Aprendiz de Descritores de Mistura Gaussiana

Defesa de Mestrado

Candidato: Breno Lima de Freitas
Orientador: Prof. Dr. Tiago Agostinho de Almeida

14 de Dezembro de 2017

Programa de Pós-graduação em Ciência da Computação – Universidade Federal de São Carlos

Agenda

1. Introdução
2. O Princípio da Descrição mais Simples
3. Aplicações do MDL em Aprendizado de Máquina
4. Motivação & Objetivos
5. Estimativa de f.d.p.
6. Construindo um método de classificação baseado no MDL
7. Avaliação experimental
8. Conclusão

Introdução

Classificação

(a) *Iris virginica*

(b) *Iris versicolor*

(c) *Iris setosa*

Figura 1: Flores do gênero *Iris* ©

Classificação

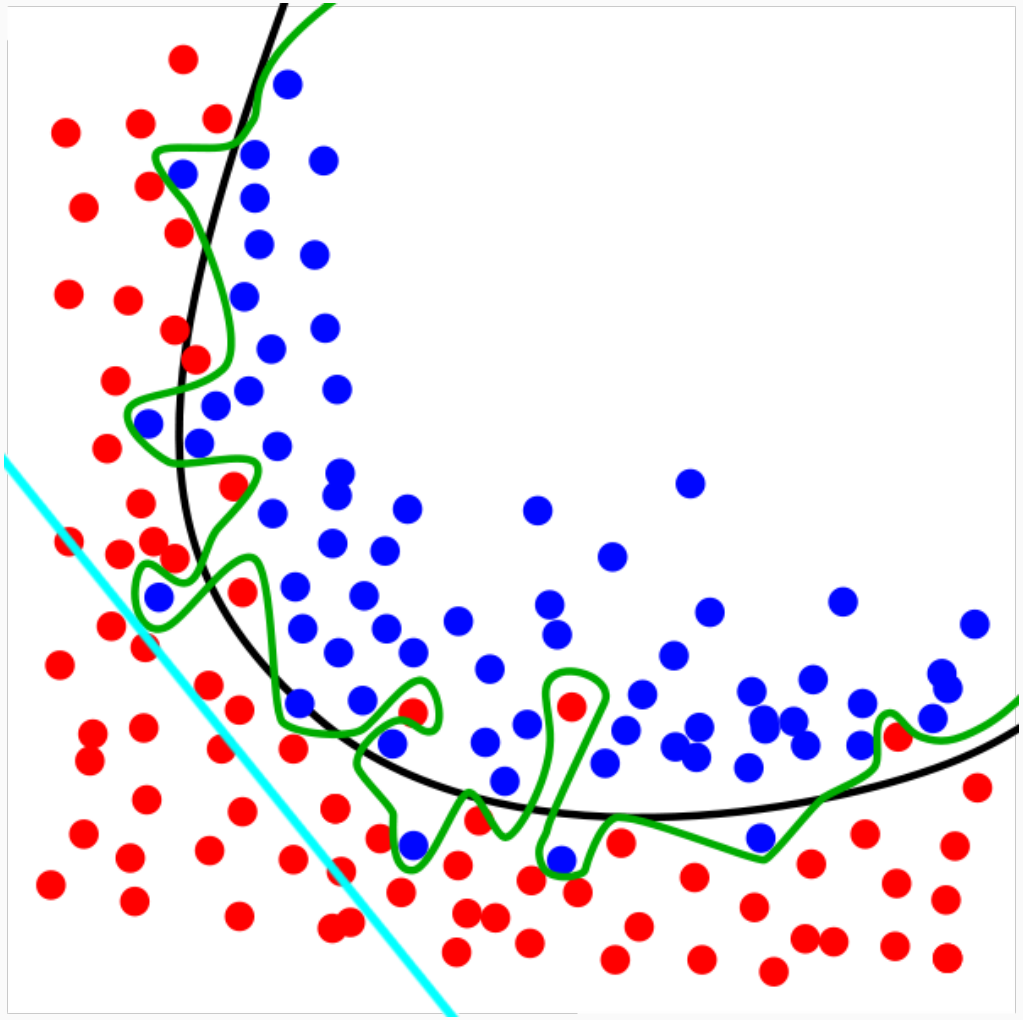


Figura 2: Hipóteses de classificadores ©

A Navalha de Occam

Figura 3: William de Occam ©

"ENTIDADES NÃO DEVEM SER MULTIPLICADAS ALÉM DO NECESSÁRIO"

Complexidade de Kolmogorov

```
110110110110110110110110110110
110110110110110110110110110110
110110110110110110110110110110
110110110110110110110110110110
110110110110110110110110110110
```

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

Complexidade de Kolmogorov

```
10101010100010110101011010101
10101010010010111101010100100
11101010001001101010010101010
11010010010101010101011010101
10101010010010111101000111001
1
```

```
print '
10101010100010110101011010101
10101010010010111101010100100
11101010001001101010010101010
11010010010101010101011010101
10101010010010111101000111001
1
'; exit!
```

O Princípio da Descrição mais Simples

Origem

- Rissanen [19] em 1978 definiu o Princípio da Descrição mais Simples (MDL);
- Formalizou a Navalha de Occam;
- Enraizado na Complexidade de Kolmogorov;
- Quanto mais temos conhecimento de um dado, mais podemos comprimi-lo.

Definições

- $\mathcal{X}$ é alfabeto finito de símbolos;
- $\mathcal{X}^n$ representa o conjunto de todas as sequências finitas de n símbolos;
- x^n abrevia $(x_1, x_2, \ldots, x_n)$;
- Uma *fonte* de probabilidade é uma sequência $P^{(1)}, P^{(2)}, \ldots$ definida em $\mathcal{X}^1, \mathcal{X}^2, \ldots$.

Procuramos por uma codificação *binária*, de um conjunto de dados, que consiga descrevê-lo de maneira única da menor forma possível

Codificações

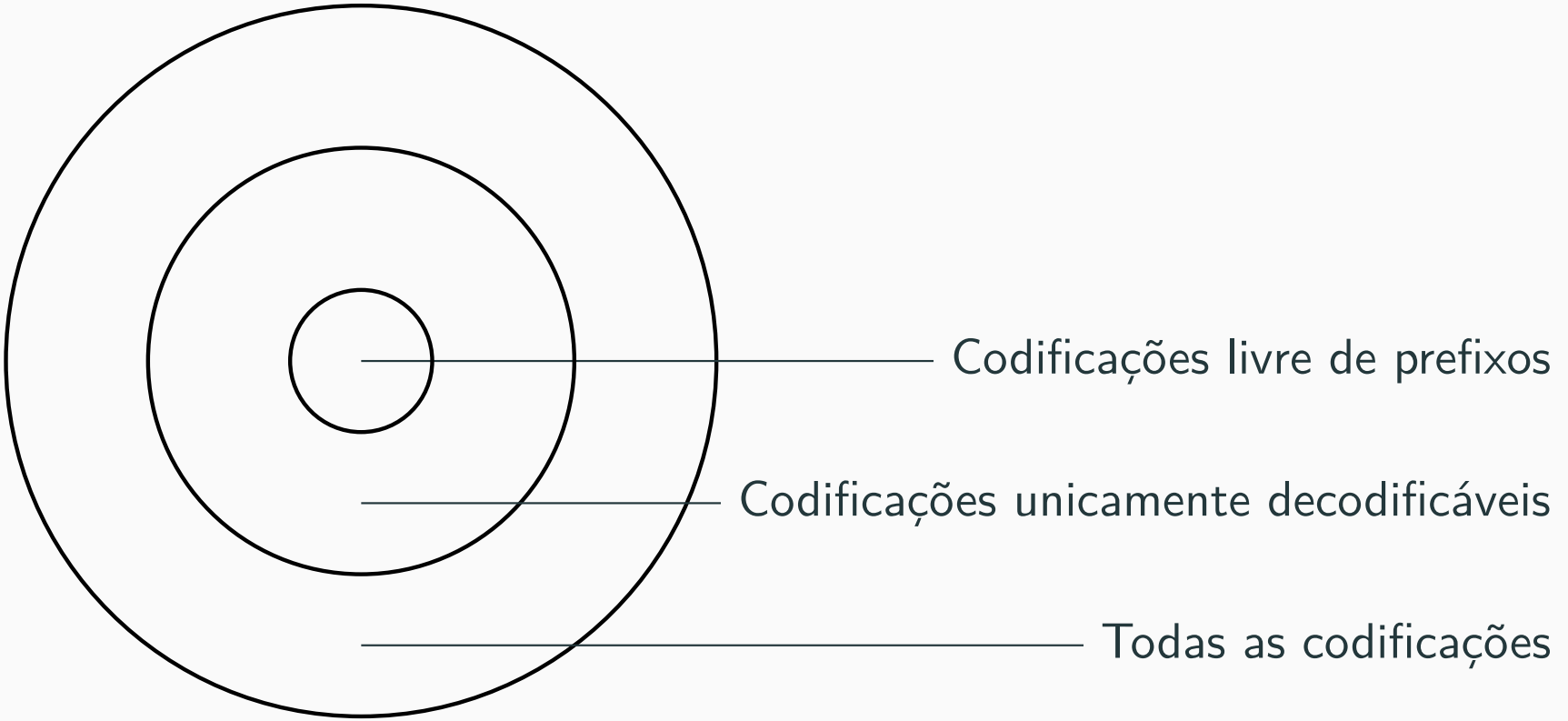


Figura 4: Relação entre diferentes classes de codificação

Inequalidade de Kraft

Teorema (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 $\sum_{i=1}^{n} D^{-l_i} \leq 1$.

Inequalidade de Kraft

Teorema (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 $\sum_{i=1}^{n} D^{-l_i} \leq 1$.

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

Codificações

- L_C representa o tamanho esperado dos dados codificados por C em uma f.d.p. P_C

Codificações

- L_C representa o tamanho esperado dos dados codificados por C em uma f.d.p. P_C
- $L_C := \sum_i x_i \cdot p(x_i)$

Codificações

- L_C representa o tamanho esperado dos dados codificados por C em uma f.d.p. P_C
- $L_C := \sum_i x_i \cdot p(x_i)$
- $E[X] = \sum_{i=1}^{\infty} x_i \cdot p_i$

Codificações

- L_C representa o tamanho esperado dos dados codificados por C em uma f.d.p. P_C
- $L_C := \sum_i x_i \cdot p(x_i)$
- $E[X] = \sum_{i=1}^{\infty} x_i \cdot p_i$

$$L := \arg\min_{L \in \mathcal{L}_Z} E_P[L(Z)] \tag{2}$$

Codificações

- L_C representa o tamanho esperado dos dados codificados por C em uma f.d.p. P_C
- $L_C := \sum_i x_i \cdot p(x_i)$
- $E[X] = \sum_{i=1}^{\infty} x_i \cdot p_i$

$$L := \underset{L \in \mathcal{L}_Z}{\arg\min}\, E_P[L(Z)] \tag{2}$$

Teorema (Lei dos grandes números)

Se X é uma variável aleatória, então

$$\lim_{|X| \to \infty} \frac{1}{|X|} \sum_{i=1}^{|X|} x_i = E[X]$$

Mas qual é o tamanho mínimo ideal para L?

Codificações

Teorema

Seja C uma codificação de um conjunto de símbolos S. 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 $-\log P(x_k)$.

É possível mostrar que o tamanho ótimo é limitado inferiormente por $-\log P(x)$ para um dado símbolo x.

Como sabemos qual P eleger?

Codificações

Proposição (Inequalidade da informação)

Se P e Q são distribuições de massa de probabilidade tal que $P \neq Q$, então $E_P[P(X)] < E_P[Q(X)]$.

Codificações

Proposição (Inequalidade da informação)

Se P e Q são distribuições de massa de probabilidade tal que $P \neq Q$, então $E_P[P(X)] < E_P[Q(X)]$.

- A partir da Inequalidade de Kraft podemos garantir a existência de uma codificação livre de prefixos;
- Pela Lei dos Grandes Números, quanto mais amostra obtivermos, mais próximo da distribuição ótima estaremos;
- Se P for a distribuição ótima, pela Inequalidade da informação, então $L_C(z) = \lceil -\log P(z) \rceil$.

MDL em duas partes

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

MDL em duas partes

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

MDL Refinado

É possível utilizar-se de duas partes, mas codificações universais são computacionalmente proibitivas

Aplicações do MDL em Aprendizado de Máquina

Árvores de Decisão

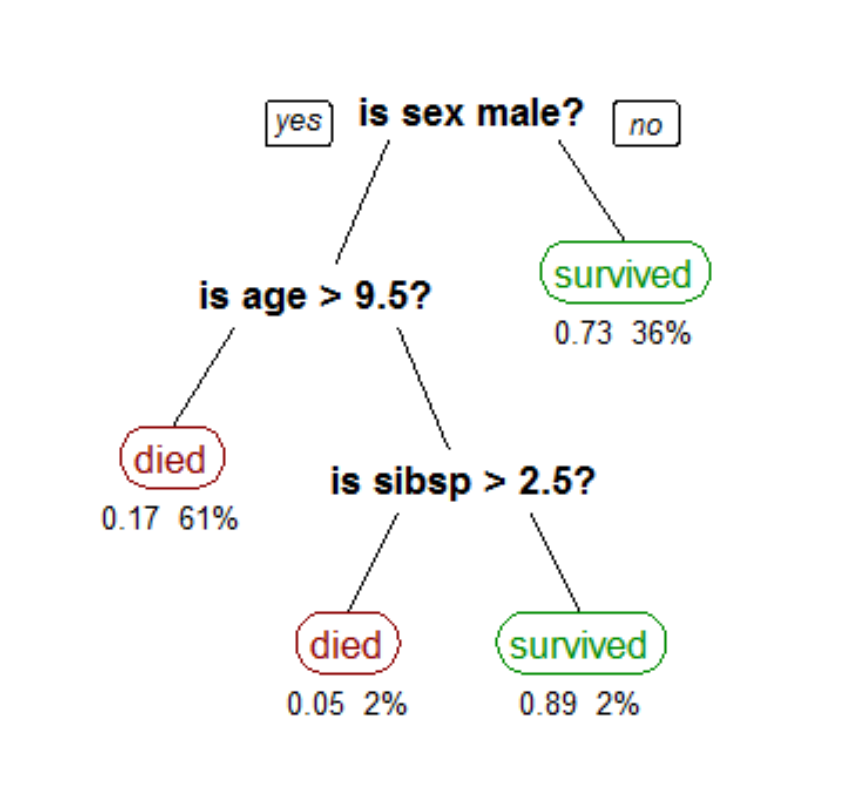


Figura 5: Árvore de Decisão para classificar sobreviventes do *Titanic* ©

- Quinlan & Rivest [18]: Geração de árvores de decisão, com construção *bottom-up*, substituindo nós folha com nós de decisão que não aumentem o custo de transmissão da informação;
- Kononenko [14]: MDL em duas partes para poda de árvores de decisão, removendo um nó de decisão baseando-se na entropia.

Redes Bayesianas

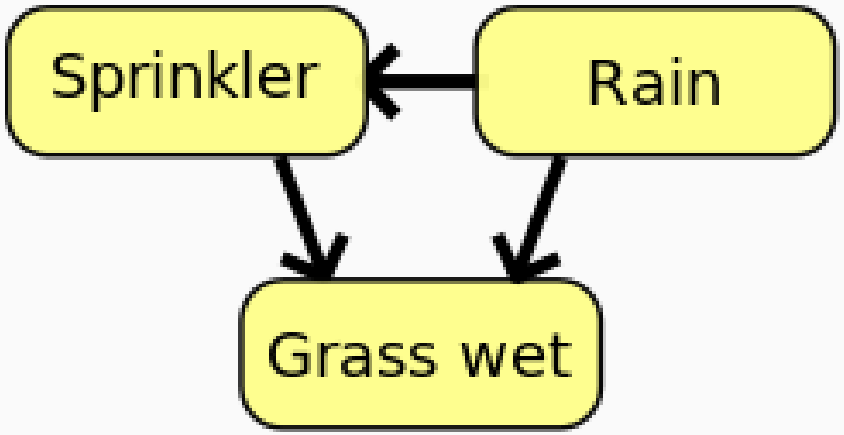


Figura 6: Rede Bayesiana simples ©

- Lam & Bacchus [16]: Treinamento de redes Bayesianas para decisão sob incertezas
- Friedman *et al.* [13]: MDL para o treinamento de redes Bayesianas para classificação.

Spams

Figura 7: Enlatado *Spam* ©

- Bratko *et al.* [7]: Compressão dinâmica de Markov com MDL, e outro combinando entropia mínima cruzada;
- Braga & Ladeira [6]: MDL e codificação de Huffmann foi abordada;
- Almeida [4] e Almeida & Yamakami [1, 2, 3]: MDL e fatores de confidência, chamado de *MDL-CF*.
- Silva *et al.* [22, 23]: Extensão do método de Almeida [4] para que trabalhasse com dados textuais quaisquer e deram um primeiro passo na tentativa de categorizar atributos contínuos.

Motivação & Objetivos

Motivação

- A extensão do método por Silva *et al.* [22, 23] foi benéfica para textos, mas não tão boa para dados contínuos;
- O método perdeu sua propriedade de treinamento incremental;

Motivação

- A extensão do método por Silva *et al.* [22, 23] foi benéfica para textos, mas não tão boa para dados contínuos;
- O método perdeu sua propriedade de treinamento incremental;
- Foi a discretização a raiz do problema?

Perguntas 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 [4] e Silva *et al.* [22, 23] para que seja capaz de processar amostras representadas também por atributos contínuos?

Estimativa de f.d.p.

Estimativa de densidade de *kernel*

$$p(X) := \sum_{i=1}^{G} w_i \cdot \mathcal{N}(X \mid \mu_i, \Sigma_i), \tag{3}$$

Estimativa de densidade de *kernel*

$$p(X) := \sum_{i=1}^{G} w_i \cdot \mathcal{N}(X \mid \mu_i, \Sigma_i), \tag{3}$$

- Um método clássico para estimar f.d.p. é assumir uma distribuição Gaussiana;
- Estimar parâmetros não é trivial [17, 27];
- Silverman [25] cunhou a *estimativa de densidade de Kernel* (EDK).

Estimativa de densidade de *kernel*

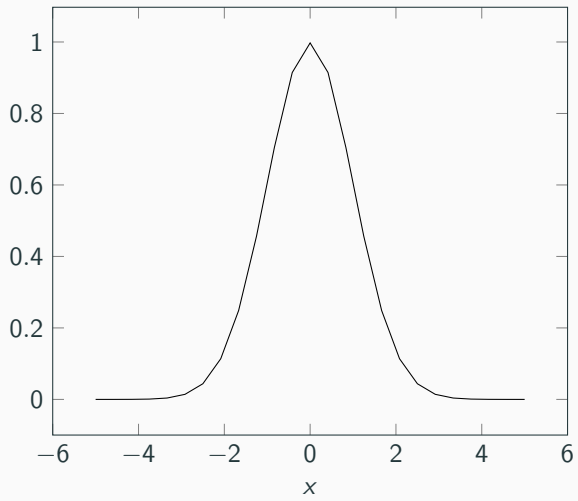


Figura 8: *Kernel* Gaussiano

- O *kernel* Gaussiano é uma escolha clássica;
- O EDK evita a necessidade de discretização;
- O número de componentes para distribuição cresce linearmente para cada amostra.

Estimativa de densidade de *kernel* online

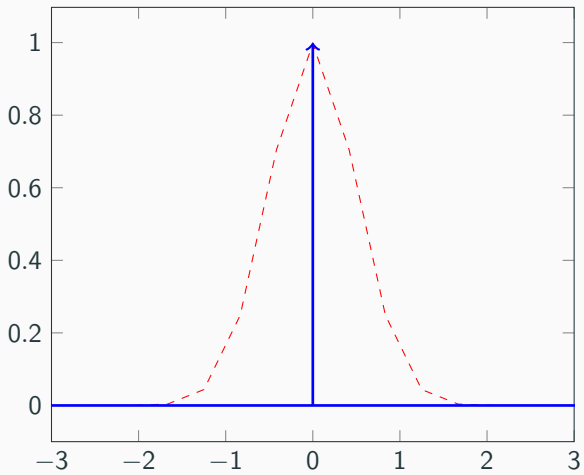


Figura 9: Gráficos de uma função Delta-Dirac (azul sólido) de $\mu = 0$ e uma distribuição Gaussiana (vermelho cerrilhado) de $\sigma^2 = \frac{1}{2}$ e $\mu = 0$.

- Kristan *et al.* [15] propuseram um EDK online (oKDE);
- Utiliza-se de uma distribuição amostral (funções Delta-Dirac);

Estimativa de densidade de *kernel* online

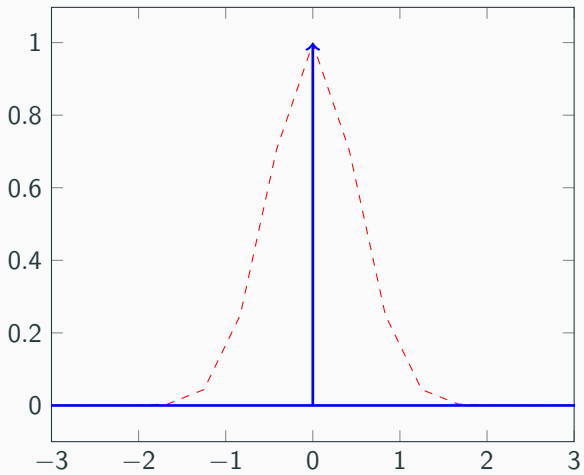


Figura 9: Gráficos de uma função Delta-Dirac (azul sólido) de $\mu = 0$ e uma distribuição Gaussiana (vermelho cerrilhado) de $\sigma^2 = \frac{1}{2}$ e $\mu = 0$.

- Kristan *et al.* [15] propuseram um EDK online (oKDE);
- Utiliza-se de uma distribuição amostral (funções Delta-Dirac);
- Como manter uma distribuição por ponto é inviável, há uma compressão de amostras;
- Para computar a largura de banda, utiliza-se a divergência de Kullback-Leibler;
- Ferreira *et al.* [12] cunharam o *xokde++*, uma versão melhorada do oKDE.

Construindo um método de classificação baseado no MDL

Predição de amostras

$$\hat{L}(\vec{x}|c) := \sum_{i=1}^{n} \lceil -\log p_{(i,c)}(\vec{x}_i) \rceil \tag{4}$$

Predição de amostras

$$\hat{L}(\vec{x}|c) := \sum_{i=1}^{n} \lceil -\log p_{(i,c)}(\vec{x}_i) \rceil \tag{4}$$

onde

$$p_{(i,c)} := \begin{cases} 2^{-\Omega} & p'_{(i,c)} \to \infty \ \vee \ p'_{(i,c)} = 0 \\ p'_{(i,c)} & c.c. \end{cases} \tag{5}$$

Como definir a complexidade do modelo?

Complexidade do modelo

- p' é um distribuição Gaussiana;
- Quanto mais componentes na distribuição, menos conhecimento temos da distribuição;

Complexidade do modelo

- p' é um distribuição Gaussiana;
- Quanto mais componentes na distribuição, menos conhecimento temos da distribuição;
- Logo, $|p'_{(\cdot,c)}| = G_c$ é uma boa escolha para a complexidade do modelo.

Primeiro versão do método de classificação

Podemos obter, então, a primeira versão do Aprendiz de Descritores de Mistura Gaussiana:

$$GMDL'(\vec{x}, K) := \arg\min_{c \in K}[\hat{L}(\vec{x}|c) + G_c] \tag{6}$$

Primeiro versão do método de classificação

Podemos obter, então, a primeira versão do Aprendiz de Descritores de Mistura Gaussiana:

$$GMDL'(\vec{x}, K) := \arg\min_{c \in K}[\hat{L}(\vec{x}|c) + G_c] \tag{6}$$

e normalizando:

$$GMDL(\vec{x}) := \begin{bmatrix} GMDL'(\vec{x}, \{c_1\}) \\ GMDL'(\vec{x}, \{c_2\}) \\ \vdots \\ GMDL'(\vec{x}, \{c_k\}) \end{bmatrix} \Big/ \sum_{k=1}^{|K|} GMDL'(\vec{x}, \{c_k\}) \tag{7}$$

Suavição de funções Delta-Dirac

- Uma distribuição é degenerada se tem baixa variância;
- O oKDE é suscetível a tais distribuições dependendo da ordem em que as amostras eram apresentadas;

Suavição de funções Delta-Dirac

- Uma distribuição é degenerada se tem baixa variância;
- O oKDE é suscetível a tais distribuições dependendo da ordem em que as amostras eram apresentadas;
- O GMDL mitiga esse problema adicionando um ruído Gaussiano de média zero e desvio padrão $\tilde{\sigma}^2$;
- Computa-se a variância populacional incrementalmente com o método proposto por Welford [26].

Podemos tornar o GMDL mais flexível?

Ponderação de Atributos

- Seleção de atributos é uma técnica clássica da área de aprendizado de máquina;
- Engenharia de atributos busca sanar problemas como a maldição da dimensionalidade e a alta variância em um conjunto;
- Muitos trabalhos na área: PCA, Ganho de Informação, Informação Mútua, etc.;
- Assim como o Naïve Bayes, o GMDL assume que as distribuições são independentes.

Ponderação de Atributos

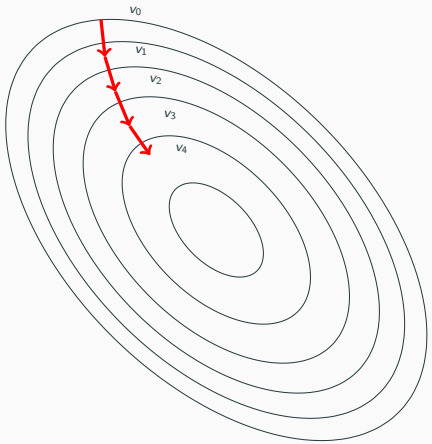


Figura 10: Ilustração do funcionamento do Gradiente Descendente em um conjunto de pontos v.

- Para aliviar essa suposição, adicionamos um expoente a cada atributo, para limitar sua influência na predição;
- Além disso, tais atributos são atualizados utilizando-se SGD;
- A atualização é dada por:

$$\nabla J_{\theta_i} = -\sum_{j=1}^{m}\sum_{k=1}^{|K|}(\delta_{y^{(j)}c_k} - L(\vec{x}^{(j)}|c_k)) \cdot \varphi_{ik}^{\theta_i} \cdot \ln\theta_i \cdot (1 - L(\vec{x}^{(j)}|c_k)) \quad (8)$$

Separação de margem

- A margem de uma barreira de decisão é a distância da barreira até os dados separados;
- O SVM é amplamente conhecido por procurar por uma função que otimiza a separação da margem;
- Silva *et al.* [24] utilizaram-se também de um fator para separar amostras do que chamaram de protótipos de classe.

Separação de margem

Utiliza-se a distância de Mahalanobis com o valor esperado das médias e a variância obtida a partir da médias:

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

Separação de margem

Utiliza-se a distância de Mahalanobis com o valor esperado das médias e a variância obtida a partir da médias:

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

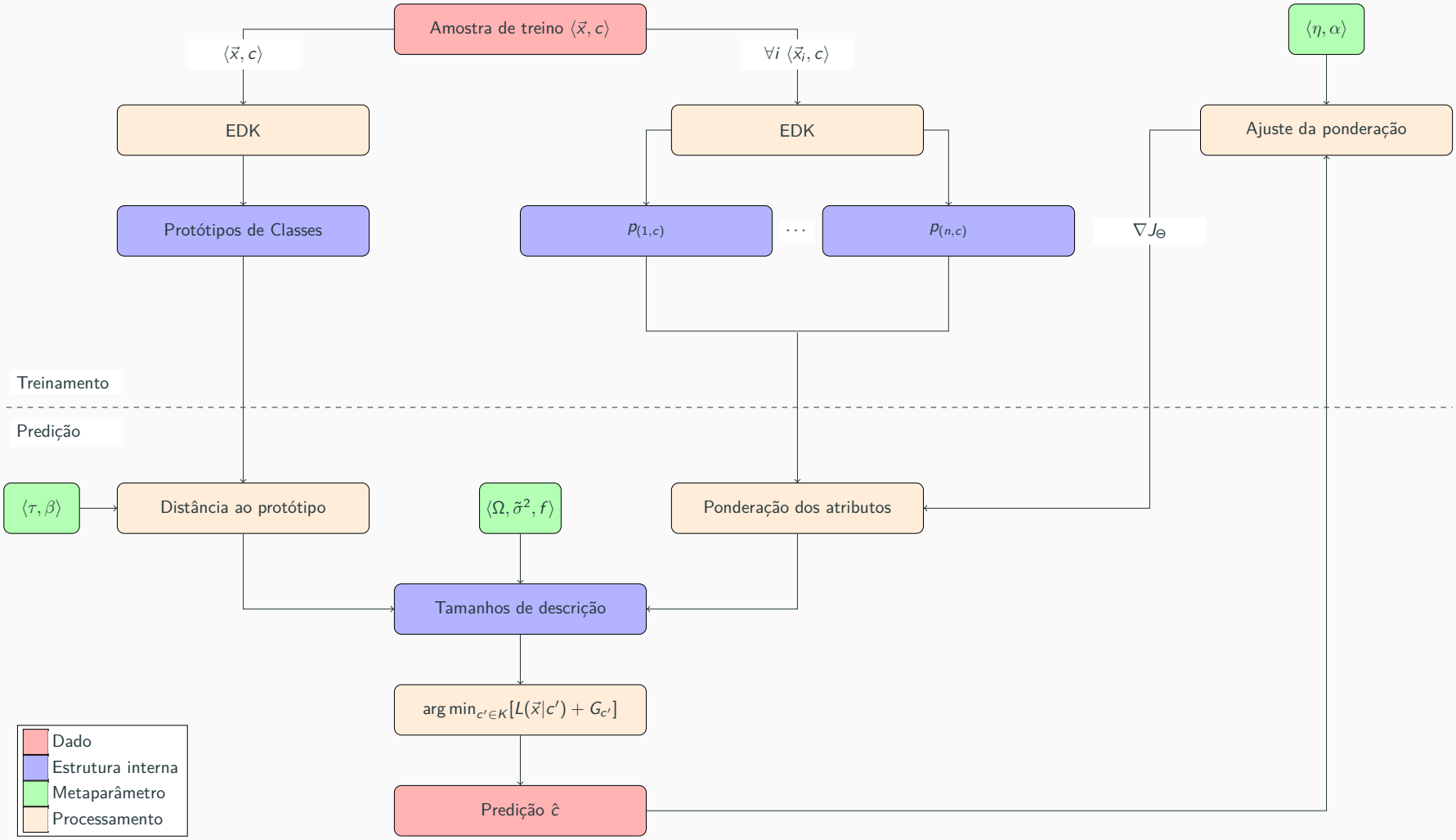
Defini-se então o parâmetro de penalidade, com zero, para a classe menos similar:

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

GMDL

Equação do GMDL

$$\hat{L}(\vec{x}|c) := \hat{S}(\vec{x}, \tilde{c})^{\tau} \cdot \sum_{i=1}^{n} \lceil -\log p_{(i,c)}(\vec{x}_i) \rceil^{\theta_i} \qquad (9)$$


Amostra de treino $\langle \vec{x}, c \rangle$
$\langle \vec{x}, c \rangle$
$\forall i \ \langle \vec{x}_i, c \rangle$
$\langle \eta, \alpha \rangle$
EDK
EDK
Ajuste da ponderação
Protótipos de Classes
$p_{(1,c)}$
$\cdots$
$p_{(n,c)}$
∇J_Θ
Treinamento
Predição
$\langle \tau, \beta \rangle$
Distância ao protótipo
$\langle \Omega, \tilde{\sigma}^2, f \rangle$
Ponderação dos atributos
Tamanhos de descrição
$\arg\min_{c' \in K}[L(\vec{x}|c') + G_{c'}]$
Predição $\hat{c}$
Dado
Estrutura interna
Metaparâmetro
Processamento

Complexidade do GMDL

- Complexidade no treinamento é $O(\frac{G^2-G}{2+G})$ (G costuma ser baixo [15]);
- Complexidade na predição é $O(n \cdot |K|)$;

Complexidade do GMDL

- Complexidade no treinamento é $O(\frac{G^2-G}{2+G})$ (G costuma ser baixo [15]);
- Complexidade na predição é $O(n \cdot |K|)$;
- Adiciona-se $O(n^{2\cdot(1-\delta_{0\tau})})$ quando utiliza-se a distância ao protótipo.

Avaliação experimental

Cenários

- Dois grandes cenários: Offline e Incremental.

Cenários

- Dois grandes cenários: Offline e Incremental.
- **Offline**:
  - Amostras apresentadas em batelada;
  - Não há correção, apenas classificação.
- **Incremental**:
  - **Correção imediata**: retroalimentação imediata;
  - **Correção limitada**: retroalimentação com uma probabilidade de 50%;
  - **Correção atrasada**: retroalimentação em tempo que cresce exponencialmente em 1,1, arredondado para baixo.

Bases de dados

		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

Cenário Offline

Avaliação

Os métodos avaliados foram:

1. Naïve Bayes gaussiano (GNB) [11];
2. Florestas aleatórias (RF) [8];
3. Máquinas de vetores de suporte (SVM) [9];
4. k-vizinhos mais próximos (kNN) [21].

Avaliação

Os métodos avaliados foram:

1. Naïve Bayes gaussiano (GNB) [11];
2. Florestas aleatórias (RF) [8];
3. Máquinas de vetores de suporte (SVM) [9];
4. *k*-vizinhos mais próximos (kNN) [21].

Utilizou-se os seguintes parâmetros:

1. **GMDL**
   - 1.1 $\tilde{\sigma}^2$: 2; 5; 10;
   - 1.2 τ: 0; 5; 10.
2. **RF**
   - 2.1 `n_estimators`: 10; 20; 30; 40; 50; 60; 70; 80; 90; 100;
   - 2.2 `criterion`: "*gini*"; "*entropy*".
3. **SVM**
   - 3.1 `C`: 0,0001; 0,001; 0,01; 0,1; 1; 10; 100; 1000.
4. **kNN**
   - 4.1 `n_neighbors`: 3; 5; 7; 9; 11; 13; 15; 17; 19.

Resultados

	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

Considerações

- RF obteve o melhor desempenho na maioria dos experimentos;
- Demais métodos também mostram bom desempenho em algumas bases;
- RF, SVM e kNN não terminaram a computação em 48 horas – Teorema dos Grandes Números na prática no GMDL;
- GMDL foi mal em bases de alta cardinalidade (*hill-valley* e *libras*);
- Não houve evidência estatística de diferença entre os métodos.

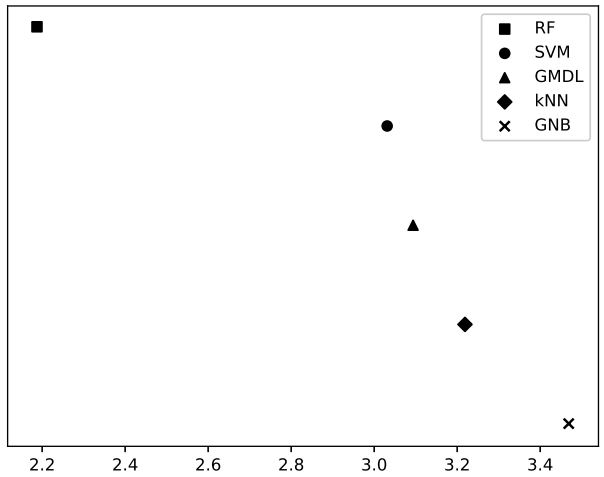


Figura 11: Ranqueamento médio de cada método avaliado no cenário *offline*.

Cenário Incremental

Avaliação

Três cenários ($\times 2$, considerando normalização):

1. *Correção imediata*
2. *Correção limitada*
3. *Correção atrasada*

Os métodos avaliados foram:

1. Perceptron multinível (MLP) [20];
2. Passivo-Agressivo (PA) [10];
3. Gradiente Descendente Estocástico (SGD) [5].

Avaliação

Com parâmetros padrões para os métodos, e o GMDL com:

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

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

Correção Imediata – Considerações

- GMDL obteve o melhor resultado na maioria das bases de dados – nenhuma menor F-medida;
- MLP também mostrou bom desempenho em algumas bases como *miniboone* e *letter*;
- MLP e o SGD tiveram desempenho mais baixo em bases com menos amostras e desbalanceadas;
- Wilcoxon mostrou que o GMDL foi superior ao SGD e ao PA, em uma análise par-a-par.

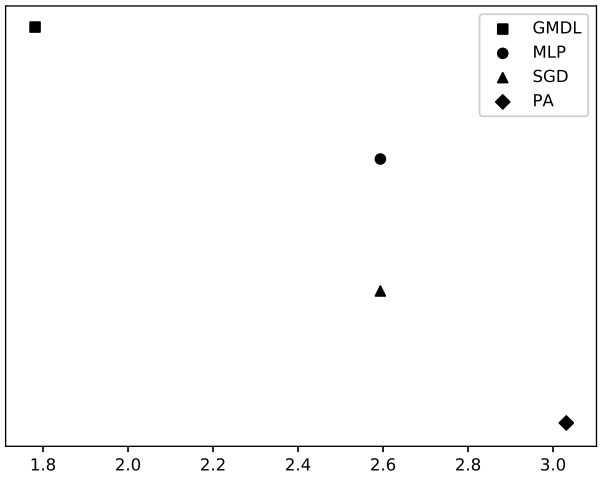


Figura 12: Ranqueamento médio de cada método avaliado no cenário de correção imediata.

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

Correção Limitada – Considerações

- O GMDL só obteve o pior desempenho na base *skin* – como o SGD;
- MLP teve o melhor resultado na *skin*, o que demonstra eficiência em baixa dimensionalidade;
- Ambos o GMDL e o PA atingiram predição perfeita na base *adult*;
- Wilcoxon mostrou que o GMDL foi superior ao SGD e ao PA, em uma análise par-a-par.

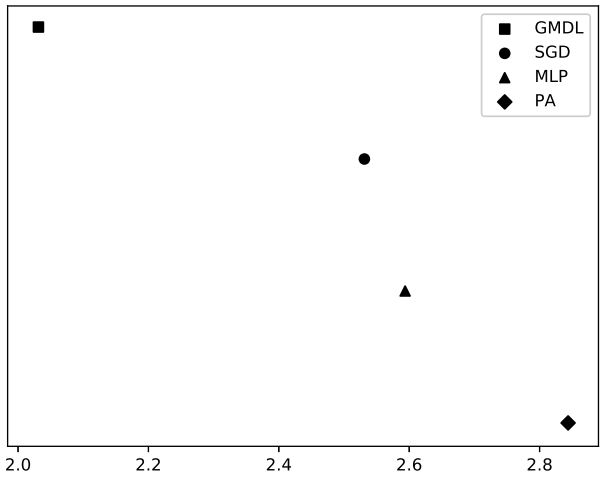


Figura 13: Ranqueamento médio de cada método avaliado no cenário de correção limitada.

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**

Correção Atrasada – Considerações

- Todos os métodos avaliados tiveram uma distribuição quase uniforme de melhor desempenho pelas bases;
- O GMDL e o PA obtiveram predição perfeita na base *adult*;
- A maioria dos piores desempenhos foram obtidos pelo MLP;
- O GMDL, na base *hill-valley*, foi consistentemente melhor que os demais métodos com as amostras não normalizadas;
- Wilcoxon mostrou que o GMDL foi superior ao MLP e ao PA, em uma análise par-a-par, no cenário não normalizado.

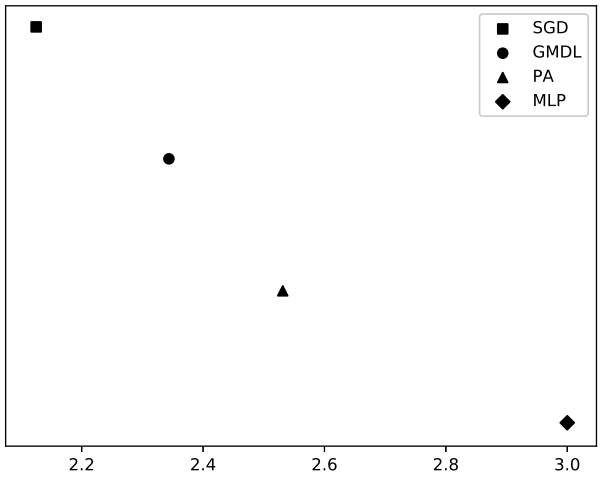


Figura 14: Ranqueamento médio de cada método avaliado no cenário de correção atrasada.

Conclusão

Conclusão

- O problema de se determinar o modelo para uma tarefa de classificação é muito estudado, mas não existe um guia direto de como se fazer;
- O GMDL é multiclasse, multinomial, naturalmente incremental e robusto ao sobreajuste e normalização dos dados;
- O GMDL baseia-se no MDL, uma formalização da ideia da Navalha de Occam: provendo ao método uma troca benéfica entre acurácia e sobrejuste dos dados;

Conclusão

- O GMDL é equivalente aos métodos GNB, RF, SVM, kNN no cenário offline;
- O GMDL se mostrou superior aos métodos SGD e PA no cenários de correção imediata e limitada;
- O GMDL é robusto à normalização e obteve resultados superiores ou equiparáveis aos métodos comparados no cenário incremental atrasado;
- Dada a versatilidade e robustez, o GMDL é um candidato a grandes problemas do mundo real.

Trabalhos futuros e melhorias

- Cálculo do tamanho da descrição;
- Demora na computação da distância ao protótipo;
- O tamanho de descrição da classe impactou negativamente no desempenho do GMDL em alguns casos;
- Utilizar o GMDL em outros problemas.

Obrigado!

Referências

[1] Almeida, T. A. & Yamakami, A. (2012a). Advances in spam filtering techniques. *Computational Intelligence for Privacy and Security*, **394**(2012), 199–214.

[2] Almeida, T. A. & Yamakami, A. (2012b). Facing the spammers: A very effective approach to avoid junk e-mails. *Expert Systems with Applications*, **39**(7), 6557–6561.

[3] Almeida, T. A. & Yamakami, A. (2016). Compression-based spam filter. *Security and Communication Networks*, **9**(4), 327–335.

[4] Almeida, T. A. D. (2010). *SPAM: do Surgimento à Extinção*. Ph.d. thesis, State University of Campinas.

[5] Bottou, L. (2010). Large-Scale Machine Learning with Stochastic Gradient Descent. In *19th International Conference on Computational Statistics (COMPSTAT'10)*, 177–186, Paris, França. Springer.

[6] Braga, I. A. & 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)*, 11–20, Belém, Brasil.

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

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

[9] Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine learning*, **20**(3), 273–297.

[10] Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, **7**(Mar), 551—-585.

[11] Duda, R. O. & Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. Wiley, 1 edition.

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

[13] Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning*, **29**(2), 131–163.

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

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

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

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

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

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

[20] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in .... *Psychological Review*, **65**(6), 386–408.

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

[22] Silva, R. M., Almeida, T. A., & Yamakami, A. (2016a). Detecção Automática de SPIM e SMS Spam usando Método baseado no Princípio da Descrição mais Simples. Technical report, Universidade Estadual de Campinas, Campinas, Brasil.

[23] Silva, R. M., Almeida, T. A., & Yamakami, A. (2016b). 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)*, 470–475, Anaheim, CA, EUA. IEEE.

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

[25] Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*, volume 26. Springer, Boston, MA, 1 edition.

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

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