Protótipo de um Método de Classificação por Descrição Mínima

Breno Lima de Freitas, Tiago A. Almeida, Renato M. Silva

[1]Departamento de Computação (DComp)
Universidade Federal de São Carlos (UFSCar) – Sorocaba – SP – Brasil

brenolimadefreitas@gmail.com, talmeida@ufscar.br,
renatoms@dt.fee.unicamp.br

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 the best 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 future samples not overfitting on the known data. In this paper, we introduce the initial prototype of 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. Although the proposed method being an initial prototype, the conducted experiments on well-known public datasets showed that it performs as good as traditional classification approaches.*

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 apresentada uma proposta inicial de 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 e numéricos. Apesar de ser um protótipo inicial, experimentos com bases de dados públicas e bem conhecidas mostraram que ele é competitivo com métodos consolidados de classificação.*

1. Introdução

Ao longo das últimas décadas, a quantidade de dados tem aumentado em ritmo sem precedentes, o armazenamento e manipulação das informações tem se tornado mais acessível e o poder de processamento computacional tem aumentado. Ao mesmo tempo, tem crescido a necessidade de extrair informações dos dados de forma automática. Nesse contexto, diversos métodos de aprendizado de máquina vêm sendo propostos com a finalidade de categorizar (ou classificar) dados. Eles baseiam-se na seleção de um modelo, dada uma hipótese, que tem como objetivo ajustar-se a um conjunto de dados conhecidos para que possa computar uma saída (classe) para uma amostra nunca observada. 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 que compõem o conjunto de dados de treinamento, quanto para amostras ainda não observadas.

Dado o grande número de possíveis modelos que podem ser usados para fazer predições, os métodos de classificação utilizam critérios de seleção distintos (*e.g.*, probabilidade, otimização, distância) para tentar selecionar o melhor modelo possível. Uma possível estratégia de seleção de modelos é a utilização de critérios baseados na *navalha de Occam*. Esse é um princípio filosófico, cunhado por William Occam, um filósofo e frade inglês, baseado na seguinte ideia: "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, dada a presença de múltiplas opções que descrevem o mesmo problema, seja dada preferência ao modelo mais simples.

Uma das mais conhecidas formalizações para o uso da navalha de Occam no problema de seleção de modelos foi definida por Rissanen [Rissanen 1978] por meio do princípio da descrição mais simples (*minimum description length* – MDL). Tal princípio também está enraizado nas ideias da complexidade de Komolgorov [Kolmogorov 2016] e define que o modelo que melhor se adapta aos dados e possui menor tamanho de descrição – portanto, o menos complexo – deve ser o modelo selecionado. Deste modo, o MDL elege modelos que preservam um equilíbrio favorável entre a sua capacidade de ajuste aos dados de treinamento e sua complexidade, evitando naturalmente o problema de sobreajustamento (*overfitting*).

O MDL foi utilizado em diferentes contextos de aprendizado de máquina, tais como: geração de árvores de decisão [Ross Quinlan e Rivest 1989, Kononenko 1998], redes Bayesianas [Lam e Bacchus 1994, Friedman et al. 1997] e categorização de textos [Bratko et al. 2006, Braga e Ladeira 2008, Silva et al. 2017]. O método MDLText, introduzido por [Silva et al. 2017], é um bom exemplo de sucesso da utilização de métodos baseados no princípio MDL em problemas de classificação. Porém, ele é especializado em problemas de categorização de texto.

Neste trabalho, é oferecida uma proposta inicial de um método de classificação que incorpora as vantagens teóricas oferecidas pelo princípio MDL. O método proposto, nomeado *Aprendiz de Descritores de Mistura Gaussiana (Gaussian Mixture Descriptor Learner* – GMDL), é genérico e pode ser utilizado em qualquer problema de classificação binária ou multi-classe, cujos dados possam ser representados por atributos categóricos ou numéricos.

O GMDL, por ser baseado no princípio MDL, é robusto a um dos principais problemas que afeta vários métodos de classificação da literatura: o sobreajustamento aos dados. Além disso, ele cria seu modelo de predição de maneira incremental. Portanto, diferente da maioria do métodos de classificação tradicionais, ele não é limitado ao cenário de aprendizado em *batch*, onde todos os dados de treinamento devem ser apresentados de uma única vez. O GMDL também pode ser utilizado em problemas que possuem um fluxo de dados contínuo, onde o modelo de predição precisa ser atualizado quando novas amostras são apresentadas para treinamento. Atualmente, na era do *big data*, onde muitos problemas contém uma quantidade massiva de dados, métodos que geram modelos de predição de forma incremental são bastante desejáveis, pois essa característica os torna naturalmente escaláveis [Hoi et al. 2014].

O restante desse trabalho está organizado da seguinte forma: a Seção 2 apresenta os principais conceitos básicos sobre o princípio MDL. O método de classificação proposto neste trabalho é apresentado na Seção 3. Na Seção 4, é descrita a metodologia experimental. A Seção 5 apresenta os resultados obtidos. Por fim, a Seção 6 oferece as conclusões e direcionamentos para trabalhos futuros.

2. O princípio da descrição mais simples

Rissanen [Rissanen 1978, Rissanen 1983] 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 o menor tamanho de descrição deve ser priorizado. Este princípio, oriundo da Teoria da Informação, delineia que quanto mais se conhece os dados, maior será a regularidade descoberta e, portanto, mais pode-se comprimí-los [Grünwald 2005].

O MDL possui raízes na complexidade de Kolmogorov [Kolmogorov 2016], definida como o menor tamanho de um programa que imprime uma dada sequência e finaliza. Portanto, quanto menor a complexidade de Kolmogorov de um modelo, mais conhecimento esse modelo possui sobre a sequência sendo codificada, já que existe uma codificação menor dos dados. Logo, o modelo com menor complexidade de Kolmogorov deve ser selecionado para representar a sequência [Barron et al. 1998]. A unificação dos conceitos da Navalha de Occam e da complexidade de Kolmogorov oferecem ao MDL um equilíbrio benéfico entre a seleção da complexidade do modelo e seu ajuste aos dados, o que evita, portanto, um modelo complexo sobreajustado aos dados [Grünwald 2000].

O MDL procura por uma codificação C de um conjunto de dados D que consiga descrevê-lo de maneira única da menor forma possível. Isto é, procura-se um C no qual $L_C(D)$ seja o menor possível, onde L_C é uma função que descreve o tamanho dos dados codificados com o auxílio de C. No contexto desta pesquisa, entre todas as possíveis codificações, foram utilizadas aquelas descritas na base binária, usando *bit* como medida de informação. Como tratam-se de codificações binárias, no restante do texto será utilizado *log* para representar o logaritmo na base 2.

O MDL foi originalmente cunhado como um método de duas partes – também chamado de MDL primitivo (do inglês, *crude MDL*) - onde, dado um conjunto de modelos candidatos M, o modelo M que deve ser escolhido é aquele que minimiza a seguinte equação:

$$M_{MDL} := \arg\min_{M \in \mathcal{M}} [L(M) + L(D|M)]. \tag{1}$$

É possível mostrar que o tamanho ótimo de descrição é limitado inferiormente por $-\log P(x^n)$ para uma dada sequência de símbolos $x^n \in \mathcal{X}^n$ em um alfabeto $\mathcal{X}$. Uma prova para esta afirmação pode ser encontrada em [MacKay 2003, Seção 5.3]. Deste modo, obtém-se uma boa estimativa para $L(D|M)$, uma vez 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 é aceitável para a segunda parte da Equação 1, uma vez que (i) a log-semelhança é um método estatístico padrão para medir o quão bem um modelo está ajustado 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.

[Rissanen 1983, Barron e Cover 1991, Barron et al. 1998, Hansen e Yu 2001] deram os primeiros passos para refinar o MDL usando o que é chamado de codificação universal, abandonando o MDL de duas partes em favor de uma descrição mais formal e direta [Grünwald 2005]. Um fator $\bar{L}_{\mathcal{L}}$ é uma codificação universal se comprime todo símbolo de uma sequência quase tão bem quanto a codificação em $\mathcal{L}$ que mais comprimiria tal sequência. Idealmente, busca-se uma codificação que seja universal no sentido que possa comprimir ao máximo todas as sequências dado o conjunto das funções de tamanho de código. No entanto, como mostrado por [Grünwald 2005], não é possível obter tal função. Portanto, busca-se uma codificação que seja aproximadamente tão boa quanto a melhor, que comprime todas as sequências possíveis. Em [Grünwald 2005, Proposição

2.14] foi mostrado que é possível encontrar uma distribuição, que aproxime razoavelmente bem os dados de uma codificação, utilizando uma distribuição normalizada baseada na complexidade do modelo utilizado para a codificação. Deste modelo surge o *MDL refinado*, que pode ser matematicamente formalizado pela seguinte equação:

$$M_{MDL} := \arg\min_{M \in \mathcal{M}} \bar{L}(D|M). \tag{2}$$

Não existe um consenso de como usar de maneira prática a codificação universal empregada pelo *MDL refinado*. Sendo assim, no método proposto neste trabalho, o tamanho de descrição calculado pela Equação 2 considera a distribuição de probabilidade dos dados, conforme detalhado na seção apresentada a seguir.

3. Aprendiz de Descritores de Mistura Gaussiana

Um dado (ou amostra) $\vec{x}$ pode ser representado por um conjunto de atributos reais $(x_1, \cdots, x_n)$. Utilizando os conceitos do MDL, procura-se pelo modelo M que mais comprima $\vec{x}$. Cada modelo M induz uma distribuição de probabilidade P para uma dada codificação de dados. É conhecido que a menor codificação possível para uma determinada codificação induzida por uma distribuição de probabilidade P é definida por $\lceil -\log P \rceil$ [MacKay 2003, Seção 5.3].

Idealmente, se a distribuição de probabilidade real que descreve um conjunto de dados for conhecida previamente, sob uma perspectiva Bayesiana, é possível prever com perfeição uma amostra não rotulada. No entanto, em aplicações reais isso não é possível, sendo necessário fazer uma estimativa de tal distribuição. Portanto, deseja-se alguma estimativa de distribuição de probabilidade que evite discretizações e seja incremental, para que seja mantida a habilidade do método de aprender com novas amostras para um fluxo contínuo de dados. Deste modo, neste trabalho, a Equação 2 foi utilizada no protótipo de um método de classificação baseado em distribuições de probabilidade, nomeado Aprendiz de Descritores de Mistura Gaussiana (*Gaussian Mixture Descriptor Learner* – GMDL), onde cada modelo induz uma distribuição para cada uma das classes do problema sendo avaliado.

No GMDL, para estimar uma função de densidade de probabilidade para o conjunto de dados, foi utilizada a forma clássica e paramétrica que é assumir que a distribuição é uma mistura Gaussiana [McLachlan e Peel 2004]. No entanto, tal técnica, no geral, depende da definição prévia dos parâmetros da mistura [McLachlan e Peel 2004, Zivkovic e van der Heijden 2004]. Estimar tais parâmetros não é uma tarefa trivial. O uso, por exemplo, do número errado de componentes, pode fazer com que o método não represente bem a função real [Kristan et al. 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). Este método possui uma função chamada de *kernel*, que integra a um e tem média zero. Ele também utiliza um parâmetro b chamado de *largura de banda*, cujo papel é suavizar a soma dos *kernels* do estimador da distribuição de probabilidade. A seleção do *kernel* e da largura de banda tem grande influência na estimativa da função [Silverman 1986]. Uma escolha clássica de *kernel* é o Gaussiano, que mapeia a entrada para uma saída normalmente distribuída.

O EDK possui uma das características desejadas para a distribuição apresentada: não é necessária a discretização dos dados a priori. Com isso, é possível estimar as distribuições de probabilidade sem sofrer as perdas de dados oriundas de discretizações e manter uma estimativa da distribuição real dos dados. Essa é uma característica desejada

para o método que está sendo proposto nesse trabalho, uma vez que deseja-se que ele seja genérico e possa ser usado em problemas com atributos numéricos sem a necessidade de realizar discretizações. Porém, também deseja-se que o GMDL gere seu modelo de predição de forma incremental. Portanto, para que o EDK seja incorporado no método que está sendo proposto, é necessário que o EDK também seja incremental.

3.1. Estimativa de Densidade de Kernel Online

Uma das grandes dificuldades em se transformar o EDK em um método incremental é que o número de componentes que o método utiliza para a estimativa da distribuição cresce linearmente para cada nova amostra apresentada [Kristan et al. 2011]. Isso deve-se ao fato dele precisar manter informações o suficiente para generalizar para novas amostras sem a necessidade de coletar novamente informações das amostras vistas anteriormente [Ferreira et al. 2016]. Recentemente, [Kristan et al. 2011] propuseram um método de estimativa de densidade de *kernel* totalmente incremental, chamado de oKDE. Esta abordagem é ideal para contemplar a segunda característica desejada para uma estimativa de função de distribuição de probabilidade.

A ideia principal do oKDE é que, ao invés de tentar construir diretamente a distribuição alvo, mantém-se 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 apresentada a seguir:

$$p_{DD}(x) := \lim_{\sigma \to 0} \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{3}$$

Cada distribuição amostral é, portanto, um conjunto de funções de densidade infinita em um único ponto, sendo que uma distribuição amostral pode ser vista como uma mistura Gaussiana. Neste caso, a largura de banda é dada pela matriz de covariância da distribuição.

Uma vez que manter uma função Delta-Dirac para cada observação não é adequado para um cenário de aprendizado incremental, o número de componentes no oKDE é reduzido por meio de 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 m-dimensional, tal que $m < n$, onde procura-se por uma estimativa de distribuição que não ultrapasse uma dada taxa de erro D_{th}. Durante a estimativa do oKDE, também é utilizado um parâmetro f chamado de *fator de esquecimento*. Ele é utilizado para atribuir um peso às amostras antigas, diminuindo a influência delas em um fluxo de dados – ideal para cenários onde o fluxo de dados é temporal. Um fator de esquecimento igual a um significa que os pesos das amostras não se diferenciam por sua posição temporal.

O segundo ponto importante na descrição do oKDE é a estimativa da largura de banda ótima. A qualidade da estimativa de banda tem impacto direto na qualidade da distribuição aproximada. A estimativa da largura de banda pode ser vista como um problema de otimização onde deseja-se minimizar a distância da distribuição aproximada para a distribuição real [Lüthke 2013]. Usualmente, nesse cenário, utiliza-se a divergência de Kullback-Leibler como medida de distância [Kristan et al. 2011]. No entanto, a distribuição original não é conhecida. Para tais cenários, uma abordagem tradicional é utilizar o erro quadrático médio assintótico integrativo (do inglês, *asymptotic mean integrated squared error – AMISE*) [Wand e Jones 1994]. Os autores então a utilizam para definir a melhor largura de banda [Lüthke 2013].

[Ferreira et al. 2016] cunharam uma versão do oKDE, chamada de *xokde++*, que procura sanar problemas de inconsistência numérica e propor uma abordagem computacional mais robusta em termos de uso de memória e processamento. No *xokde++* também é mantida 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. O método que está sendo proposto neste trabalho utiliza essa versão melhorada do oKDE. No restante do texto, por simplicidade, o *xokde++* será referenciado como oKDE.

3.2. Predição incremental de amostras

O oKDE provê uma maneira não-parametrizada e incremental de estimar uma função de densidade para um conjunto de dados. Utilizando o oKDE, pode-se estimar densidades para compor os modelos avaliados pelo GMDL. Com esta aproximação espera-se que, com o número crescente de amostras, obtenha-se uma estimativa cada vez mais fidedigna da distribuição real.

É possível definir a função de tamanho de descrição $\hat{L}$ como a soma das descrições de seus atributos quando codificados pela aproximação de suas funções de densidade p' para cada classe. Como mostrado em [MacKay 2003, Seção 5.3], sabe-se que a menor codificação que pode ser obtida é delimitada superiormente pelo logaritmo de sua função de probabilidade. Portanto, $\hat{L}$ pode ser formalmente definido como:

$$\hat{L}(\vec{x}|c) := \sum_{i=1}^{n} \lceil -\log p'_{(i,c)}(x_i) \rceil. \tag{4}$$

Na Equação 4, $\vec{x}$ é um vetor $(x_1, \cdots, x_n)$ de atributos e $c \in K$, a classe avaliada. Note que p' pode tender ao infinito, quando avalia-se uma função Delta-Dirac, e pode ser zero, caso não haja amostras o suficiente para medir a densidade em um ponto. Portanto, a Equação 4 pode ser reescrita como:

$$\hat{L}(\vec{x}|c) := \sum_{i=1}^{n} \lceil -\log p_{(i,c)}(x_i) \rceil, \tag{5}$$

onde o valor de $p_{(i,c)}$ pode ser calculado da seguinte forma:

$$p_{(i,c)} := \begin{cases} 2^{-\Omega} & p'_{(i,c)} \to \infty \ \vee \ p'_{(i,c)} = 0 \\ p'_{(i,c)} & c.c. \end{cases} \tag{6}$$

Na Equação 6, Ω é um meta-parâmetro que funciona como regularizador nos casos onde não há informação o suficiente para computar-se uma probabilidade. É possível também alterar o fator de esquecimento f original no oKDE como um meta-parâmetro de p'. Para simplificar a notação, f será omitido.

O fator $L \in [0, 1]$ será definido como a versão normalizada de $\hat{L}$, isto é, o vetor resultante da divisão de cada elemento de $\hat{L}$ pela soma de todos os elementos. Tal versão provê melhor estabilidade numérica e, convenientemente, uma pseudo-probabilidade quando avaliada como $1 - L(\vec{x})$.

Deste modo, a primeira versão do GMDL pode ser definida pela seguinte equação:

$$GMDL(\vec{x}) := \arg\min_{c \in K} L(\vec{x}|c). \tag{7}$$

Diante do exposto, o GMDL pode ser resumido da seguinte forma: dado um conjunto de possíveis classes $c_1, c_2, ..., c_{|K|}$, quando um dado $\vec{x}$ desconhecido for apresentado, ele será rotulado com a classe c_k que possui o menor tamanho de descrição $L(\vec{x}|c_k)$ em relação a $\vec{x}$.

3.3. Suavização de funções Delta-Dirac e distribuições degeneradas

Originalmente, o método proposto por [Kristan et al. 2011] (oKDE), que foi incorporado ao GMDL, era suscetível às distribuições degeneradas, isto é, distribuições com baixa variância.

O método refinado por [Ferreira et al. 2016] tentou mitigar esse problema e melhorar a estabilidade numérica. Os autores utilizaram o logaritmo da matriz de covariância para suas computações. Logaritmos são amplamente utilizados 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 para minimizar o problema de distribuições degeneradas, proposta por [Ferreira et al. 2016], é computar sua decomposição em autovalores e autovetores, além de analisar seus autovalores procurando por aqueles menores que 10^{-9}, os quais são corrigidos por 1% da média dos autovalores. Contudo, é importante observar que esta técnica não soluciona o problema quando há apenas uma dimensão envolvida.

Para solucionar tal problema, um ruído $\mathcal{N}(0, \bar{\sigma}^2)$ pode ser adicionado. É fato conhecido que a soma de duas variáveis aleatórias normalmente distribuídas também é normal. Esta perturbação mantém o centro da distribuição e altera apenas seu desvio padrão. A primeira característica é muito útil, uma vez que, para funções Delta-Dirac, o ponto central de tal função continua possuindo 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 tornar-se menor do que o limiar de 10^{-9} definido por [Ferreira et al. 2016]. Este processo, dado o meta-parâmetro $\bar{\sigma}^2$, que determina o desvio padrão do ruído a ser aplicado, pode ser definido por:

$$p'_{(i,c)} \leftarrow \begin{cases} x_i & \sigma^2_{ic} \geq 10^{-9} \\ x_i + \sum_{n \sim \mathcal{N}(0,\bar{\sigma}^2)} n, \sigma^2_{ic} > 10^{-9} & c.c. \end{cases}.$$

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 [Tony F. Chan, Gene H. Golub 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.

4. Análise experimental

Para avaliar o desempenho do protótipo proposto do GMDL, foram realizados experimentos com bases de dados tradicionais, mostradas na Tabela 1, que estão disponíveis publicamente no site da UCI[1] [Lichman 2013]. Essas bases foram escolhidas por serem popularmente conhecidas, consolidadas e amplamente utilizadas em diversos trabalhos de aprendizado de máquina.

[1] *UC Irvine (UCI) Machine Learning Repository.* Disponível em: `http://archive.ics.uci.edu/ml/index.php`. Acessado em 28/06/2017.

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

	Tamanho		Composição das classes		Composição dos atributos	
Base de dados	m	n	$\|K\|$	Amostras por classe	Dados contínuos	Dados discretos
BreastCancer	569	30	2	212, 357	✓	
CoverType	581012	10	7	2747, 9493, 17367, 20510, 211840, 283301, 35754		✓
Iris	150	4	3	50, 50, 50	✓	
Letter	20000	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		✓
Skin	245057	3	2	50859, 194198	✓	
Wine	178	13	3	48, 59, 71	✓	✓
WineRed	1599	11	6	10, 18, 53, 199, 638, 681	✓	
WineWhite	4898	11	7	5, 20, 163, 175, 880, 1457, 2198		✓

Na Tabela 1, m corresponde ao número total de amostras, n é o número de atributos e $|K|$ é a quantidade de classes. É importante destacar que as bases escolhidas possuem características distintas, o que aumenta o escopo de validação do método proposto. Algumas delas tem um baixo número de amostras, enquanto outras superam 200 mil amostras. Algumas são binárias, enquanto outras são multiclasse, sendo que a maior delas tem 26 classes. Além disso, foram avaliadas bases de dados balanceadas e outras com alto grau de desbalanceamento. Por fim, foram avaliadas bases de dados com atributos contínuos e discretos.

4.1. Pré-processamento e avaliação

Em todos os experimentos, as amostras foram normalizadas para possuírem média zero e desvio padrão unitário. Além disso, para avaliar o desempenho de cada método, dividiu-se, de forma estratificada, cada base de dados em 80% para treinamento e validação e 20% para teste. As medidas de desempenho utilizadas foram a micro e a macro F-medida.

Apesar do método proposto neste trabalho ser naturalmente incremental, o cenário escolhido para os experimentos foi o aprendizado em *batch*, em que todos os dados de treinamento são apresentados de uma única vez aos métodos de classificação. O objetivo é analisar o método proposto no cenário clássico de aprendizado e compará-lo com técnicas tradicionais e consolidadas, sendo que a maioria deles não suporta aprendizado incremental.

4.2. Métodos

Para avaliar o desempenho do método proposto, seus resultados foram comparados aos obtidos por métodos clássicos da literatura: máquinas de vetores de suporte (SVM) [Cortes e Vapnik 1995], floresta aleatória (RF) [Breiman 1996], *naïve* Bayes [Lewis e Ringuette 1994], utilizando sua variação Gaussiana, que é propícia para dados contínuos (NB) e k-vizinhos mais próximos (kNN) [Salton e McGill 1986].

O GMDL foi implementado totalmente na linguagem de programação `C++`. Os demais métodos foram implementados na linguagem de programação `Python` usando funções disponíveis na biblioteca `Scikit-learn`[2].

Como o desempenho dos métodos SVM, RF, kNN e GMDL pode ser afetado pelas escolhas dos seus hiper-parâmetros, foram realizadas buscas em grade para encontrar os melhores valores possíveis. Elas foram realizadas na partição de treinamento, usando validação cruzada 5-*fold* estratificada. Nesse processo, a macro F-medida foi utilizada

[2] `Scikit-learn`, disponível em `https://scikit-learn.org`. Acessado em 04/07/2017.

para avaliar o desempenho dos métodos. Os intervalos de valores avaliados na busca em grade foram os seguintes:

- SVM: custo $\in \{0{,}5; 1; 3\}$, *kernel* $\in \{$'*rbf*'; '*linear*'$\}$ e $\gamma \in \left\{0{,}1; 1; \frac{1}{n}\right\}$;
- RF: número de árvores $\in \{5; 10; 20; 30\}$ e *criterion* (função usada para avaliar a qualidade de uma determinada divisão dos dados) $\in \{$'*gini*'; '*entropy*'$\}$;
- kNN: número de vizinhos $\in \{3; 5; 7; 9; 11\}$;
- GMDL: $\sigma^2 \in \{0{,}5; 1; 2; 3\}$ e $\Omega \in \{8; 16; 32; 64\}$.

Utilizou-se o fator de esquecimento unitário para o GMDL, tendo em vista que o cenário dos experimentos não é incremental.

5. Resultados

A Tabela 2 apresenta os valores de macro e micro F-medida obtidos por cada método. Os valores destacados em negrito indicam os melhores resultados por base de dados.

Tabela 2. Resultados obtidos na avaliação experimental.

Método	Macro F	Micro F
BreastCancer		
GMDL	**0,95**	**0,96**
kNN	0,94	0,95
NB	0,92	0,93
RF	**0,95**	**0,96**
SVM	0,92	0,93
CoverType		
GMDL	0,47	0,48
kNN	0,75	0,84
NB	0,45	0,63
RF	**0,91**	**0,95**
SVM	0,78	0,85
Iris		
GMDL	0,94	0,93
kNN	**0,97**	**0,97**
NB	0,93	0,93
RF	0,93	0,93
SVM	**0,97**	**0,97**
Letter		
GMDL	0,69	0,68
kNN	0,91	0,91
NB	0,64	0,64
RF	0,95	0,95
SVM	**0,96**	**0,96**

Método	Macro F	Micro F
Skin		
GMDL	0,92	0,94
kNN	**1,00**	**1,00**
NB	0,87	0,92
RF	**1,00**	**1,00**
SVM	0,99	0,99
Wine		
GMDL	**1,00**	**1,00**
kNN	0,95	0,94
NB	0,97	0,97
RF	**1,00**	**1,00**
SVM	0,97	0,97
WineRed		
GMDL	0,25	0,39
kNN	0,29	0,55
NB	0,31	0,54
RF	**0,32**	**0,65**
SVM	**0,32**	0,60
WineWhite		
GMDL	0,26	0,34
kNN	0,28	0,53
NB	0,23	0,41
RF	**0,39**	**0,66**
SVM	0,27	0,58

De modo geral, observa-se que o protótipo do método proposto foi marginalmente superior ao NB na maioria das bases de dados. É notório também que o método RF obteve quase sempre os melhores resultados. Porém, o GMDL obteve desempenho equiparável ao método RF na base de dados *Wine*, atingindo o melhor valor possível de macro e micro F-medida.

O GMDL obteve resultados inferiores à maioria dos métodos nas bases de dados altamente desbalanceadas como a *CoverType* e *WineRed*. O NB também obteve resultados ruins nessas duas bases de dados. Isso indica que métodos probabilísticos como o GMDL e NB podem ser mais afetados pelo problema de desbalanceamento do que métodos baseados em outras estratégias de seleção de modelo.

Tabela 3. *Ranking* dos métodos avaliados com base na macro F-medida.

Base de dados	GMDL	kNN	NB	RF	SVM
BreastCancer	1,5	3	4,5	1,5	4,5
CoverType	4	3	5	1	2
Iris	3	1,5	4,5	4,5	1,5
Letter	4	3	5	2	1
Skin	4,5	1,5	4,5	1,5	3
Wine	1,5	5	3,5	1,5	3,5
WineRed	5	4	3	1,5	1,5
WineWhite	4	2	5	1	3
***Rank* total**	**23**	**27,5**	**35**	**14,5**	**29**
***Rank* médio**	**2,88**	**3,44**	**4,38**	**1,81**	**3,63**

Para avaliar de forma mais precisa o desempenho geral dos métodos, foi realizado o teste não paramétrico de Friedman, usando as macro F-medidas obtidas pelos métodos e seguindo a metodologia apresentada em [Zar 2009, Seção 12.7]. A Tabela 3 apresenta os *rankings* dos métodos avaliados com base na macro F-medida. É importante destacar que apesar de ainda ser apenas um protótipo, o desempenho geral do GMDL apresentou *ranking* médio melhor que o de métodos estabelecidos, tais como SVM, kNN e NB.

O teste de Friedman verifica se é possível ou não rejeitar a hipótese nula, que afirma que os métodos possuem desempenhos equivalentes. Como foram usados 5 métodos e 8 bases de dados, para um intervalo de confiança $\alpha = 0{,}05$, o valor crítico na distribuição chi-quadrado é $9{,}3$. Dado que o valor crítico calculado pelo teste de Friedman foi $9{,}11$, a hipótese nula não pôde ser rejeitada e, portanto, é possível afirmar que os métodos foram estatisticamente equivalentes para as bases de dados utilizadas. Essa equivalência aos métodos tradicionais evidencia a capacidade do GMDL e o quão promissor ele pode ser, uma vez que muitas melhorias ainda podem ser efetuadas. Além disso, é importante destacar que o GMDL possui características altamente desejáveis em métodos de classificação, como ser naturalmente incremental, multiclasse e robusto ao sobreajustamento aos dados.

6. Conclusão

Neste trabalho, foi apresentado o protótipo de um método de classificação baseado no princípio MDL. Este método, nomeado GMDL, utiliza estimativas de densidade para cada um dos atributos das amostras em uma mistura Gaussiana e emprega a densidade em um ponto na mistura para fazer a predição da classe de amostras não-rotuladas. O método provê naturalmente uma troca benéfica entre a acurácia e a complexidade do modelo, uma característica oriunda da natureza da navalha de Occam e formalizada pelo princípio MDL, o que o torna robusto ao conhecido problema de sobreajustamento aos dados.

O método apresentado foi criado para ser genérico, isto é, poder ser aplicado em qualquer problema de classificação que possa ser representado por atributos categóricos ou numéricos. Ele ainda pode ser naturalmente aplicado em problemas multiclasse, sem a necessidade de utilização de técnicas de decomposição dos problemas em vários problemas binários. Ainda, ele é eficiente e gera seu modelo de predição de maneira incremental, o que o torna escalável e apto a ser aplicado em problemas de larga escala.

O desempenho do GMDL foi avaliado em um cenário de aprendizado em *batch*, usando bases de dados clássicas da literatura e comparado aos resultados obtidos por métodos considerados referência na área: NB, KNN, SVM e RF. O método proposto obteve o segundo melhor *ranking* médio e a análise estatística dos resultados evidenciou que o GMDL foi equivalente a todos os métodos usados na comparação. Esse resultado indica

que o GMDL, apesar de ainda estar em fase inicial de criação, já pode apresentar resultados no mesmo nível de métodos clássicos, mas podendo funcionar de maneira totalmente incremental e sem a necessidade de ajuste inicial nos dados.

Em trabalhos futuros, serão adotadas novas estratégias para tentar melhorar o poder preditivo do método. Por exemplo, uma função de atribuição de pesos aos atributos de acordo com seu poder preditivo poderá ser acoplada à função principal do GMDL. Também, está prevista a condução de experimentos no cenário de classificação *online* utilizando os dados sem qualquer tipo de tratamento. Ainda, pretende-se avaliar o GMDL em cenários onde os dados são esparsos e possuem alta dimensão, usando um número maior de bases de dados e explorando mais extensivamente os meta-parâmetros de cada método.

Agradecimentos

Os autores são gratos à agência de fomento CAPES (Proc. 1709642) pelo apoio financeiro a este projeto de pesquisa.

Referências

Barron, A. e Cover, T. (1991). Minimum complexity density estimation. *IEEE Transactions on Information Theory*, 37(4):1034–1054.

Barron, A., Rissanen, J., e Yu, B. (1998). The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760.

Braga, I. A. e Ladeira, M. (2008). Filtragem adaptativa de spam com o princípio minimum description length. In *Anais do XXVIII Congresso da Sociedade Brasileira de Computação (SBC'08)*, pages 11–20, Belém, Brasil. Sociedade Brasileira de Computação (SBC).

Bratko, A., Cormack, G. V., Filipič, B., Lynam, T. R., e Zupan, B. (2006). Spam Filtering Using Statistical Data Compression Models. *Journal of Machine Learning Research*, 7(12):2673–2698.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Cortes, C. e Vapnik, V. N. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Ferreira, J., Matos, D. M., e Ribeiro, R. (2016). Fast and Extensible Online Multivariate Kernel Density Estimation. *CoRR*, abs/1606.0(1):1–17.

Friedman, N., Geiger, D., e Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning*, 29(2):131–163.

Grünwald, P. (2000). Model Selection Based on Minimum Description Length. *Journal of mathematical psychology*, 44:133–152.

Grünwald, P. (2005). A tutorial introduction to the minimum description length principle. *Advances in minimum description length: Theory and applications*, 1(1):23–81.

Hansen, M. H. e Yu, B. (2001). Model Selection and the Principle of Minimum Description Length. *Journal of the American Statistical Association*, 96(454):746–774.

Hoi, S. C. H., Wang, J., e Zhao, P. (2014). Libol: A library for online learning algorithms. *Journal of Machine Learning Research*, 15(1):495–499.

Kolmogorov, A. N. (2016). On Tables of Random Numbers. *Sankhya: The Indian Journal of Statistics, Series A*, 53(4):369–376.

Kononenko, I. (1998). The minimum description length based decision tree pruning. In *Pacific Rim International Conference on Artificial Intelligence (PRICAI'98)*, pages 228–237, Singapura, Singapura. Springer, Springer.

Kristan, M., Leonardis, A., e Skočaj, D. (2011). Multivariate online kernel density estimation with Gaussian kernels. *Pattern Recognition*, 44(10-11):2630–2642.

Lam, W. e Bacchus, F. (1994). Learning Bayesian Belief Networks: An Approach Based on the Mdl Principle. *Computational Intelligence*, 10(3):269–293.

Lewis, D. D. e Ringuette, M. (1994). A Comparison of Two Learning Algorithms for Text Categorization. In *3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, pages 81–93, Las Vegas, NV, EUA. Information Science Research Institute, University of Nevada.

Lichman, M. (2013). UCI Machine Learning Repository.

Ling, R. F. (1974). Comparison of Several Algorithms for Computing Sample Means and Variances. *Journal of the American Statistical Association*, 69(348):859.

Lüthke, J. (2013). *Location Prediction Based on Mobility Patterns in Location Histories*. M.sc. thesis, Hamburg University of Technology.

MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*, volume 4. Cambridge University Press.

McLachlan, G. e Peel, D. (2004). *Finite mixture models*, volume 1. John Wiley & Sons, Inc.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5):465–471.

Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431.

Ross Quinlan, J. e Rivest, R. L. (1989). Inferring decision trees using the minimum description lenght principle. *Information and Computation*, 80(3):227–248.

Salton, G. e McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

Silva, R. M., Almeida, T. A., e Yamakami, A. (2017). MDLText: An efficient and lightweight text classifier. *Knowledge-Based Systems*, 118:152–164.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*, volume 26. CRC Press, 1 edition.

Tony F. Chan, Gene H. Golub, R. J. L. (1983). Algorithms for Computing the Sample Variance: Analysis and Recommendations. *The American Statistician*, 37(3):242–247.

Wand, M. P. e Jones, M. C. (1994). *Kernel smoothing*, volume 1. Crc Press, 1 edition.

Welford, B. P. (1962). Note on a Method for Calculating Corrected Sums of Squares and Products. *Technometrics*, 4(3):419.

Zar, J. H. (2009). *Biostatistical Analysis*. Prentice Hall, 5 edition.

Zivkovic, Z. e van der Heijden, F. (2004). Recursive unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):651–6.