por Bellman (1957), em seu estudo de otimização dinâmica. Em linhas gerais, este problema ocorre ao analizar-se conjuntos de dados onde os atributos estão em uma ordem de grandeza considerável. Neste cenário, são necessárias muitas amostras para que o poder preditivo dos métodos não se degrade, ao mesmo tempo que, dado o número de atributos, o processamento escala exponencialmente no número de combinações possíveis entre amostras e atributos.

A seleção de atributos é, também, empregada na tentativa de reduzir a variância em um conjunto de treinamento para encontrar modelos mais simples que descrevem um conjunto de dados. A *engenharia de atributos*, como é chamado tal processo, parte do pressuposto que existem dados que podem ser redundantes – por exemplo, uma variável contínua de baixa variância – ou extremamente preditivos, neste caso apelidados de *atributos dourados* (do inglês, *golden features*).

Muitos trabalhos neste âmbito vêm sido estudados. O método da *Análise de Componentes Principais* (PCA), criado por Pearson (1901), é um método muito utilizado na literatura para redução de dimensionalidade. Seu funcionamento baseia-se numa transformação linear ortogonal que mapeia os dados de sua dimensionalidade original para outra dimensionalidade menor, onde suas componentes os descrevem em ordem decrescente de variância. Outras técnicas também são comuns na literatura na tentativa de encontrar

atributos mais preditivos, como o *ganho de informação* e a *informação mútua*, ambos baseados em entropia, e a estatística χ^2 que calcula a dependência entre um atributo e a classe.

Métodos baseados em predição Bayesiana assumem que os atributos de uma amostra são independentes, e assim podem relaxar a fórmula Bayesiana – daí o nome de Bayes Ingênuo (do inglês, *Naïve Bayes*). Esta suposição é uma maneira de combater a alta dimensionalidade que geralmente aparece em bases de dados. No entanto, na maioria dos casos essa suposição é violada, o que leva os métodos a uma predição subótima (ZAIDI et al., 2013). Na literatura, existem, principalmente, duas abordagens para tentar melhorar a preditividade de tais métodos: *métodos semi-Naïve Bayes* e *métodos de ponderação de atributos* (ZAIDI et al., 2013). Os primeiros, como abordados por Langley e Sage (1994) e Friedman, Geiger e Goldszmidt (1997b), tentam aliviar a suposição de independência de atributos inserindo fatores e selecionando um conjunto de atributos que sejam potencialmente independentes. Os últimos foram pouco explorados na literatura e foram vistos mais como métodos de seleção de atributos importantes. No entanto, pesquisas recentes mostraram que tal estratégia pode prover resultados equivalentes ou melhores que os demais de forma menos custosa (ZAIDI et al., 2013; XIANG et al., 2014).

A Equação 3.11 é baseada na soma das densidades de probabilidade, que chamamos de tamanho de descrição. No entanto, a estimativa oriunda do oKDE baseia-se na premissa de que as distribuições dos atributos para uma dada classe são independentes, uma vez que as mesmas são estimadas separadamente. Portanto, ao invés de tentar utilizar coeficientes de preditividade de atributos, pesos nos atributos são utilizados para a limitação de tal suposição, como feito por Zaidi et al. (2013). Esta técnica se baseia na otimização desses pesos, cujo objetivo é melhorar o desempenho do método de classificação como um todo.

Inicialmente, a Equação 3.8 é alterada adicionando um peso θ para cada atributo, como mostrado a seguir.

$$\hat{L}(\vec{x}|c) := \sum_{i=1}^{n} \lceil -\log p_{(i,c)}(\vec{x}_i) \rceil^{\theta_i} \tag{3.12}$$

onde Θ é um vetor de n elementos representando pesos nos atributos. Por simplicidade de notação, $\hat{L}(\vec{x}|c)$ será usado como:

$$\hat{L}(\vec{x}|c) \triangleq \sum_{i=1}^{n} \varphi_{ic}^{\theta_i}. \tag{3.13}$$

Procura-se, então, encontrar valores para Θ que minimizem alguma função de erro. Assim como empregado por Zaidi et al. (2013), o erro quadrádico médio é utilizado como função de erro, e deste modo, é possível tratá-lo como um problema de otimização. Para isto precisa-se, primeiramente, definir a função de custo $J(\Theta)$, como segue.

$$J(\Theta) = \frac{1}{2}\sum_{j=1}^{m}\sum_{k=1}^{|K|}(\delta_{y^{(j)}c_k} - L(\vec{x}^{(j)}|c_k) - G_{c_k})^2 \tag{3.14}$$

onde δ é o delta de Kronecker. Isto é, uma função δ_{ij} que tem valor 1 se $i = j$ e 0, caso contrário. Antes de calcular a derivada da função de custo, é útil encontrar as derivadas das componentes $\frac{\partial \hat{L}(\vec{x}|c_j)}{\partial \theta_i}$ e $\frac{\partial L(\vec{x}|c_j)}{\partial \theta_i}$.

A derivada de $\frac{\partial \hat{L}(\vec{x}|c_j)}{\partial \theta_i}$ para a matriz Jacobiana $\frac{\partial \hat{L}(\vec{x})}{\partial \Theta}$ é:

$$\begin{aligned}\frac{\partial \hat{L}(\vec{x}|c_j)}{\partial \theta_i} &= \frac{\partial}{\partial \theta_i}\sum_{i'=1}^{n}\varphi_{i'j}^{\theta_i} \\ &= \varphi_{ij}^{\theta_i}\cdot\ln\theta_i\end{aligned}$$

De modo similar, a derivada de $\frac{\partial L(\vec{x})}{\partial \Theta}$ para a matriz Jacobiana $\frac{\partial L(\vec{x})}{\partial \Theta}$ é:

$$\begin{aligned}\frac{\partial L(\vec{x}|c_j)}{\partial \theta_i} &= \frac{\partial}{\partial \theta_i}\frac{\sum_{i=1}^{n}\varphi_{ij}^{\theta_i}}{\sum_{k=1}^{|K|}\sum_{i'=1}^{n}\varphi_{i'k}^{\theta_{i'}}} \\ &= \frac{\varphi_{ij}^{\theta_i}\cdot\ln\theta_i\cdot(\sum_{k=1}^{|K|}\sum_{i'=1}^{n}\varphi_{i'k}^{\theta_{i'}}) - \varphi_{ij}^{\theta_i}\cdot\ln\theta_i\cdot\hat{L}(\vec{x}|c_j)}{(\sum_{k=1}^{|K|}\sum_{i'=1}^{n}\varphi_{i'k}^{\theta_{i'}})^2} \\ &= \varphi_{ij}^{\theta_i}\cdot\ln\theta_i - \varphi_{ij}^{\theta_i}\cdot\ln\theta_i\cdot L(\vec{x}|c_j) \\ &= \varphi_{ij}^{\theta_i}\cdot\ln\theta_i\cdot(1 - L(\vec{x}|c_j))\end{aligned}$$

Com estes resultados parciais, pode-se encontrar o gradiente da função de custo, como mostrado a seguir.

$$\begin{aligned}\nabla J_{\theta_i} &= \frac{\partial}{\partial \theta_i}\frac{1}{2}\sum_{j=1}^{m}\sum_{k=1}^{|K|}(\delta_{y^{(j)}c_k} - L(\vec{x}^{(j)}|c_k))^2 \\ &= -\sum_{j=1}^{m}\sum_{k=1}^{|K|}(\delta_{y^{(j)}c_k} - L(\vec{x}^{(j)}|c_k))\cdot\frac{\partial L(\vec{x}^{(j)}|c_k)}{\partial \theta_i} \\ &= -\sum_{j=1}^{m}\sum_{k=1}^{|K|}(\delta_{y^{(j)}c_k} - L(\vec{x}^{(j)}|c_k))\cdot\varphi_{ik}^{\theta_i}\cdot\ln\theta_i\cdot(1 - L(\vec{x}^{(j)}|c_k))\end{aligned} \tag{3.15}$$

Com base no gradiente da Equação 3.15, pode-se então computar, em tempo de execução, os valores para Θ que minimizam o erro da predição. Inicialmente, pondera-se todos os pesos a $0,\bar{9}$. Note que, $\Theta \in (0, 1)$, uma vez que $\log 1 = 0$, e, portanto, o método teria custo zero inicialmente e $\log 0 \to \infty$ é um valor que não faz sentido teórico para o problema.

É possível otimizar Θ utilizando-se de qualquer método de otimização. Um método clássico na literatura é o *gradiente descendente*. A ideia do método é dada uma função $f(x)$, se ela é definida e diferenciável em uma vizinhança de um ponto v, então f decresce mais rápido se seguir na direção do gradiente negativo de f em v. Isto é, se $v_{n+1} = v_n - \eta \nabla f(v_n)$, para um η suficientemente pequeno, então o valor de f em v_{n+1} é menor que em v_n. O valor η é popularmente chamado de *taxa de aprendizado* e é um meta-parâmetro a ser ajustado. Este processo é ilustrado pela Figura 7.

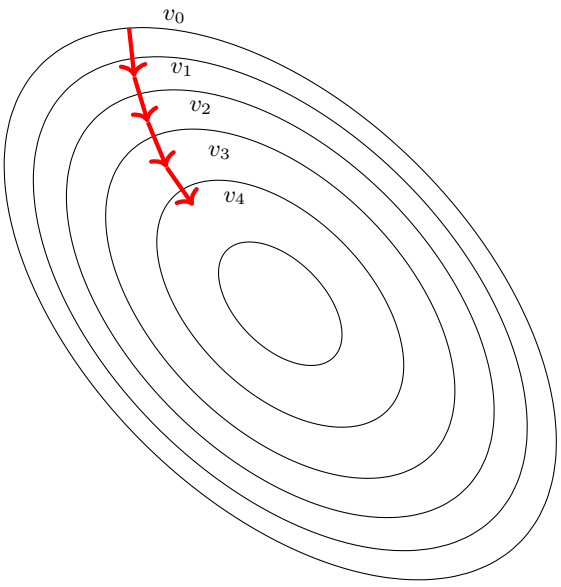


Figura 7 – Ilustração do funcionamento do Gradiente Descendente em um conjunto de pontos v.

O método do gradiente descendente não é apropriado no cenário onde as amostras vem de um fluxo contínuo de dados, uma vez que baseia-se na otimização dado um conjunto de amostras. No entanto, o método conta com uma variação chamada de *gradiente descendente estocástico* que, ao invés de computar o gradiente exato com base em um conjunto de amostras, estima-o com base em uma única amostra. No entanto, esta formulação do gradiente descendente pode acarretar baixa precisão e convergir em situações de escolha de η moderadas (BOTTOU, 2010). A ideia de *momento* foi descrita por Rumelhart, Hinton e Williams (1986) para o caso de redes neurais. Assim como o momento da física, o momento na função gradiente tende a fazer com que o gradiente sofra menos alterações em suas atualizações. A atualização do Θ é dada então por $\theta_i = \theta_i - \Delta\theta_i$, com $\Delta\theta_i := \eta \nabla J_{\theta_i} + \alpha \Delta\theta_i$. Nesta última equação, a variável α é o meta-parâmetro de momento do gradiente descendente. Note que $\Delta\theta_i$ é computado no início de cada iteração, para que o método possa ajustar seus pesos dado uma amostra e sua predição.

3.4 Protótipo de Fronteira de Separação de Classes

Em problemas de classificação, a *fronteira* de uma barreira de decisão é a distância da barreira até os dados separados. O SVM é um método amplamente conhecido que busca aumentar a fronteira de separação entre os dados, atingindo assim, predições mais confiáveis (SARAVANAN; SIDDABATTUNI; RAMACHANDRAN, 2008). Na definição do método baseado no MDL para classificação de textos de Silva, Almeida e Yamakami (2017), chamado de `MDLText`, os autores propuseram um fator de penalidade $\hat{S}$ para distanciar as classes menos prováveis para uma dada amostra baseada na similaridade de cosseno. Isto é possível uma vez que os autores conseguem um *protótipo de classe* baseado na contagem de termos; então a distância de cosseno é uma escolha apropriada para a distância de ponto a ponto.

Dada a eficiência das predições do `MDLText`, uma característica interessante de se possuir no GMDL seria também a penalização de classes mais distantes para a computação do tamanho de descrição. No entanto, como o GMDL está baseado em estimativa de função de densidade, não é possível utilizar a distância de cosseno. Sendo assim, é necessária a utilização de uma função de distância entre ponto e distribuição. Uma escolha comum neste cenário é a *distância de Mahalanobis* (MAHALANOBIS, 1936), que é uma generalização da métrica da quantidade de desvios padrões que um ponto está da média da distribuição sendo avaliada, variando no domínio $[0, \infty)$. Esta medida é dada pela equação a seguir.

$$D(\vec{x}, f) := \sqrt{(\vec{x} - \mu_f) \cdot \Sigma_f^{-1} \cdot (\vec{x} - \mu_f)^T}$$

A ideia geral é que não só se está medindo a distância de um ponto ao centro de uma distribuição, mas também considera-se a esparsidade dos dados em relação à média. A matriz de covariância mede o quão espalhados estão os dados em relação às suas médias e como variam em conjunto com as demais dimensões. A inversa da matriz de covariância mede, então, o quão próximos à média das componentes estão os dados, isto é, quanto mais próximos estiverem de suas respectivas médias – ou o quão mais próximo da identidade está Σ^{-1} – maior o impacto na distância de Mahalanobis.

Ainda é necessário definir o protótipo de classe para cada uma das classes computadas. Uma aproximação pode ser feita definindo-se uma distribuição multivariada para cada classe, baseando-se em todos os conjunto de atributos ao invés de considerá-los individualmente, como feito para calcular o tamanho de descrição. Deste modo, mantém-se também $|K|$ misturas Gaussianas multivariadas como protótipos de classes obtidas com o oKDE. É possível definir, então, um protótipo de classe como a 2-tupla $\tilde{c} := \langle \bar{\mu}_c, \bar{\Sigma}_c \rangle$ para uma dada classe c.

Como tratam-se de misturas, é necessário encontrar seus vetores de média e matriz

de covariância antes de aplicar a distância de Mahalanobis. A média é simplesmente o valor esperado das distribuições, isto é:

$$\bar{\mu}_c := \sum_{i=1}^{G_c} w_i \cdot \mu_c^i.$$

A variância, em contraste, pode ser definida em condição à relação das médias como:

$$\bar{\Sigma}_c := \sum_{i=1}^{G_c} w_i \cdot \Sigma_c^i + \sum_{i=1}^{G_c} w_i \cdot (\mu_c^i - \bar{\mu}_c^i) \cdot (\mu_c^i - \bar{\mu}_c^i)^T.$$

Uma vez computado os valores a todos os protótipos, uma normalização entre zero e um dos dados é feita, tal que D igual a zero representa baixa similaridade e D igual a um, a classe mais similar. Assim, pode-se definir $\hat{S}$ de modo semelhante a Silva, Almeida e Yamakami (2017), como:

$$\hat{S}(\vec{x}, \tilde{c}) := -\log\left(\frac{1 - D(\vec{x}, \tilde{c}) + 2^{\beta}}{2}\right)$$

Note que, para evitar inconsistência, um meta-parâmetro β é adicionado. Ele é necessário dada a normalização efetuada, já que sempre que $\hat{S}$ é computado, este valor tenderia a infinito para a classe mais semelhante. Deste modo, pode-se redefinir a Equação 3.12 como:

$$\hat{L}(\vec{x}|c) := \hat{S}(\vec{x}, \tilde{c})^{\tau} \cdot \sum_{i=1}^{n} \lceil -\log p_{(i,c)}(\vec{x}_i) \rceil^{\theta_i} \tag{3.16}$$

Assim, penaliza-se diretamente o tamanho da descrição de um vetor de atributos dada sua similaridade às distribuições das classes: a classe que mais difere da amostra sendo classificada terá seu tamanho de descrição multiplicado por β, enquanto a mais semelhante não sofrerá alteração no tamanho de código. Diferentemente da metodologia adotada por Silva, Almeida e Yamakami (2017), foi adicionado também um meta-parâmetro τ, que permite um ajuste fino no peso dado à distância aos protótipos.

3.5 Visão Geral do Método Proposto

O método proposto apenas necessita de estimativas de densidade de funções de probabilidade na fase de treinamento. Estas estimativas são feitas de maneira incremental e sem discretização pelo oKDE. Para cada uma das classes, são estimadas suas distribuições multivariadas chamadas de protótipos, e distribuições individuais para cada atributo.

A fase de predição do método baseia-se, principalmente, nos seguintes blocos:

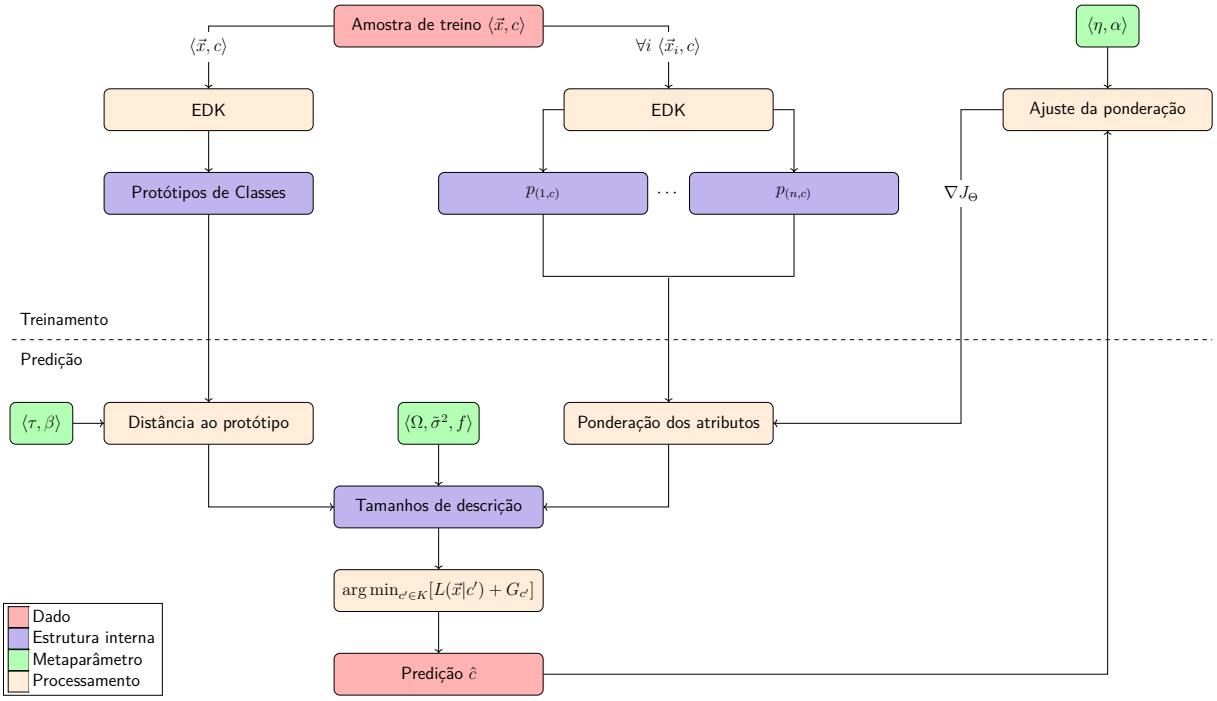


Figura 8 – Diagrama de bloco do funcionamento das fases de treinamento e predição do GMDL.

1. Na avaliação de densidade para cada atributo da amostra;

2. Na distância do conjunto de atributos aos protótipos de classes;

3. Na ponderação dos atributos.

A interação destes processos é ilustrada pela Figura 8 na forma de um diagrama de blocos: Dada uma amostra, são estimados diferentes funções densidade tanto para o cálculo da distância ao protótipo quanto para cada atributo; com tais densidades, calcula-se então o tamanho da descrição da amostra em cada classe, e a classe com menor descrição é escolhida como rótulo para a amostra; tal rótulo é utilizado para o cálculo do peso dos atributos. Note que, apesar do ajuste de ponderação fazer parte do treinamento, ele somente pode ser computado dado o rótulo que o método previu para uma determinada amostra, como explicitado pela Equação 3.15.

Os Algoritmos 1 e 2 ilustram, respectivamente, os passos para os estágios de treinamento e predição do GMDL.

O código fonte do GMDL, escrito na linguagem de programação *C++*, está disponível publicamente na plataforma *GitHub*[1].

[1] Código fonte do GMDL. Disponível em <https://github.com/brenolf/gmdl>, acessado em 25 de Novembro de 2017.

Algoritmo 1 Estágio de treinamento do GMDL.

Entrada: Amostra $\langle \vec{x}, c \rangle$, classe prevista $\hat{c}$, se houver, e metaparâmetros $\langle \Omega, \tilde{\sigma}^2, f, \eta, \alpha \rangle$

$\tilde{c} \leftarrow p_c(\vec{x}; f)$
para $i \in \{1, \ldots, n\}$ **faça**
  se $p'_{(i,c)} \to \infty \ \vee \ p'_{(i,c)} = 0$ **então**
    $p_{(i,c)} \leftarrow 2^{-\Omega}$
  senão
    enquanto $\sigma^2_{ic} \geq 10^{-9}$ **faça**
      $\vec{x}_i \leftarrow X_1, X \sim \mathcal{N}(0, \tilde{\sigma}^2)$
    fim enquanto
    $p_{(i,c)} \leftarrow p'_{(i,c)}(\vec{x}_i; f)$ ▷ Equação 3.9
  fim se
  $n_{ic} \leftarrow n_{ic} + 1$
  $M_{2,n_{ic}} \leftarrow M_{2,n_{ic}-1} + (\vec{x}_i - \bar{x}_{n_{ic}-1}) \cdot (\vec{x}_i - \bar{x}_{n_{ic}})$
  $\bar{x}_{n_{ic}} \leftarrow \bar{x}_{n_{ic}-1} + \frac{\vec{x}_i - \bar{x}_{n_{ic}-1}}{n_{ic}}$
  $\sigma^2_{ic} \leftarrow \frac{M_{2,n_{ic}}}{n_{ic}}$
fim para
para $i \in \{1, \ldots, n\}$ **faça**
  $\varphi_{i\hat{c}} \leftarrow \lceil -\log p_{(i,\hat{c})}(\vec{x}_i) \rceil$
  $\nabla J_{\theta_i} \leftarrow (\delta_{i\hat{c}} - L(\vec{x})|\hat{c})) \cdot \varphi^{\theta_i}_{i\hat{c}} \cdot \ln \theta_i \cdot (1 - L(\vec{x})|\hat{c}))$
  $\Delta\theta_i \leftarrow \eta \nabla J_{\theta_i} + \alpha \Delta\theta_i$
  $\theta_i \leftarrow \theta_i - \Delta\theta_i$
fim para

Algoritmo 2 Estágio de predição do GMDL.

Entrada: Amostra $\vec{x}$ e metaparâmetros $\langle \tau, \beta \rangle$
Saída: Classe prevista $\hat{c}$

para $c \in K$ **faça**
  para $i \in \{1, \ldots, n\}$ **faça**
    $\hat{L}(\vec{x}, c) \leftarrow \hat{L}(\vec{x}, c) + \lceil -\log p_{(i,c)}(\vec{x}_i) \rceil^{\theta_i}$
  fim para
  $\hat{L}(\vec{x}, c) \leftarrow (\hat{L}(\vec{x}, c) + G_c) \cdot \hat{S}(\vec{x}, \tilde{c}; \beta)^{\tau}$
fim para
para $c \in K$ **faça**
  $L(\vec{x}, c) \leftarrow \frac{\hat{L}(\vec{x}, c)}{\sum_{k=1}^{|K|} \hat{L}(\vec{x}|c_k)}$
fim para
retorna $\hat{c} \leftarrow \arg\min_{c \in K} L(\vec{x}|c)$

3.5.1 Análise de Complexidade Assintótica

O estágio de treinamento do GMDL baseia-se na computação de estimativas de densidades para uma única classe em seus n atributos. A complexidade do oKDE é limitada por $O(\frac{G^2-G}{2+G})$, onde G é o número de componentes da mistura Gaussiana avaliada. Deste modo, o treinamento tem complexidade quadrática em relação ao número de componentes na mistura, que foi validada experimentalmente baixa por Kristan, Leonardis e Skočaj (2011), na ordem de $O(10^1)$.

O estágio de predição possui uma etapa pseudo-linear para a estimativa de tamanhos de descrição, com complexidade $O(n \cdot |K|)$. Além desta etapa, o algoritmo realiza multiplicações vetor-matriz para calcular as distâncias aos protótipos de classes. Este cálculo adiciona a complexidade de $O(n^{2\cdot(1-\delta_{0\tau})})$, que se sobressai em relação à primeira quando τ não é zero. Isto é, o método proposto possui complexidade quadrática de predição quando $\tau \neq 0$ e pseudo-linear quando $\hat{S} = 1$.

3.6 Considerações Finais

Neste capítulo, foi introduzido o GMDL. O método proposto é multiclasse, incremental e pode ser usado em dados com atributos categóricos e numéricos. Dado sua natureza baseada no princípio do MDL, o GMDL provê uma troca benéfica entre a complexidade do modelo e o ajuste aos dados.

O GMDL seleciona uma classe para uma determinada amostra computando descritores para cada um dos atributos da base, que são então utilizados para obter um valor de descrição. Tais descritores são baseados em um método incremental de estimativa de densidade de probabilidade chamado oKDE. Considerando-se apenas estes componentes, o GMDL é muito similar a uma estimativa de verossimilhança máxima normalizada (GRÜNWALD, 2005).

O GMDL também permite que sejam utilizados fatores de penalidade na construção do tamanho da descrição. Um descritor multivariado é estimado para cada classe e é considerado seu protótipo, o qual pode ser utilizado para medir o quão longe uma amostra avaliada está de uma distribuição de uma classe. Além disso, é possível utilizar pesos nos atributos que são ajustados incrementalmente com gradiente descendente estocástico, permitindo que atributos de pouca relevância sejam automaticamente desconsiderados.

4 Avaliação Experimental

Neste capítulo, são apresentados os detalhes dos experimentos conduzidos para a avaliação do método proposto, os resultados obtidos e a discussão dos mesmos. Para avaliação do método, foram considerados dois grandes cenários: *offline*, onde os métodos treinam com uma porcentagem das amostras e classificam a parte restante; e o cenário incremental, onde amostras são apresentadas uma a uma e correções podem ou não serem apresentadas para o método. Este capítulo, está estruturado da seguinte maneira: a Seção 4.1 apresenta as bases de dados utilizadas nos dois grandes cenários de avaliação. A Seção 4.2 apresenta as medidas de desempenho utilizadas para avaliar os métodos. Finalmente, as Seções 4.3 e 4.4 apresentam os resultados dos experimentos nos cenários *offline* e incremental, respectivamente.

4.1 Bases de Dados e Metodologia de Avaliação

Foram empregadas dezesseis bases de dados reais, disponíveis publicamente no site da UCI (LICHMAN, 2013)[1]. A Tabela 1 apresenta a quantidade de amostras em cada classe, bem como a distribuição de amostras por classe. O conjunto de bases selecionadas possui características distintas que visam avaliar a robustez do método proposto mediante diferentes aspectos, tais como: quantidade e tipos de atributos (categórico, discreto, contínuo), balanceamento, quantidade de classes e quantidade de amostras.

As bases 1, 2, 4, 5, 7, 8, 9, 10 e 13 foram utilizadas no extensivo trabalho de Fernández-Delgado et al. (2014). Algumas dessas bases também foram utilizadas anteriormente por Kristan e Leonardis (2014), além das bases 3, 11, 14, 15 e 16. As bases 6 e 12 foram selecionadas pelo seu alto número de amostras, o que as fazem boas representantes de cenários reais onde há um número massivo de dados.

Neste trabalho, as bases foram pré-processadas tendo todos os atributos categóricos codificados com *one-hot encoding* e o atributo classe movido para a última coluna.

4.2 Medidas de Desempenho

A matriz de confusão, originada a partir dos resultados de predição de um classificador em um determinado problema, é uma fonte de dados para diversas métricas utilizadas para mensurar a qualidade de um classificador. Uma matriz de confusão C é uma matriz de tamanho $|K| \cdot |K|$, onde cada linha e cada coluna representa um rótulo de K, e cada

[1] Base de Dados da UCI. Disponível em <https://archive.ics.uci.edu/ml/datasets.html>, acessado em 25 de Novembro de 2017.

Tabela 1 – Informações das bases de dados utilizadas nos experimentos.

		Tamanho		**Composição das classes**	
#	Base de dados	m	n	$\|K\|$	Amostras por classe
1	adult	32.561	109	2	7.841; 24.720
2	contrac	1.473	21	3	333; 511; 629
3	covertype	581.012	10	7	2.747; 9.493; 17.367; 20.510; 35.754; 211.840; 283.301
4	fertility	100	40	2	12; 88
5	hill-valley	1.212	100	2	600; 612
6	ht-sensor	928.991	11	3	276.967; 305.444; 346.580
7	iris	150	4	3	50; 50; 50
8	letter	20.000	16	26	734; 734; 736; 739; 747; 748; 752; 753; 755; 758; 761; 764; 766; 768; 773; 775; 783; 783; 786; 787; 789; 792; 796; 803; 805; 813
9	libras	360	90	15	24; 24; 24; 24; 24; 24; 24; 24; 24; 24; 24; 24; 24; 24; 24
10	miniboone	130.064	50	2	36.499; 93.565
11	skin	245.057	3	2	50.859; 194.198
12	susy	5.000.000	18	2	2.287.827; 2.712.173
13	wdbc	569	30	2	212; 357
14	wine	178	13	3	48; 59; 71
15	wine-red	1.599	11	6	10; 18; 53; 199; 638; 681
16	wine-white	4.898	11	7	5; 20; 163; 175; 880; 1.457; 2.198

item $C_{c_1 c_2}$ representa o número de amostras classificadas como c_1, sendo que seu rótulo real é c_2.

Idealmente, procura-se por uma matriz de confusão onde apenas a diagonal esteja preenchida, isto é, o classificador não cometeu nenhum erro. Como este geralmente não é caso, é útil avaliar o desempenho do classificador com base nos rótulos atribuídos para cada amostra. Os *verdadeiros positivos* (*vp*) são as amostras pertencentes à classe c_1 que foram corretamente classificadas como pertencentes à classe c_1; os *falsos positivos* (*fp*) são as amostras não pertencentes à classe c_1, que foram atribuídas à classe c_1; e os *falsos negativos* (*fn*) são as amostras pertencentes à classe c_1, que não foram atribuídas à classe c_1. Com tais medidas, é possível calcular diversas métricas de eficiência de um classificador, descritas a seguir.

1. *Sensitividade ou recall*: proporção de amostras da classe analisada, identificada corretamente. Pode ser calculada pela seguinte equação:

$$s_i := \frac{vp_i}{vp_i + fn_i}$$

2. *Precisão*: proporção de amostras da classe analisada, que realmente pertencem àquela

classe. Pode ser calculada pela seguinte equação:

$$p_i := \frac{vp_i}{vp_i + fp_i}$$

3. *F-medida*: média harmônica da precisão e sensitividade. Varia de 0 a 1, onde valores mais próximos de 1 demonstram melhor desempenho. Pode ser calculada pela seguinte equação:

$$F := 2 \times \frac{p_i \times s_i}{p_i + s_i}$$

Para problemas multiclasse, é comum se utilizar medidas que consideram a média dos resultados obtidos para cada uma das classes avaliadas. Neste trabalho, foi empregada a média macro, onde o resultado é a obtido pela média dos resultados para cada classe do problema, considerando que todas as classes tem a mesma importância (SOKOLOVA; LAPALME, 2009). As equações para tais médias são apresentadas a seguir.

1. *Macro sensitividade*: média entre todas as sensitividades dentre todas as classes do problema. Pode ser calculada pela seguinte equação:

$$macro_s := \frac{1}{|K|} \times \sum_{i=1}^{|K|} s_i$$

2. *Macro precisão*: média entre todas as precisões dentre todas as classes do problema. Pode ser calculada pela seguinte equação:

$$macro_p := \frac{1}{|K|} \times \sum_{i=1}^{|K|} p_i$$

3. *Macro F-medida*: média harmônica da macro precisão e macro sensitividade. Pode ser calculada pela seguinte equação:

$$macro - F := 2 \times \frac{macro_p \times macro_s}{macro_p + macro_s}$$

Para garantir que os resultados não foram obtidos por acaso, este trabalho utiliza o teste de Friedman para comparar os métodos avaliados, seguindo a metodologia apresentada em Zar (2009, Seção 12.7). Esse teste avalia se a hipótese nula – que neste caso afirma não haver diferenças entre os resultados obtidos – pode ser rejeitada. Isto se dá com base no ranqueamento computado através do desempenho obtido por cada método de classificação para cada base de dados. A Equação 4.1 mostra como computar a medida χ_F^2, utilizada para avaliar a hipótese nula.

$$\chi_F^2 := \left[\frac{12}{DM(M+1)} \sum_{i=1}^{M} R_i^2 \right] - 3D(M+1), \tag{4.1}$$

onde M é a quantidade de métodos avaliados, D é o número de bases de dados e R a soma dos *rankings* de cada método. Caso χ_F^2 seja menor que o valor crítico para um dado grau de confiança, então a hipótese nula deve ser rejeitada. Este valor crítico é obtido pelo número de bases, número de métodos avaliados e um fator de confiança α.

Os métodos de classificação foram ranqueados de acordo com a macro F-medida. O método com melhor desempenho recebe a posição 1 de *ranking*, enquanto o método com pior desempenho recebe a posição M.

Caso a hipótese nula seja rejeitada, é necessário avaliar os métodos par-a-par. De acordo com Benavoli, Corani e Mangili (2016), o teste de Wilcoxon deve ser empregado para evitar problemas conhecidos com métodos baseados em média de *ranks*. Similarmente ao método de Friedman, um valor T é calculado com base na distribuição do *ranking* dos dois métodos avaliados. A hipótese é rejeitada da mesma forma que no método de Friedman, comparando T com um valor crítico baseado num fator de confiança α e o número de bases.

Para calcular T, é computada a diferença d_i entre as medidas de desempenho dos dois métodos avaliados em cada base i. As diferenças são então ranqueadas de acordo com seus valores absolutos, utilizando médias quando há empate (DEMŠAR, 2006). Então, define-se R^+ e R^- como:

$$R^+ := \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i), \qquad R^- := \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i).$$

Isto é, R^+ é a soma dos *ranks* onde o segundo algoritmo foi melhor que o primeiro; e R^- é seu complemento. T é, então, definido como o menor valor entre R^+ e R^-.

4.3 Cenário *Offline*

Este capítulo descreve os experimentos conduzidos para a avaliação do GMDL no cenário *offline*. Este cenário é um caso clássico, onde amostras rotuladas são apresentadas para o método de classificação em batelada e então, amostras não rotuladas são apresentadas para a avaliação do modelo gerado. Esta seção está estrutarada da seguinte maneira: a Seção 4.3.1 apresenta os métodos de classificação avaliados, a Seção 4.3.2 apresenta detalhes sobre a realização do experimento e a Seção 4.3.3 apresenta os resultados obtidos.

4.3.1 Métodos

Para avaliar a eficiência do método proposto, o GMDL foi comparado com métodos clássicos na literatura para aprendizado *offline*. Estes métodos representam a hipótese de maneiras diferentes, tais como otimização, árvores e probabilidade. Além disso, são considerados como base para experimentos do tipo. Eles são:

1. Naïve Bayes gaussiano (GNB) (DUDA; HART, 1973);
2. Florestas aleatórias (RF) (BREIMAN, 1996);
3. Máquinas de vetores de suporte (SVM) (CORTES; VAPNIK, 1995);
4. k-vizinhos mais próximos (kNN) (SALTON; MCGILL, 1986).

4.3.2 Avaliação

Para validação do método, foi utilizada a técnica de validação cruzada aninhada com 5-*fold*. Esta técnica primeiramente separa a base em 5 *fold*. Depois, é iniciado um processo composto por 5 rodadas, onde cada *fold* é usado exatamente uma vez como conjunto de teste e a cada rodada, os 4 *fold* que não foram selecionados para formarem o conjunto de teste, são usados para formar o conjunto de treinamento; esse conjunto também é dividido em 5 *fold*, sendo que um deles é usado para validação de metaparâmetros em uma busca em grade. Depois deste processo, uma métrica final é computada com base na média dos resultados obtidos em cada *fold*. Isto permite que cada amostra esteja exatamente uma vez na partição de teste, fazendo com que eventuais tendências da base não afetem demasiadamente o desempenho do modelo gerado. Além disso, todas as amostras foram ordenadas aleatoriamente, utilizando a semente `1234567890`.

A seleção dos melhores parâmetros foi feita considerando a macro F-medida. Os atributos utilizados na busca em grade são mostrados a seguir. Os métodos implementados na biblioteca `Scikit-learn`, são apresentados com os nomes dos parâmetros utilizados em suas interfaces.

1. **GMDL**
   a) $\tilde{\sigma}^2$: 2; 5; 10;
   b) τ: 0; 5; 10.
2. **RF**
   a) `n_estimators`: 10; 20; 30; 40; 50; 60; 70; 80; 90; 100;
   b) `criterion`: "*gini*"; "*entropy*".
3. **SVM**
   a) `C`: 0,0001; 0,001; 0,01; 0,1; 1; 10; 100; 1000.
4. **kNN**
   a) `n_neighbors`: 3; 5; 7; 9; 11; 13; 15; 17; 19.

Os parâmetros α, f e η do GMDL foram mantidos como 0,001, 1 e 0,9, respectivamente. Os valores do gradiente foram mantidos como valores padrões encontrados na literatura (SRIVASTAVA et al., 2014), onde α foi reduzido ainda mais em detrimento do tamanho das bases utilizadas. O fator de esquecimento f foi mantido como 1, uma vez que as bases públicas utilizadas não contém fator temporal envolvido, como tendência e ruído temporal. Além disso, o SVM utilizado nos experimentos foi o linear, uma vez que o SVM com *kernel* Gaussiano é computacionalmente inviável de ser utilizado mesmo em bases relativamente pequenas (KRISTAN; LEONARDIS, 2014). Todos os outros parâmetros dos demais métodos foram mantidos com os valores padrões da biblioteca.

O parâmetro τ do GMDL foi utilizado somente nas bases *iris*, *wine*, *fertility*, *wdbc*. Tal fato deve-se à lentidão da computação na busca em grade para bases maiores. Portanto, apesar dos benefícios que podem ser obtidos pela utilização da distância em protótipo, foi-se escolhido a não utilização neste experimento devido ao alto custo computacional.

As bases de dados utilizadas no experimento foram previamente normalizadas, considerando o *fold* de treinamento, para possuírem média zero e desvio padrão um.

Todos os experimentos foram realizados no servidor da Universidade Federal de São Carlos – *campus* Sorocaba, disponibilizado pelo Laboratório de Sistemas Distribuídos e Inteligentes. Este servidor conta com 40 núcleos Intel de sexta geração de 2.3Mhz e 4Mb de cache. Os métodos que não concluíram o trabalho de classificação dentro de 48h, foram terminados e reportados com F-medida zero.

4.3.3 Resultados

A Tabela 2 apresenta os resultados obtidos pelos métodos de classificação para cada base de dados. O desempenho de cada técnica foi avaliado pela macro F-medida obtida na validação cruzada aninhada. Para facilitar a comparação, os resultados são apresentados em um mapa de calor em tons de cinza sendo que quanto mais escura for a célula, melhor o resultado. Além disso, o melhor resultado de cada base é apresentado em negrito.

Os resultados apontam que o método RF obteve o melhor desempenho na maioria dos experimentos. No entanto, os demais métodos também mostram bom desempenho em algumas bases. Além disso, é possível notar o problema que bases com muitas amostras (como a *susy*) causa em alguns dos métodos, uma vez que RF, SVM e kNN não terminaram a computação em 48 horas. Isto deve-se principalmente pela maneira que estes métodos funcionam, seja otimizando um grande conjunto de dados, como RF e SVM, seja analisando-os extensivamente, como o kNN. O GMDL e o GNB mostram-se mais robustos para problemas em uma escala deste tamanho. Em especial, é possível notar a influência do Teorema 1.2 no GMDL: um maior número de amostras permite uma melhor estimativa de densidade das distribuições.

Tabela 2 – F-medidas obtidas para cada método avaliado no cenário *offline*.

	GMDL	GNB	RF	SVM	kNN
adult	**1,000**	**1,000**	**1,000**	**1,000**	0,916
contrac	0,477	0,468	**0,489**	0,484	0,445
covertype	0,411	0,401	**0,629**	0,273	0,492
fertility	0,567	0,656	**0,688**	0,599	0,468
hill-valley	0,480	0,521	0,582	**0,726**	0,532
ht-sensor	**0,549**	0,537	0,480	0,499	0,458
iris	0,948	0,953	0,940	0,954	**0,967**
letter	0,703	0,648	**0,962**	0,694	0,936
libras	0,586	0,603	**0,741**	0,623	0,628
miniboone	0,847	0,556	**0,920**	0,885	0,871
skin	0,917	0,880	**0,999**	0,887	**0,999**
susy	**0,745**	0,737	0,000	0,000	0,000
wdbc	0,938	0,928	0,960	0,962	**0,970**
wine	0,979	0,957	0,962	**0,984**	0,958
wine-red	0,299	**0,305**	0,274	0,251	0,268
wine-white	0,267	0,279	**0,281**	0,238	0,242

O GMDL mostrou-se pouco efetivo nas bases *hill-valley* e *libras*. Ambas são bases de alta cardinalidade de atributos, onde os erros das estimativas do oKDE podem se agregar quando estimando o tamanho de descrição de classe. Além disso, como apontado por Kristan e Leonardis (2014), a computação do protótipo de classe como uma distribuição multivariada sofre uma grande suavização pelo alto número de variáveis envolvidas.

É possível notar que todos os métodos tiveram ótimo desempenho na base *adult*, que é uma base desbalanceada. Tal resultado pode ser explicado pela alta capacidade preditiva dos atributos desta base, que pode ser notado pelo alto desempenho do kNN, que depende diretamente da similaridade entre amostras.

A fim de obter-se uma análise mais confiável dos resultados, foi realizado o teste estatístico não-paramétrico de Friedman considerando o ranqueamento dos métodos. O ranqueamento é mostrado pela Tabela 3. Para um intervalo de confiança $\alpha = 0{,}05$, com 16 bases de dados e 5 métodos, o valor crítico é 0,711. Os *ranks* médios neste intervalo crítico são mostrados pela Figura 9. Uma vez que $\chi_F^2 = 6$, a hipótese nula foi rejeitada, e conclui-se que há uma diferença significativa nos resultados obtidos. Deste modo, foi conduzido um teste *post-hoc* com o método Wilcoxon para cada um dos métodos, par-a-par, comparando com o GMDL. Nenhum deles teve a hipótese rejeitada, o que indica que não há evidência estatística para dizer que são, par-a-par, superiores ao GMDL.

Apesar do GMDL não apresentar uma vantagem estatística em comparação aos demais métodos avaliados neste experimento, considerando a natureza de tais métodos, pode-se afirmar que o GMDL possui uma vantagem sobre eles sendo naturalmente incremental, o que o torna flexível a outros cenários, além de escalável.

Tabela 3 – Ranqueamento total obtido para cada método avaliado no cenário *offline*.

	GMDL	GNB	RF	SVM	kNN
adult	**2,5**	**2,5**	**2,5**	**2,5**	5,0
contrac	3,0	4,0	**1,0**	2,0	5,0
covertype	3,0	4,0	**1,0**	5,0	2,0
fertility	4,0	2,0	**1,0**	3,0	5,0
hill-valley	5,0	4,0	2,0	**1,0**	3,0
ht-sensor	**1,0**	2,0	4,0	3,0	5,0
iris	4,0	3,0	5,0	2,0	**1,0**
letter	3,0	5,0	**1,0**	4,0	2,0
libras	5,0	4,0	**1,0**	3,0	2,0
miniboone	4,0	5,0	**1,0**	2,0	3,0
skin	3,0	5,0	**1,5**	4,0	**1,5**
susy	**1,0**	2,0	4,0	4,0	4,0
wdbc	4,0	5,0	3,0	2,0	**1,0**
wine	2,0	5,0	3,0	**1,0**	4,0
wine-red	2,0	**1,0**	3,0	5,0	4,0
wine-white	3,0	2,0	**1,0**	5,0	4,0
Soma	49,5	55,5	35,0	48,5	51,5

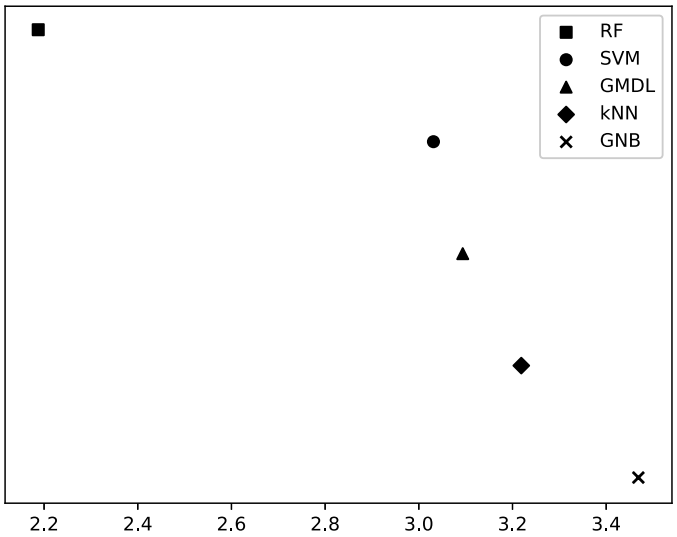


Figura 9 – Ranqueamento médio de cada método avaliado no cenário *offline*.

4.4 Cenário Incremental

Este cenário é mais próximo à utilização de métodos de classificação em aplicações reais. Os métodos são treinados com poucas amostras rolutadas, e então recebem tanto amostras não rotuladas para classificação, quanto amostras de correção para que possam ajustar o modelo ao decorrer do tempo. Existem diversas situações que podem ser simuladas neste cenário, variando quantidade de amostras corrigidas, tempo para correção, dentre

outras. Esta seção está estrutarada da seguinte maneira: a Seção 4.3.1 apresenta os métodos de classificação avaliados, a Seção 4.3.2 apresenta detalhes sobre a realização dos experimentos e a Seção 4.3.3 apresenta os resultados obtidos.

4.4.1 Métodos

Para avaliar a eficiência do método proposto, o GMDL foi comparado com métodos tradicionais disponíveis na literatura para treinamento incremental, que não necessitam da discretização dos atributos. Além disso, são considerados como base para experimentos do tipo. São eles:

1. Perceptron multinível (MLP) (ROSENBLATT, 1958);
2. Passivo-Agressivo (PA) (CRAMMER et al., 2006);
3. Gradiente Descendente Estocástico (SGD) (BOTTOU, 2010).

4.4.2 Avaliação

Para validação do método, 10% da base foi disponibilizada ao método para treinamento, assim como realizado por Kristan e Leonardis (2014). O restante das amostras são, então, apresentadas uma a uma para classificação, havendo retroalimentação para aprendizado incremental. Como explorado anteriormente, é possível simular ambientes mais próximos aos reais alterando fatores como o atraso e a quantidade de amostras apresentadas. Foram simulados três cenários diferentes:

1. *Correção imediata*: quando uma amostra é classificada com o rótulo errado, o método recebe a amostra para treinamento com seu rótulo correto imediatamente;
2. *Correção limitada*: quando uma amostra é classificada com o rótulo errado, o método recebe a amostra para treinamento com seu rótulo correto com uma probabilidade de 50%. Para isto, um gerador de números aleatórios foi utilizado com a semente `1234567890`;
3. *Correção atrasada*: quando uma amostra é classificada com o rótulo errado, o método recebe a amostra para treinamento com seu rótulo correto em um tempo que cresce exponencialmente em 1,1, arredondado para baixo. Isto é, as amostras são apresentadas com atraso de $\{\lfloor 1{,}1 \rfloor^0, \lfloor 1{,}1 \rfloor^1, \lfloor 1{,}1 \rfloor^2, \ldots\}$, fazendo com que quanto mais um método erre, mais demore para que ele receba uma correção.

Cada um destes cenários foi testado tanto com dados normalizados quanto não normalizados. Os dados foram normalizados de forma incremental, utilizando a base de

treinamento como referência, para terem média zero e desvio padrão um. Foram, então, consideradas as melhores F-medidas dentre os cenários com e sem normalização para reportar os resultados.

Todos os parâmetros dos métodos implementados na biblioteca `Scikit-learn` foram mantidos com seus valores padrões. Os parâmetros utilizados no SGD com regularização L2 e função de custo *hinge*, tornam o método similar a um SVM linear com aprendizado incremental (BOTTOU, 2010). O GMDL, por sua vez, utilizou os seguintes parâmetros:

1. $\tilde{\sigma}^2$: 2;
2. τ: 0;
3. α: 0,001;
4. f: 1;
5. η: 0,9.

Os valores de α, f e η seguiram os mesmos preceitos expostos na Seção 4.3.2 para serem determinados. O valor de τ foi definido como zero para permitir uma computação mais rápida. E finalmente, foi-se escolhido o menor valor de $\tilde{\sigma}^2$ dos experimentos *offline*, a fim de causar menos perturbações nas amostras.

Todos os experimentos foram realizados no servidor da Universidade Federal de São Carlos – *campus* Sorocaba, disponibilizado pelo Laboratório de Sistemas Distribuídos e Inteligentes. Este servidor conta com 40 núcleos Intel de sexta geração de 2.3Mhz e 4Mb de cache.

4.4.3 Resultados

Esta seção apresenta os resultados dos experimentos de aprendizado incremental considerando diferentes cenários de correção.

4.4.3.1 Correção Imediata

A Tabela 4 apresenta os resultados obtidos pelos métodos de classificação para cada base de dados no cenário onde correções são efetuadas no mesmo instante que uma classificação incorreta é realizada. O desempenho de cada técnica foi avaliado pela macro F-medida obtida nas amostras de teste. Para facilitar a comparação, os resultados são apresentados em um mapa de calor em tons de cinza sendo que quanto mais escura for a célula, melhor o resultado. Além disso, o melhor resultado de cada base é apresentado em negrito.

Tabela 4 – F-medidas obtidas para cada método avaliado no cenário de correção imediata.

	GMDL	MLP	PA	SGD
adult	**1,000**	0,989	**1,000**	0,999
contrac	**0,465**	0,449	0,393	0,410
covertype	0,400	**0,526**	0,294	0,377
fertility	0,537	0,477	0,626	**0,650**
hill-valley	0,492	0,487	**0,639**	0,528
ht-sensor	0,587	**0,943**	0,463	0,482
iris	**0,947**	0,590	0,795	0,843
letter	0,670	**0,726**	0,446	0,569
libras	**0,514**	0,471	0,373	0,291
miniboone	0,817	**0,833**	0,756	0,765
skin	0,835	**0,996**	0,811	0,802
susy	0,687	0,667	0,677	**0,703**
wdbc	0,924	0,906	0,938	**0,944**
wine	**0,969**	0,755	0,947	0,934
wine-red	**0,299**	0,251	0,238	0,251
wine-white	**0,243**	0,232	0,199	0,200

Os resultados apontam que o GMDL obteve o melhor resultado na maioria das bases de dados, não tendo a menor F-medida em nenhuma delas. No entanto, é possível notar que o MLP também mostrou bom desempenho em algumas bases como *miniboone* e *letter*. Essas bases tem um número consideravelmente grande de amostras, o que permite que a otimização dos parâmetros da rede, convirjam de modo mais preciso. Tal fato pode ser observado também pelo desempenho do SGD na base *susy*, onde o método alcançou a melhor F-medida.

Neste experimento, o GMDL apresentou resultados consistentes com o cenário *offline*, tendo desempenho tão bom quanto ou melhor em bases como *ht-sensor*, *hill-valley* e *adult*. Na base *adult*, o GMDL alcançou uma predição perfeita no cenário incremental, assim como o PA. Tal resultado para o PA explica-se numa base onde atributos são altamente preditivos, uma vez que erros na classificação fazem o modelo se adaptar rapidamente às demais amostras, que compartilham de características semelhantes.

Para as demais bases, o GMDL foi consistentemente pior que no cenário *offline*. Isso pode ser influenciado pelo ajuste dos parâmetros dado a normalização incremental que as amostras sofreram. É possível notar também, que o MLP e o SGD tiveram desempenho mais baixo em várias bases, especialmente as bases com menos amostras e desbalanceadas. Esta deficiência ilustra os problemas de tais métodos para otimização em cenários reais.

A fim de obter-se uma análise mais confiável dos resultados, foi realizado o teste estatístico não-paramétrico de Friedman considerando o ranqueamento dos métodos. O ranqueamento é mostrado pela Tabela 5. Para um intervalo de confiança $\alpha = 0{,}05$, com 16 bases de dados e 4 métodos, o valor crítico é 0,352. Os *ranks* médios neste intervalo crítico

Tabela 5 – Ranqueamento total obtido para cada método avaliado no cenário de correção imediata.

	GMDL	MLP	PA	SGD
adult	**1,5**	4,0	**1,5**	3,0
contrac	**1,0**	2,0	4,0	3,0
covertype	2,0	**1,0**	4,0	3,0
fertility	3,0	4,0	2,0	**1,0**
hill-valley	3,0	4,0	**1,0**	2,0
ht-sensor	2,0	**1,0**	4,0	3,0
iris	**1,0**	4,0	3,0	2,0
letter	2,0	**1,0**	4,0	3,0
libras	**1,0**	2,0	3,0	4,0
miniboone	2,0	**1,0**	4,0	3,0
skin	2,0	**1,0**	3,0	4,0
susy	2,0	4,0	3,0	**1,0**
wdbc	3,0	4,0	2,0	**1,0**
wine	**1,0**	4,0	2,0	3,0
wine-red	**1,0**	2,5	4,0	2,5
wine-white	**1,0**	2,0	4,0	3,0
Soma	28,5	41,5	48,5	41,5

são mostrados pela Figura 10. Uma vez que $\chi^2_F = 7{,}838$, a hipótese nula foi rejeitada, e conclui-se que há uma diferença significativa nos resultados obtidos.

Deste modo, foi conduzido um teste *post-hoc* com o método Wilcoxon para cada um dos métodos, par-a-par, comparando com o GMDL. Para um conjunto de 16 bases, considerando um intervalo de confiança $\alpha = 0{,}05$, o valor crítico é 30. O valor dos coeficientes T para o MLP, SGD e PA, comparados com o GMDL foram de 54, 27 e 24, respectivamente. Portanto, pode-se afirmar que o GMDL foi superior ao SGD e ao PA, em uma análise par-a-par. No entanto, não há evidência estatística o suficiente para afirmar que o GMDL foi superior ao MLP.

4.4.3.2 Correção Limitada

A Tabela 6 apresenta os resultados obtidos pelos métodos de classificação para cada base de dados no cenário onde amostras incorretas são corrigidas imediatamente com uma probabilidade de 50%. O desempenho de cada técnica foi avaliado pela macro F-medida obtida nas amostras de teste. Para facilitar a comparação, os resultados são apresentados em um mapa de calor em tons de cinza sendo que quanto mais escura for a célula, melhor o resultado. Além disso, o melhor resultado de cada base é apresentado em negrito.

Os resultados apontam que o GMDL obteve o melhor resultado em diversas bases de dados, obtendo a menor F-medida somente na base *skin*, onde empatou com o SGD e

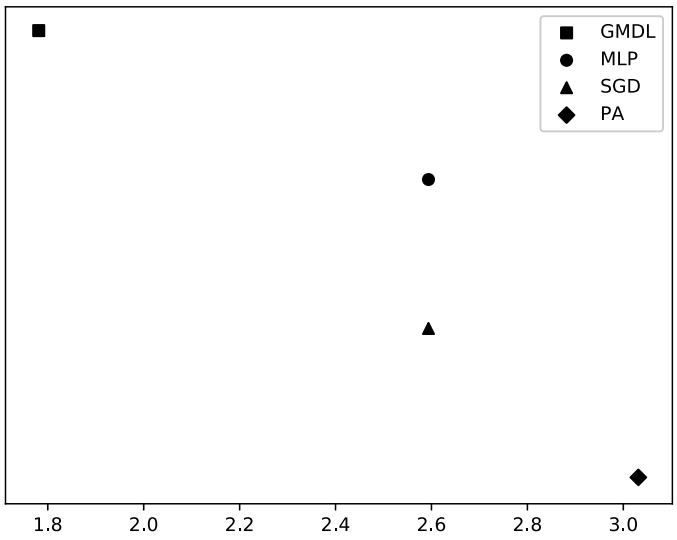


Figura 10 – Ranqueamento médio de cada método avaliado no cenário de correção imediata.

Tabela 6 – F-medidas obtidas para cada método avaliado no cenário de correção limitada.

	GMDL	MLP	PA	SGD
adult	**1,000**	0,989	**1,000**	0,999
contrac	**0,453**	0,383	0,375	0,396
covertype	0,413	**0,521**	0,283	0,315
fertility	0,492	0,477	**0,607**	0,518
hill-valley	0,497	0,493	**0,627**	0,525
ht-sensor	0,563	**0,943**	0,430	0,459
iris	**0,939**	0,566	0,789	0,747
letter	0,670	**0,710**	0,402	0,560
libras	**0,392**	**0,392**	0,328	0,258
miniboone	0,770	**0,831**	0,749	0,762
skin	0,800	**0,995**	0,801	0,800
susy	0,690	0,672	0,668	**0,703**
wdbc	0,924	0,906	0,929	**0,944**
wine	0,932	0,755	**0,947**	0,934
wine-red	**0,306**	0,252	0,231	0,240
wine-white	**0,235**	0,208	0,198	0,226

foi muito próximo ao desempenho do PA. O MLP teve um resultado muito superior aos demais métodos nesta base, e apenas ligeiramente pior que seu desempenho no cenário de correção imediata. Isto demonstra que o MLP possui um desempenho muito bom em bases com menor número de atributos.

Ambos o GMDL e o PA atingiram predição perfeita na base *adult*. Isso mostra que eles conseguiram representar de forma efetiva as nuâncias de uma base com amostras

semelhantes. De um ponto de vista matemático, isto demonstra que o GMDL conseguiu estimar a distribuição dos atributos reais de forma fidedigna, mesmo no cenário onde nem sempre há correção para consolidar ainda mais o modelo.

Tabela 7 – Ranqueamento total obtido para cada método avaliado no cenário de correção limitada.

	GMDL	MLP	PA	SGD
adult	**1,5**	4,0	**1,5**	3,0
contrac	**1,0**	3,0	4,0	2,0
covertype	2,0	**1,0**	4,0	3,0
fertility	3,0	4,0	**1,0**	2,0
hill-valley	3,0	4,0	**1,0**	2,0
ht-sensor	2,0	**1,0**	4,0	3,0
iris	**1,0**	4,0	2,0	3,0
letter	2,0	**1,0**	4,0	3,0
libras	**1,5**	**1,5**	3,0	4,0
miniboone	2,0	**1,0**	4,0	3,0
skin	3,5	**1,0**	2,0	3,5
susy	2,0	3,0	4,0	**1,0**
wdbc	3,0	4,0	2,0	**1,0**
wine	3,0	4,0	**1,0**	2,0
wine-red	**1,0**	2,0	4,0	3,0
wine-white	**1,0**	3,0	4,0	2,0
Soma	32,5	41,5	45,5	40,5

A fim de obter-se uma análise mais confiável dos resultados, foi realizado o teste estatístico não-paramétrico de Friedman considerando o ranqueamento dos métodos. O ranqueamento é mostrado pela Tabela 7. Para um intervalo de confiança $\alpha = 0{,}05$, com 16 bases de dados e 4 métodos, o valor crítico é 0,352. Os *ranks* médios neste intervalo crítico são mostrados pela Figura 11. Uma vez que $\chi_F^2 = 3{,}338$, a hipótese nula foi rejeitada, e conclui-se que há uma diferença significativa nos resultados obtidos.

Deste modo, foi conduzido um teste *post-hoc* com o método Wilcoxon para cada um dos métodos, par-a-par, comparando com o GMDL. Para um conjunto de 16 bases, considerando um intervalo de confiança $\alpha = 0{,}05$, o valor crítico é 30. O valor dos coeficientes T para o MLP, SGD e PA, comparados com o GMDL foram de 55, 28 e 27,5, respectivamente. Portanto, pode-se afirmar que o desempenho do GMDL foi superior ao SGD e ao PA. No entanto, não há informação estatística o suficiente para afirmar que o GMDL obteve desempenho melhor que o MLP.

4.4.3.3 Correção Atrasada

A Tabela 8 apresenta os resultados obtidos pelos métodos de classificação para cada base de dados no cenário onde a correção torna-se mais atrasada quanto mais o

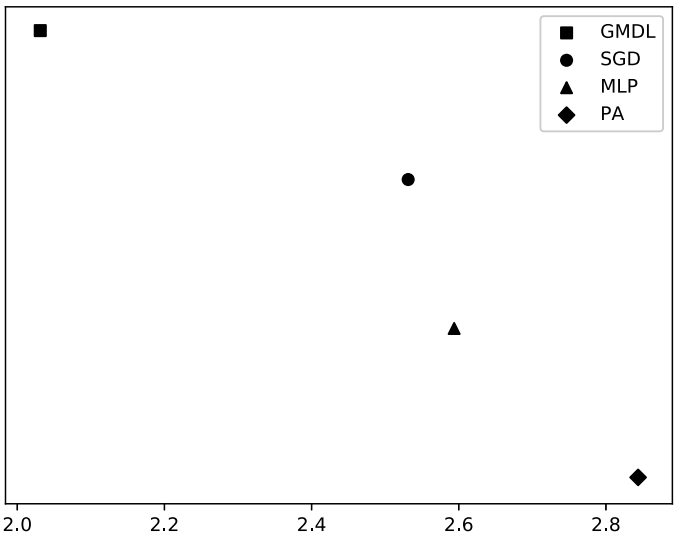


Figura 11 – Ranqueamento médio de cada método avaliado no cenário de correção limitada.

classificador comete erros. O desempenho de cada técnica foi avaliado pela macro F-medida obtida nas amostras de teste. Para facilitar a comparação, os resultados são apresentados em um mapa de calor em tons de cinza sendo que quanto mais escura for a célula, melhor o resultado. Além disso, o melhor resultado de cada base é apresentado em negrito.

Tabela 8 – F-medidas obtidas para cada método avaliado no cenário de correção atrasada.

	GMDL	MLP	PA	SGD
adult	**1,000**	0,907	**1,000**	0,999
contrac	**0,444**	0,403	0,402	0,432
covertype	0,307	0,382	0,288	**0,401**
fertility	0,482	0,480	**0,568**	0,565
hill-valley	0,520	0,525	**0,541**	0,507
ht-sensor	0,347	**0,714**	0,522	0,549
iris	**0,947**	0,468	0,767	0,740
letter	0,308	0,074	0,493	**0,537**
libras	0,207	0,159	**0,310**	0,271
miniboone	0,558	**0,810**	0,756	0,779
skin	**0,917**	0,911	0,684	0,456
susy	0,672	0,734	0,656	**0,782**
wdbc	**0,940**	0,908	0,930	0,936
wine	0,906	0,713	**0,947**	0,928
wine-red	**0,281**	0,221	0,235	0,236
wine-white	0,232	0,174	0,145	**0,251**

Os resultados mostram que todos os métodos avaliados tiveram uma distribuição quase uniforme de melhor desempenho pelas bases. Como neste cenário erros constantes diminuem a chance do método aprender, espera-se que bases onde os métodos tiveram

um desempenho pior anteriormente, tenham um desempenho pior ainda neste teste, e vice-versa.

Pode-se notar que o GMDL e o PA obtiveram predição perfeita na base *adult*, o que demonstra que os modelos gerados para representação dos dados foi obtido com perfeição já no começo do treinamento, uma vez que nenhum erro foi cometido.

A maioria dos piores desempenhos foram obtidos pelo MLP. É possível que a falta de correção impediu que o peso dos neurônios fossem atualizados eficientemente. Esta é uma deficiência conhecida das redes neurais. É possível notar, em contraste, que o GMDL, apesar do impacto, não sofreu do mesmo mal cosistentemente. Na base *hill-valley*, o GMDL obteve uma F-medida melhor que no teste *offline*.

Para mostrar que o GMDL é pouco sensível a normalização dos dados, a Tabela 9 apresenta as F-medidas obtidas neste cenário quando as amostras não são normalizadas. Quando comparado apenas com entradas não normalizadas, é notória a diferença de desempenho do GMDL sobre os demais métodos: o MLP e o PA sofrem demasiadamente pela falta de normalização e atraso na correção. Este fato demonstra a robustez do GMDL a amostras não normalizadas.

Tabela 9 – F-medidas obtidas para cada método avaliado no cenário de correção atrasada onde as amostras não foram normalizadas.

	GMDL	MLP	PA	SGD
adult	**1,000**	0,492	0,508	0,506
contrac	**0,444**	0,403	0,358	0,358
covertype	**0,307**	0,145	0,194	0,264
fertility	0,435	0,477	**0,568**	0,497
hill-valley	0,481	0,507	**0,541**	0,507
ht-sensor	0,347	0,383	0,429	**0,495**
iris	**0,947**	0,468	0,471	0,677
letter	0,308	0,064	0,359	**0,410**
libras	**0,207**	0,094	0,048	0,038
miniboone	0,558	0,514	0,684	**0,697**
skin	**0,917**	0,658	0,684	0,439
susy	0,672	0,734	0,656	**0,782**
wdbc	**0,940**	0,535	0,470	0,530
wine	**0,906**	0,354	0,321	0,258
wine-red	**0,281**	0,205	0,150	0,177
wine-white	**0,232**	0,121	0,134	0,146

A fim de obter-se uma análise mais confiável dos resultados, foi realizado o teste estatístico não-paramétrico de Friedman considerando o ranqueamento dos métodos. O ranqueamento é mostrado pela Tabela 10. Para um intervalo de confiança $\alpha = 0{,}05$, com 16 bases de dados e 4 métodos, o valor crítico é 0,352. Os *ranks* médios neste intervalo crítico

são mostrados pela Figura 12. Uma vez que $\chi_F^2 = 3{,}994$, a hipótese nula foi rejeitada, e conclui-se que há uma diferença significativa nos resultados obtidos.

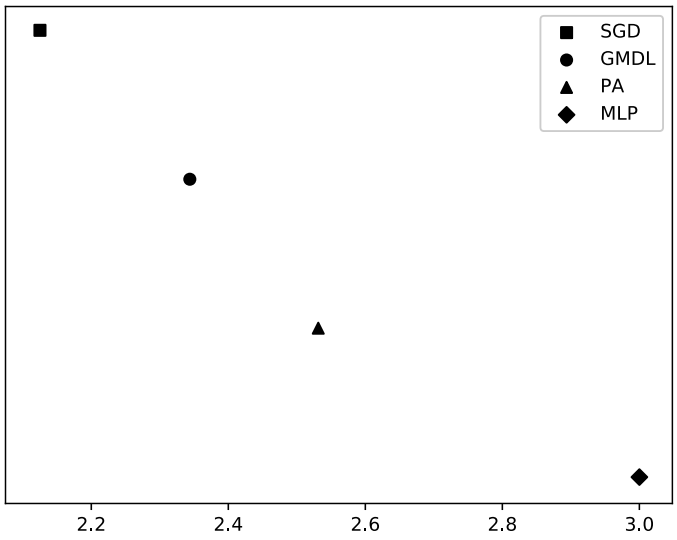


Figura 12 – Ranqueamento médio de cada método avaliado no cenário de correção atrasada.

Deste modo, foi conduzido um teste *post-hoc* com o método Wilcoxon para cada um dos métodos, par-a-par, comparando com o GMDL. Para um conjunto de 16 bases, considerando um intervalo de confiança $\alpha = 0{,}05$, o valor crítico é 30. O valor dos coeficientes T para o MLP, SGD e PA, comparados com o GMDL foram de 50, 46 e 55, respectivamente. Portanto, não existe uma diferença estatística considerável entre o GMDL e os demais métodos no cenário de correção atrasada. No entanto, quando considera-se apenas o cenário onde as amostras não foram normalizadas os valores de T para o MLP, SGD e PA, comparados com o GMDL foram de 13, 37 e 27, respectivamente. Pode-se afirmar, portanto que o GMDL é estatisticamente superior a tanto o MLP quanto ao PA quando considera-se apenas as amostras não normalizadas.

4.5 Considerações Finais

Neste capítulo, foram apresentados os experimentos conduzidos para a comparação do GMDL com outros métodos em diferentes cenários. Tais métodos são amplamente utilizados como base de comparação na literatura.

A análise estatística dos resultados mostrou que o método proposto é equivalente aos métodos GNB, RF, SVM, kNN no cenário *offline*. No entanto, é importante notar que o GMDL é naturalmente incremental, robusto a normalização dos dados, e capaz de processar dados inteiros e também contínuos. Tais características o tornam um método mais flexível que os demais. Além disso, a análise dos métodos no cenário incremental

Tabela 10 – Ranqueamento total obtido para cada método avaliado no cenário de correção atrasada.

	GMDL	MLP	PA	SGD
adult	**1,5**	4,0	**1,5**	3,0
contrac	**1,0**	3,0	4,0	2,0
covertype	3,0	2,0	4,0	**1,0**
fertility	3,0	4,0	**1,0**	2,0
hill-valley	3,0	2,0	**1,0**	4,0
ht-sensor	4,0	**1,0**	3,0	2,0
iris	**1,0**	4,0	2,0	3,0
letter	3,0	4,0	2,0	**1,0**
libras	3,0	4,0	**1,0**	2,0
miniboone	4,0	**1,0**	3,0	2,0
skin	**1,0**	2,0	3,0	4,0
susy	3,0	2,0	4,0	**1,0**
wdbc	**1,0**	4,0	3,0	2,0
wine	3,0	4,0	**1,0**	2,0
wine-red	**1,0**	4,0	3,0	2,0
wine-white	2,0	3,0	4,0	**1,0**
Soma	37,5	48,0	40,5	34,0

mostrou que o GMDL foi superior aos métodos SGD e PA nos cenários de correção imediata e limitada. No cenário de correção atrasada, o GMDL foi equivalente aos demais métodos, mas mostrou-se superior ao MLP e ao PA quando considera-se apenas os resultados onde as amostras não são normalizadas.

Conclusão

Em problemas de classificação é muito comum obter-se vários modelos – ou hipóteses – para um mesmo problema. Cada hipótese descreve os dados como um conjunto de parâmetros a serem utilizados pelo método de classificação utilizados para rotular amostras não vistas. O problema de determinar qual modelo melhor generaliza a base sem causar sobreajuste é um problema muito investigado na área de aprendizado de máquina.

Nesta dissertação, foi apresentado um novo método de classificação multinomial baseado no princípio do MDL, o GMDL. O método proposto é multiclasse, multinomial, naturalmente incremental e robusto ao sobreajuste e normalização dos dados. Tais características tornam-o um bom candidato a aplicações em grande escala e em tempo real.

O GMDL baseia-se no princípio do MDL, que é uma formalização da ideia da Navalha de Occam. Este princípio provê ao método uma troca benéfica entre acurácia e sobrejuste dos dados. O GMDL toma como base a interpretação de codificação do MDL, utilizando-se de estimativas de densidade de distribuições para computar os tamanhos de descrição das amostras. Essas estimativas são realizadas utilizando-se o oKDE como estimador incremental, o que permite não só que amostras de quaisquer tipo sejam estimadas, como também permite que o método seja totalmente incremental e possa melhorar seu desempenho com o passar do tempo.

Para avaliar o método proposto, foi comparado o seu desempenho em dois grandes cenários: *offline* e incremental. No cenário *offline* utilizou-se validação aninhada em 5-*fold* e busca em grade para encontrar-se os melhores parâmetros dos métodos avaliados. O GMDL foi comparado com outros 4 métodos considerados base na literatura, em 16 bases com características distintas. No cenário incremental, considerou-se ainda três cenários distintos no que tange à correção de rótulos errados pelos métodos: correção atrasada, correção imediata e correção limitada. Todos os cenários foram ainda considerados para o caso de amostras serem ou não normalizadas, trazendo características de um ambiente real à simulação. Além disto, utilizou-se de técnicas estatísticas para validar os resultados obtidos.

A análise mostrou que o GMDL é equivalente aos métodos GNB, RF, SVM, kNN no cenário *offline*. Nos cenários incrementais, o GMDL se mostrou superior aos métodos SGD e PA no cenários de correção imediata e limitada. No entanto, o GMDL obteve resultados consistentes com aqueles vistos no teste *offline*, o que corrobora o fato do GMDL ser versátil em comparação aos outros métodos. No cenário de correção atrasada, o GMDL foi equivalente aos demais métodos, mas superior ao MLP e ao PA no cenário

não normalizado. Neste cenário, é possível notar o impacto que a falta de normalização causa nos métodos comparados; o GMDL mostrou-se, apesar disso, resiliente a este mal. Além disso, o GMDL se mostrou particularmente eficiente para bases onde amostras são boas representantes da distribuição subjacente, estimando distribuições que permitiram predição perfeita em uma das bases.

O GMDL mostrou-se robusto e eficiente nos cenários testados, sendo um bom candidato a casos onde há um grande fluxo de dados e/ou a computação com métodos tradicionais pode ser custosa. Além disso, o GMDL apresenta uma interpretação nova sobre o princípio do MDL aplicado à classificação, estendendo o método `MDLText` introduzido por Silva, Almeida e Yamakami (2017), permitindo que não só amostras textuais sejam rotuladas, mas amostras de qualquer tipo. Este fato torna o GMDL um candidato a grandes problemas do mundo real.

Limitações e Trabalhos Futuros

Durante o desenvolvimento deste trabalho, foram verificados pontos de extensão e melhoria da proposta apresentada. A seguir, são apresentados direcionamentos a trabalhos futuros:

- O desempenho positivo do protótipo de classe dá indícios de que o cálculo do tamanho da descrição possa ser puramente dependente de uma distribuição que estime todos os atributos de uma única vez. No entanto, problemas de suavização demasiada pela projeção dos atributos na variedade n-dimensional, deve ser considerada;

- A demora na computação da distância ao protótipo, demonstra que seria possível propor um método de aproximação de distância de ponto a uma distribuição mais eficiente: possivelmente aproximando-se candidatos representantes de uma distribuição, como os protótipos de classse do `MDLText` ou oKDE. Este cálculo pode ser feito a partir de clusterização de pontos aleatórios na distribuição computados a partir de uma distribuição uniforme;

- O tamanho de descrição da classe impactou negativamente no desempenho do GMDL em alguns casos. Uma vertente de pesquisa seria validar novas complexidades de modelo, avaliando o desempenho de diferentes $L(M)$;

- Dado o grande número de hiperparâmetros do GMDL, seria interessante investigar a sensibilidade dos mesmos;

- Outro viés de pesquisa seria utilizar o GMDL em outros problemas, como classificação de textos e de imagem.

Publicações

Durante o período de curso de mestrado, o seguinte trabalho foi produzido em colaboração com outros pesquisadores:

- FREITAS, B. L.; ALMEIDA, T.A.; SILVA, R. M.. "Protótipo de um Método de Classificação por Descrição Mínima". Anais do XIV Encontro Nacional de Inteligência Artificial e Computacional (ENIAC'17), v. 1. p. 1-12, Uberlândia, Brasil. 2017.

Referências

ALMEIDA, T. A.; YAMAKAMI, A. Advances in spam filtering techniques. *Computational Intelligence for Privacy and Security*, Springer, v. 394, n. 2012, p. 199–214, 2012. Citado na página 15.

ALMEIDA, T. A.; YAMAKAMI, A. Facing the spammers: A very effective approach to avoid junk e-mails. *Expert Systems with Applications*, Elsevier, v. 39, n. 7, p. 6557–6561, 2012. ISSN 09574174. Citado na página 15.

ALMEIDA, T. A.; YAMAKAMI, A. Compression-based spam filter. *Security and Communication Networks*, Wiley-Blackwell, v. 9, n. 4, p. 327–335, 2016. Citado na página 15.

ALMEIDA, T. A. D. *SPAM: do Surgimento à Extinção*. 124 p. Tese (Ph.D. Thesis) — State University of Campinas, 2010. Citado 3 vezes nas páginas 2, 15 e 16.

BARRON, A.; COVER, T. Minimum complexity density estimation. *IEEE Transactions on Information Theory*, IEEE, v. 37, n. 4, p. 1034–1054, jul 1991. ISSN 00189448. Citado na página 9.

BARRON, A.; RISSANEN, J.; YU, B. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, IEEE, v. 44, n. 6, p. 2743–2760, 1998. Citado 2 vezes nas páginas 3 e 9.

BEGUM, N. et al. Towards a minimum description length based stopping criterion for semi-supervised time series classification. In: IEEE. *14th International Conference on Information Reuse Integration (IRI'13)*. São Francisco, CA, EUA, 2013. p. 333–340. ISBN 9781479910502. ISSN 21945357. Citado 2 vezes nas páginas 14 e 16.

BELLMAN, R. *Dynamic Programming*. 1. ed. [S.l.]: Princeton University Press, 1957. 342 p. ISBN 069107951X. Citado na página 26.

BENAVOLI, A.; CORANI, G.; MANGILI, F. Should We Really Use Post-Hoc Tests Based on Mean-Ranks? *Journal of Machine Learning Research*, v. 17, p. 1–10, 2016. ISSN 15337928. Citado na página 38.

BOSIN, A.; DESSÌ, N.; PES, B. High-Dimensional Micro-array Data Classification Using Minimum Description Length and Domain Expert Knowledge. *Advances in Applied Artificial Intelligence*, Springer Berlin Heidelberg, v. 4031, p. 790 – 799, 2006. Citado na página 14.

BOTTOU, L. Large-Scale Machine Learning with Stochastic Gradient Descent. In: *19th International Conference on Computational Statistics (COMPSTAT'10)*. Paris, França: Springer, 2010. p. 177–186. ISBN 0269-2155. ISSN 0269-2155. Citado 3 vezes nas páginas 29, 43 e 44.

BRAGA, I. A.; LADEIRA, M. Filtragem adaptativa de spam com o princípio minimum description length. In: *Anais do XXVIII Congresso da Sociedade Brasileira de Computação (SBC'08)*. Belém, Brasil: [s.n.], 2008. p. 11–20. Citado na página 15.

BRATKO, A. et al. Spam Filtering Using Statistical Data Compression Models. *Journal of Machine Learning Research*, Microtome Publishing, v. 7, n. 12, p. 2673–2698, 2006. ISSN 15324435. Citado na página 15.

BREIMAN, L. Bagging predictors. *Machine learning*, v. 24, n. 2, p. 123–140, 1996. Citado na página 39.

CHAN, T. F.; GOLUB, G. H.; LEVEQUE, R. J. Algorithms for Computing the Sample Variance: Analysis and Recommendations. *The American Statistician*, Taylor & Francis, v. 37, n. 3, p. 242–247, 1983. ISSN 00031305. Citado na página 26.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, v. 20, n. 3, p. 273–297, 1995. Citado 2 vezes nas páginas 1 e 39.

COVER, T. M.; THOMAS, J. A. Elements of information. In: *Elements of information*. [S.l.]: John Wiley & Sons, Inc., 1991. cap. 3, p. 50–51. Citado 2 vezes nas páginas 6 e 7.

CRAMMER, K. et al. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, v. 7, n. Mar, p. 551—-585, 2006. ISSN 1532-4435. Citado na página 43.

DEMŠAR, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, Microtome Publishing, v. 7, n. 1, p. 1–30, 2006. Citado na página 38.

DOMINGOS, P. The role of Occam's Razor in knowledge discovery. *Data mining and knowledge discovery*, Springer, v. 3, n. 4, p. 409–425, 1999. ISSN 13845810 (ISSN). Citado 2 vezes nas páginas 1 e 18.

DUDA, R. O.; HART, P. E. *Pattern Classification and Scene Analysis*. 1. ed. [S.l.]: Wiley, 1973. 512 p. ISBN 0471223611. Citado 2 vezes nas páginas 1 e 39.

FERNÁNDEZ-DELGADO, M. et al. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, v. 15, p. 3133–3181, 2014. ISSN 1532-4435. Citado na página 35.

FERREIRA, J.; MATOS, D. M.; RIBEIRO, R. Fast and Extensible Online Multivariate Kernel Density Estimation. *CoRR*, abs/1606.0, n. 1, p. 1–17, 2016. Citado 4 vezes nas páginas 20, 22, 24 e 25.

FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian Network Classifiers. *Machine Learning*, Springer, v. 29, n. 2, p. 131–163, 1997. ISSN 1573-0565. Citado na página 13.

FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian Network Classifiers. *Machine learning*, v. 29, p. 131–163, 1997. ISSN 0885-6125. Citado na página 27.

GREGOR, K. et al. Deep AutoRegressive Networks. In: *31st International Conference on Machine Learning (ICML'14)*. Pequim, China: [s.n.], 2014. v. 32, p. 1242—-1250. ISBN 9781634393973. Citado na página 14.

GRÜNWALD, P. Model Selection Based on Minimum Description Length. *Journal of Mathematical Psychology*, Elsevier, v. 44, p. 133–152, 2000. ISSN 0022-2496. Citado 2 vezes nas páginas 15 e 3.

GRÜNWALD, P. A tutorial introduction to the minimum description length principle. *Advances in minimum description length: Theory and applications*, MIT Press, v. 1, n. 1, p. 23–81, 2005. Citado 6 vezes nas páginas 3, 9, 11, 17, 23 e 34.

GRÜNWALD, P.; HARREMOËS, P. Finiteness of redundancy, regret, shtarkov sums, and jeffreys integrals in exponential families. In: IEEE. *2009 IEEE International Symposium on Information Theory (ISIT'09)*. Seul, Coréia do Sul, 2009. p. 714–718. ISBN 9781424443130. ISSN 21578102. Citado na página 11.

HANSEN, M. H.; YU, B. Model Selection and the Principle of Minimum Description Length. *Journal of the American Statistical Association*, Taylor & Francis, v. 96, n. 454, p. 746–774, 2001. ISSN 0162-1459. Citado na página 9.

HINTON, G. E.; ZEMEL, R. S. Autoencoders, Minimum Description Length and Helmholtz free Energy. In: *Advances in Neural Information Processing Systems (NIPS'93)*. Denver, CO, EUA: NIPS Foundation, 1993. v. 1, n. 1, p. 1–9. ISBN 1049-5258. ISSN 15205207. Citado na página 13.

HU, B. et al. *Using the minimum description length to discover the intrinsic cardinality and dimensionality of time series*. [S.l.]: Springer, 2015. 358–399 p. ISSN 13845810. ISBN 1061801403. Citado na página 15.

KOLMOGOROV, A. N. On Tables of Random Numbers. *Sankhya: The Indian Journal of Statistics, Series A*, Springer, v. 53, n. 4, p. 369–376, 1963. Citado 3 vezes nas páginas 1, 3 e 17.

KONONENKO, I. The minimum description length based decision tree pruning. In: *Pacific Rim International Conference on Artificial Intelligence (PRICAI'98)*. Singapura, Singapura: Springer, 1998. p. 228–237. Citado na página 13.

KRAFT, L. G. *A device for quantizing, grouping, and coding amplitude-modulated pulses*. 1—-66 p. Tese (Doutorado) — Massachusetts Institute of Technology, 1949. Citado na página 5.

KRISTAN, M.; LEONARDIS, A. Online discriminative kernel density estimator with gaussian kernels. *Systems and Cybernetics*, IEEE, v. 44, n. 3, p. 355–365, 2014. ISSN 21682267. Citado 5 vezes nas páginas 17, 35, 40, 41 e 43.

KRISTAN, M.; LEONARDIS, A.; SKOČAJ, D. Multivariate online kernel density estimation with Gaussian kernels. *Pattern Recognition*, Elsevier, v. 44, n. 10-11, p. 2630–2642, 2011. ISSN 00313203. Citado 5 vezes nas páginas 19, 20, 22, 24 e 34.

LAM, W.; BACCHUS, F. Learning Bayesian Belief Networks: An Approach Based on the Mdl Principle. *Computational Intelligence*, Wiley Online Library, v. 10, n. 3, p. 269–293, 1994. ISSN 1467-8640. Citado na página 13.

LANGLEY, P.; JOHN, G. H. Estimating continuous distributions in Bayesian classifier. In: *11th conference on Uncertainty in Artificial Intelligence (UAI'95)*. Montreal, Canadá: Morgan Kaufmann Publishers Inc., 1995. p. 399–406. ISBN 1-55860-385-9. ISSN 1558603859. Citado na página 19.

LANGLEY, P.; SAGE, S. Induction of Selective Bayesian Classifiers. In: *10th International Conference on Uncertainty in Artificial Intelligence (UAI'94)*. Seattle, WA, EUA: Morgan Kaufmann Publishers Inc., 1994. p. 399–406. ISBN 1-55860-332-8. Citado na página 27.

LI, J.; MOUCHÈRE, H.; VIARD-GAUDIN, C. An annotation assistance system using an unsupervised codebook composed of handwritten graphical multi-stroke symbols. *Pattern Recognition Letters*, Elsevier B.V., v. 35, n. 1, p. 46–57, 2014. ISSN 01678655. Citado na página 14.

LICHMAN, M. *UCI Machine Learning Repository*. 2013. Citado 2 vezes nas páginas 2 e 35.

LING, R. F. Comparison of Several Algorithms for Computing Sample Means and Variances. *Journal of the American Statistical Association*, Taylor & Francis, v. 69, n. 348, p. 859, dec 1974. ISSN 01621459. Citado na página 26.

LU, J.; YANG, Y.; WEBB, G. I. Incremental discretization for naive-bayes classifier. In: SPRINGER. *2nd International Conference on Advanced Data Mining and Applications (ADMA'06)*. Xian, China, 2006. p. 223–238. ISBN 3-540-37025-0. Citado na página 19.

LÜTHKE, J. *Location Prediction Based on Mobility Patterns in Location Histories*. 1—-40 p. Tese (M.Sc. Thesis) — Hamburg University of Technology, 2013. Citado na página 22.

MACKAY, D. J. C. *Information Theory, Inference and Learning Algorithms*. [S.l.]: Cambridge University Press, 2003. 628 p. ISSN 0263-5747. ISBN 0521642981. Citado na página 12.

MAHALANOBIS, P. C. On the Generalized Distance in Statistics. *Proceedings of National Institute of Sciences (India)*, v. 2, n. 1, p. 49–55, 1936. Citado na página 30.

MCLACHLAN, G.; PEEL, D. *Finite mixture models*. [S.l.]: John Wiley & Sons, Inc., 2004. Citado na página 19.

MIETTINEN, P.; VREEKEN, J. MDL4BMF: Minimum Description Length for Boolean Matrix Factorization. *ACM Transactions on Knowledge Discovery from Data (TKDD'14)*, ACM, v. 8, n. 4, p. 1–31, 2014. ISSN 15564681. Citado na página 15.

PEARSON, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, Taylor & Francis, v. 2, n. 1, p. 559–572, 1901. ISSN 1941-5982. Citado na página 26.

POTAPOV, A. S. Principle of representational minimum description length in image analysis and pattern recognition. *Pattern Recognition and Image Analysis*, v. 22, n. 1, p. 82–91, 2012. ISSN 1054-6618. Citado na página 14.

QUINLAN, J. R.; RIVEST, R. L. Inferring decision trees using the minimum description lenght principle. *Information and Computation*, Elsevier, v. 80, n. 3, p. 227–248, 1989. ISSN 10902651. Citado 2 vezes nas páginas 13 e 15.

RISSANEN, J. Modeling by shortest data description. *Automatica*, Elsevier, v. 14, n. 5, p. 465–471, sep 1978. ISSN 00051098. Citado 4 vezes nas páginas 1, 3, 6 e 17.

RISSANEN, J. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, The Institute of Mathematical Statistics, v. 11, n. 2, p. 416–431, 1983. ISSN 00905364. Citado 3 vezes nas páginas 3, 6 e 9.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in .... *Psychological Review*, v. 65, n. 6, p. 386–408, 1958. ISSN 1939-1471(Electronic);0033-295X(Print). Citado 2 vezes nas páginas 1 e 43.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, n. 6088, p. 533–536, oct 1986. ISSN 0028-0836. Citado na página 29.

RUSSELL, S. J. et al. *Artificial intelligence: a modern approach*. 1st. ed. [S.l.]: Prentice Hall, 1995. 947 p. ISSN 09569944. ISBN 0131038052. Citado na página 12.

SALTON, G.; MCGILL, M. J. *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986. Citado na página 39.

SARAVANAN, N.; SIDDABATTUNI, V. N. S. K.; RAMACHANDRAN, K. I. A comparative study on classification of features by SVM and PSVM extracted using Morlet wavelet for fault diagnosis of spur bevel gear box. *Expert Systems with Applications*, v. 35, n. 3, p. 1351–1366, 2008. ISSN 09574174. Citado na página 30.

SHEINVALD, J.; DOM, B.; NIBLACK, W. A modeling approach to feature selection. In: *10th International Conference on Pattern Recognition (ICPR'90)*. Atlantic City, NJ, USA: IEEE, 1990. v. 1, n. 408, p. 535–539. ISBN 0-8186-2062-5. Citado na página 14.

SILVA, R. M.; ALMEIDA, T. A.; YAMAKAMI, A. *Detecção Automática de SPIM e SMS Spam usando Método baseado no Princípio da Descrição mais Simples*. Campinas, Brasil, 2016. 1–12 p. Citado na página 2.

SILVA, R. M.; ALMEIDA, T. A.; YAMAKAMI, A. Towards Web Spam Filtering Using a Classifier Based on the Minimum Description Length Principle. In: *15th IEEE International Conference on Machine Learning and Applications (ICMLA'16)*. Anaheim, CA, EUA: IEEE, 2016. p. 470–475. ISBN 978-1-5090-6167-9. Citado 4 vezes nas páginas 2, 15, 17 e 18.

SILVA, R. M.; ALMEIDA, T. A.; YAMAKAMI, A. MDLText: An efficient and lightweight text classifier. *Knowledge-Based Systems*, v. 118, p. 152–164, feb 2017. ISSN 09507051. Citado 6 vezes nas páginas 15, 16, 23, 30, 31 e 54.

SILVERMAN, B. W. *Density Estimation for Statistics and Data Analysis*. 1. ed. Boston, MA: Springer, 1986. ISBN 978-0-412-24620-3. Citado na página 19.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, Elsevier Ltd, v. 45, n. 4, p. 427–437, 2009. ISSN 03064573. Citado na página 37.

SRIVASTAVA, N. et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, v. 15, p. 1929–1958, 2014. ISSN 15337928. Citado na página 40.

TATAW, O. M.; RAKTHANMANON, T.; KEOGH, E. Clustering of symbols using minimal description length. In: IEEE. *12th International Conference on Document Analysis and Recognition (ICDAR'13)*. Washington, DC, EUA, 2013. p. 180–184. ISBN 978-0-7695-4999-6. ISSN 15205363. Citado 2 vezes nas páginas 14 e 15.

Van Leeuwen, M.; SIEBES, A. StreamKrimp: Detecting Change in Data Streams. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML'08)*. Antwerp, Belgium: Springer, 2008. p. 672–687. Citado 2 vezes nas páginas 12 e 18.

WAND, M. P.; JONES, M. C. *Kernel smoothing*. 1. ed. [S.l.]: CRC Press, 1994. 224 p. ISBN 9780412552700. Citado na página 22.

WELFORD, B. P. Note on a Method for Calculating Corrected Sums of Squares and Products. *Technometrics*, American Statistical Association, v. 4, n. 3, p. 419, aug 1962. ISSN 00401706. Citado na página 26.

XIANG, Z. et al. Novel Naive Bayes based on Attribute Weighting in Kernel Density Estimation. In: *Joint 7th International Conference on Soft Computing and Intelligent Systems and 15th International Symposium on Advanced Intelligent Systems (SCIS&ISIS'14)*. Kitakyushu, Japão: IEEE, 2014. p. 1439–1442. ISBN 978-1-4799-5955-6. Citado na página 27.

ZAIDI, N. A. et al. Alleviating Naive Bayes attribute independence assumption by attribute weighting. *Journal of Machine Learning Research*, Microtome Publishing, v. 14, n. 1, p. 1947–1988, 2013. ISSN 15324435. Citado na página 27.

ZAR, J. H. *Biostatistical Analysis*. 5. ed. [S.l.]: Prentice Hall, 2009. ISBN 0131008463. Citado na página 37.

ZIVKOVIC, Z.; HEIJDEN, F. van der. Recursive unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 5, p. 651–6, 2004. ISSN 0162-8828. Citado na página 19.