

Detecção de novidades aplicada à estimação de confiabilidade de classificação de sinais de sonar passivo

Geancarlo G. M. da Rocha †
geancarlo.rocha@gmail.com

†Programa de Pós-Graduação em Engenharia
Elétrica (PPEEL)
Centro Federal de Educação Tecnológica Celso
Suckow da Fonseca (CEFET/RJ)
Av. Maracanã 229, Bloco E, Sala 205

João Baptista de O. de Souza e Filho ‡ †
jbfilho@poli.ufrj.br

‡Departamento de Eletrônica e Computação (DEL)
Escola Politécnica (POLI)
Universidade Federal do Rio de Janeiro (UFRJ)
Av. Horácio Macedo 1030, CT, Sala H-217

Resumo—Técnicas de aprendizagem de máquina são atrativas para a tarefa de classificação de contatos a partir de sinais de sonar passivo. A obtenção de bases de dados completas, porém, é uma tarefa difícil, senão impossível, tendo em vista a natureza sensível de tal aplicação. Através de métodos desenvolvidos na sub-área de detecção de novidades, buscamos viabilizar a geração de coeficientes indicadores da confiabilidade de sistemas de classificação automática de contatos. Esta etapa, realizada após a classificação, tem como objetivo evitar que dados atípicos possam ser classificados de forma similar àqueles típicos por sistemas discriminativos. Avaliamos o uso de tais técnicas através de sinais obtidos em ensaios controlados realizados em raia acústica e que envolveram 8 classes distintas de embarcações. Os resultados exploratórios avaliando a capacidade de diferentes modelos em distinguir sinais de classes conhecidas e de classes desconhecidas são apresentados, assim como são realizadas comparações quanto à complexidade computacional dos algoritmos. Verificou-se que para todas as 8 classes de dados, é possível obter detectores com, em média, um valor de área sob a curva ROC superior à 0,95.

Keywords—*Sonar passivo, detecção de novidades, análise de confiabilidade, sistemas de suporte à decisão.*

I. INTRODUÇÃO

A utilização de técnicas de aprendizagem de máquina para a classificação de contatos em sistemas de sonar passivo é de grande importância prática. Ao automatizar esta tarefa, é possível reduzir a dependência do sistema quanto ao operador de sonar e, portanto, reduzir possíveis erros humanos. Infelizmente, não é concebível que haja disponibilidade de dados contemplando todas as classes de embarcações para o desenvolvimento de sistemas de classificação automática de contatos. Além disso, é possível que a característica da embarcação que leva à definição de sua classe não seja identificável através dos sinais de sonar disponíveis. Claramente, o ideal é que, além de fornecer uma classificação deste sinal, uma estimativa da confiabilidade daquela predição seja fornecida ao operador de sonar, que pode dedicar um tratamento diferenciado

caso o sistema indique um alto nível de incerteza a ela associada.

Uma forma de se determinar a confiança de uma predição é determinar a distribuição de probabilidade associada a geração dos sinais de sonar para as classes conhecidas. As predições de sinais improváveis, ou mais precisamente, aqueles que ocorrem em regiões de baixa densidade de probabilidade, devem ser identificadas como de baixa confiança. Classificadores generativos buscam estimar diretamente estas distribuições de probabilidade para cada classe, porém esta tarefa é mais difícil do que se determinar um classificador discriminativo que possa simplesmente separar as classes. As evidências para tal questão são discutidas em [1], onde foi verificado que o erro assintótico, isto é, o erro atingido quando se tem uma quantidade de dados para treino tendendo ao infinito, para o classificador discriminativo, no caso uma regressão logística, mostrou-se inferior ao erro de sua contraparte generativa, no caso um classificador bayesiano gaussiano ingênuo. Outro fator contra o uso de classificadores generativos é a dificuldade quanto à modelagem de sinais de dimensão elevada, presentes neste problema por conta da discriminação baseada em janelas de espectros do sinal, como será discutindo adiante.

A fim de conciliar tais requisitos, é possível dividir o sistema de classificação de contatos em duas etapas. Primeiramente, utiliza-se um classificador multiclasse discriminativo de alta-capacidade, tal como uma rede neural, para realizar as predições quanto às classes mais prováveis, de forma similar a [2]. Em seguida, determina-se a confiabilidade da predição através de alguma técnica que busque inferir o quão semelhante o sinal de sonar analisado é em relação à classe escolhida. Esta segunda etapa é conhecida na literatura como detecção de novidades [3], sendo o foco deste artigo.

A estrutura do artigo é detalhada a seguir. A Seção II apresenta a base de dados utilizada, contextualizado o problema. Na Seção III são apresentadas os modelos/técnicas que serão avaliados como detectores de novidade. A Seção IV apresenta considerações metodológicas do ponto de

vista prático e que surgem no contexto da aprendizagem. Em seguida, a Seção V levanta considerações relacionadas a complexidade temporal e computacional dos algoritmos explorados. As seções VI e VII apresentam os experimentos realizados, assim como os resultados e as conclusões.

II. BASE DE DADOS

A base de dados utilizada para as simulações foi fornecida pelo Instituto de Pesquisas da Marinha (IPqM). Ela é constituída pelo ruído irradiado por 25 navios, pertencentes a 8 classes diferentes de embarcações, durante 263 corridas realizadas na raia acústica da Marinha do Brasil, localizada em Arraial do Cabo. Durante cada corrida, uma embarcação percorreu a raia, e suas condições operacionais foram mantidas constantes. O ruído emitido pelas embarcações foi captado por um hidrofone localizado ao fundo do assoalho submarino da raia acústica, e amostrado a uma taxa de 22050Hz utilizando 16 bits de resolução.

O sinal digital foi processado de acordo com a cadeia de processamento proposta em [2]. Basicamente, os estágios envolvidos são: (1) formam-se blocos de 4096 amostras (sem sobreposição), que são multiplicados por janelas de Hanning; (2) calcula-se o valor do módulo da transformada de Fourier de cada bloco; (3) utiliza-se o algoritmo *two-pass split window* (TPSW) [4] para a estimação do ruído de fundo no espectro obtido anteriormente; (4) normaliza-se cada ponto deste espectro dividindo-o pela estimativa do ruído de fundo correspondente; (5) tomam-se os primeiros 557 pontos do espectro, o que corresponde a uma faixa de 0 a $\approx 3kHz$; (6) normaliza-se esta faixa de frequência para que apresente energia unitária.

Como resultado desta cadeia, cada amostra apresentada ao sistema de classificação consiste em um vetor de 557 dimensões, contendo o módulo do espectro na faixa de frequência diretamente associada ao ruído produzido pelo maquinário em operação nas embarcações [2]. O total de amostras resultantes é de 2432, 3432, 4797, 3072, 7075, 2934, 2143 e 3392, para as classes de embarcações A, B, C, D, E, F, G e H, respectivamente.

III. DETECÇÃO DE NOVIDADES

Em [3] foi conduzida uma revisão sistemática de métodos para a detecção de novidades. Segundo este trabalho, um modelo de normalidade $M(\theta)$, com parâmetros livres θ , é inferido utilizando dados (de treino) da classe normal, o qual deve ser utilizado para se atribuir um valor de escore/índice $z(\mathbf{x}_*)$ a dados desconhecidos \mathbf{x}_* . Assim, valores de escore alto indicam um maior distanciamento relativo ao modelo de normalidade $M(\theta)$, portanto uma maior chance de anormalidade. Note que é possível definir um limiar τ , de forma que: se $z(\mathbf{x}_*) > \tau$, o dado desconhecido é apontado como anormal/novidade. Claramente, a variação deste limiar fornece uma curva similar a curva ROC (*receiver operating characteristic*) [5], relacionando taxas de detecção de dados conhecidos e desconhecidos para diferentes limiares de decisão. Segundo esta proposta, diferentes técnicas podem ser utilizadas para se avaliar a similaridade dos dados às classes conhecidas. A seguir, aquelas selecionadas para esta aplicação, serão discutidas sucintamente.

A. PCA

A análise de componentes principais (PCA) é uma técnica clássica para a redução da dimensionalidade de dados [6]. Nela, tenta-se encontrar uma aproximação dos dados do conjunto de treino $\mathbf{X} \in \mathbf{R}^{N \times D}$, após a remoção de sua média, através de uma expansão linear dos dados baseada em L bases ortogonais $\mathbf{W} \in \mathbf{R}^{L \times D}$, as quais minimizam a função objetivo $J(\mathbf{W}, \mathbf{Z}) = \|\tilde{\mathbf{X}} - \mathbf{Z}\mathbf{W}\|_F^2$, onde $\tilde{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$ e $\tilde{\mathbf{x}}_i = \mathbf{x}_i - \tilde{\mathbf{x}}$. A solução ótima para este problema é dada por $\hat{\mathbf{W}} = \mathbf{V}_L$, onde \mathbf{V}_L é definido pelos L ($L < D$) autovetores associados aos maiores autovalores resultantes da decomposição em autovalores da matriz de covariância empírica dos dados, e $\hat{\mathbf{z}}_i = \tilde{\mathbf{x}}_i \hat{\mathbf{W}}^T$. Naturalmente, um aumento do número de bases (L) possibilita uma melhor aproximação dos dados.

Quanto à detecção de novidades, espera-se que o valor da função objetivo $J(\mathbf{W}, \mathbf{Z})$, supondo \mathbf{W} obtido de forma ótima para as classes conhecidas, seja superior quando avaliado para as classes desconhecidas. Assim, tanto maior o valor do erro, maior é a evidência de que o dado desconhecido seja uma novidade.

B. k-NN

A seguinte metodologia, baseada na determinação dos “k-vizinhos mais próximos” (*k-nearest neighbours*) [7], foi considerada. Para um vetor de dados sob avaliação, a mediana de sua distância euclidiana para os k-vizinhos mais próximos é utilizada como medida de similaridade às classes conhecidas. O uso da mediana, ao invés da média, visa amenizar a influência de *outliers*. Claramente, quanto menor for o valor do índice, mais similares são os dados às classes conhecidas.

Uma dificuldade deste método é que sua utilização de forma “ingênua” exige que a distância para todos os dados de treino seja determinada a cada evento de teste, o que possui um custo computacional apreciável para este problema. A redução do custo computacional em métodos de busca de vizinhos é uma área de pesquisa ativa [7], porém, por força de simplicidade, a investigação se limita ao método de busca bruta.

C. Curvas principais

As curvas principais representam uma generalização da análise de componentes principais. Ao invés de aproximar um conjunto de dados por uma reta passando pela direção principal do conjunto de dados, utiliza-se uma curva suave e auto-consistente, isto é, na qual cada ponto coincide com o valor esperado dos pontos que nela projetam [8]. Uma definição probabilística das curvas principais, onde esta técnica é associada ao processo generativo dos dados é dada em [9]. Informalmente, pode-se entender que uma curva deste tipo “passa no meio da nuvem de dados”.

Para a determinação das curvas, foi utilizado o algoritmo apresentado em [10], que aproxima a curva principal através de $k = 2c - 1$ segmentos de reta interconectados. Neste tipo de modelo, a distância euclidiana entre um vetor de dados e uma curva é utilizada como índice de similaridade do vetor a curva representativa de cada classe.

D. Misturas de Gaussianas

Nesta proposta, a distribuição dos dados é aproximada por uma mistura de K gaussianas bases [6], cada uma possuindo uma média $\boldsymbol{\mu}_k$, matriz de covariância $\boldsymbol{\Sigma}_k$ e coeficiente de mistura π_k específicos. A densidade de probabilidade para este modelo é definida como $p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. De posse desta expressão geral, podem ser estabelecidos diferentes casos especiais. Por exemplo, para $K = 1$, temos os dados modelados como provenientes de uma distribuição gaussiana multivariada. Neste caso, podem ainda ser consideradas diferentes estruturas quanto à matriz de covariância: (a) uma distribuição gaussiana “hiperesférica”, para a qual é fixado $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$; (b) uma distribuição gaussiana de variáveis supostamente independentes, onde $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\sigma})^2$; (c) uma gaussiana arbitrária, para a qual a única restrição é que $\boldsymbol{\Sigma}$ seja semidefinida positiva. Ao se incluírem mais bases ($K \geq 2$) na mistura, além destes casos, podemos ainda fixar ou não matrizes de covariância comuns, isto é, (d) $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma} \quad \forall k$, reduzindo o número de parâmetros a serem estimados, porém ainda mantendo uma boa capacidade de representação.

Os parâmetros do modelo podem ser obtidos através de estimadores de máxima verossimilhança (MLE), utilizando o algoritmo *expectation-maximization* (EM), caso a mistura contenha mais de uma componente [6]. Um índice natural para a detecção de novidades para este tipo de modelo é dado por $-\log(p(\mathbf{x}|\hat{\boldsymbol{\theta}}_{ML}))$.

E. Kernel PCA

Em [11], a PCA foi generalizada, de forma não-linear, considerando transformações de vetores $\mathbf{x} \in \mathbf{R}^D$, através de um mapeamento $\phi : \mathbf{R}^D \rightarrow \mathcal{F}$, onde \mathcal{F} é potencialmente de dimensão infinita. O teorema de Mercer [12] afirma que uma função $k(\mathbf{x}, \mathbf{y})$ simétrica e semidefinida positiva pode ser expressa como um produto interno, ou seja, $k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$, para algum mapeamento ϕ . Funções que satisfazem a estes requisitos são chamadas funções de *kernel* de Mercer, como, por exemplo, a função RBF gaussiana $k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$, onde $\gamma > 0$, doravante chamado de precisão, é que define sua largura. Utilizando esta propriedade, conhecida como truque do *kernel*, o trabalho mostrou que tais componentes principais não-lineares, originalmente obtidas através da decomposição da matriz de covariância dos dados implicitamente mapeados no espaço das características, pode ser simplesmente obtida pela decomposição em autovalores e autovetores da matriz de Gram, que é formada pelas avaliações da função de *kernel* de Mercer para pares de dados do conjunto de treino.

Assim, de forma similar à PCA, onde o erro de reconstrução da aproximação de posto reduzido pode ser usado como uma medida do grau de anormalidade, o mesmo indicador pode ser utilizado para a técnica da *kernel* PCA [13].

IV. CONSIDERAÇÕES METODOLÓGICAS

Cada tipo de modelo ou técnica apresenta um conjunto de hiperparâmetros que devem ser definidos. Para esta

Tabela I. HIPERPARÂMETROS DOS TIPOS DE MODELO/TÉCNICA.

Tipo	Hiperparâmetros	Valores testados
PCA	Número de bases L	10, 15, 20, ..., 90, 95, 99% de variância explicada
k-NN	Núm. de vizinhos k	1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 40, 50
Curva Principal	Núm. de segmentos k	10, 20, 30, ..., 90, 100% de até 99 segmentos
Mistura de Gaussianas	Núm. de componentes K	1, 2, 3, 4, 5, 6, 7
	Estrutura de $\boldsymbol{\Sigma}_k$	(a), (b), (c), (d) ¹
KPCA	Número de bases L	10, 15, 20, ..., 95, 100% de variância explicada
	Precisão do kernel RBF γ	0,01, 0,1, 0,5, 1

aplicação, tais parâmetros devem ser ajustados com o objetivo de otimizar uma função objetivo relacionada a uma eficaz detecção de novidades. Neste caso, duas funções objetivo de interesse são: (i) a área sob a curva ROC (AUC); e (ii) a taxa de detecção de dados desconhecidos (UDR), obtida ao se adotar um valor específico de limiar, o qual garanta uma dada taxa mínima de detecção de classes conhecidas, aqui arbitrada como 95%. A diferença essencial entre as duas propostas é que: (i) incentiva detectores com bom funcionamento em todos os possíveis pontos de operação, enquanto a métrica (ii) pode ser mais interessante do ponto de vista prático.

Fatalmente, em situações reais, não temos acesso a dados das classes desconhecidas, impossibilitando a seleção de hiperparâmetros pela otimização de alguma função objetivo através de ensaios de validação cruzada. Torna-se necessário, assim, utilizar métodos heurísticos. No caso de múltiplas classes conhecidas, pode-se considerar o ajuste dos hiperparâmetros de cada detector tratando-se as classes restantes como desconhecidas. Por exemplo: “Se as classes A, B e C são conhecidas, ao ajustar o detector da classe A, os dados das classes B e C representam classes desconhecidas”. No caso em que apenas uma classe é conhecida, a heurística escolhida para a definição de hiperparâmetros é específica ao tipo de detector. Por exemplo, para detectores probabilísticos, métodos tradicionais para a seleção e/ou comparação de modelos podem ser empregados, a exemplo do critério de informação de Akaike e de testes de razão de verossimilhança [14]. Para detectores baseados em erros de reconstrução, pode-se fazer uso de *scree plots*, *gap statistic* ou modelos para análise de *change-point* [15] para se identificar à partir de que ponto um aumento da complexidade do modelo não é justificável perante a redução do erro associada².

A Tabela I apresenta os tipos de modelo e hiperparâmetros considerados nos experimentos. Buscou-se explorar hiperparâmetros que implicam em modelos de diferentes regimes de comportamento. Nos modelos PCA e KPCA, o número de bases é selecionado, indiretamente, através do nível de variância explicada desejado, pois estes valores são diretamente relacionados. Por outro lado, o algoritmo

¹As diferentes estruturas consideradas foram apresentadas na Subseção III-D.

²Esta ideia é equivalente à lei econômica dos rendimentos decrescentes (*law of diminishing returns*).

de extração de curvas principais é incremental, sendo estabelecido um limite máximo de 99 segmentos por curva de classe. Ocorre, no entanto, que pode ocorrer uma convergência antecipada. Por este motivo, foram considerados percentuais deste valor máximo de segmentos.

V. CUSTO COMPUTACIONAL

Além da performance do ponto de vista da detecção de novidades, medida através do critério AUC ou UDR, é de grande importância prática o custo computacional de operação do detector. É possível que um algoritmo mais acurado não atenda às especificações necessárias quanto ao uso de memória e/ou tempo de processamento. Assim, um compromisso entre o custo computacional e performance de detecção deve ser atentado para viabilizar a aplicação em tempo real do sistema detector. Os Algoritmos 1–5 apresentam pseudo-códigos para os detectores discutidos. O custo computacional de cada linha dos algoritmos é apresentado em termos de *flops*³ ou complexidade temporal utilizando a notação “big-o” (\mathcal{O}). A complexidade temporal total de cada algoritmo também é apresentada, porém a comparação entre elas é difícil pois são impactadas pela escolha dos hiperparâmetros associados. Assim, testes são experimentos específicos relevantes a esta análise, porém fogem ao escopo deste trabalho.

VI. EXPERIMENTOS E RESULTADOS

Os dados das 8 classes foram particionados em 10 subconjuntos, seguindo um esquema de validação cruzada de 10-*fold* estratificados. Em cada *fold*, foram treinados detectores de diversos tipos empregando os valores de hiperparâmetros apresentados na Tabela I. Ao fim do processo de projeto, para cada classe haviam 10 detectores de cada tipo (modelo-hiperparametrização). Estes detectores foram avaliados quanto às duas medidas de performance apresentadas na Seção IV nas respectivas partições de teste, considerando os dados das outras classes como classes desconhecidas.

Os melhores resultados para cada combinação de fator (critério-classe-modelo) são apresentados na Tabela II e Figura 1, com os respectivos hiperparâmetros dados na Tabela III. Além da média e desvio padrão dos valores de AUC e UDR obtidos, a Tabela II também indica o modelo que apresenta maior média de performance. Na maioria das classes, o método de vizinhos mais próximos (k-NN) apresenta maiores médias, porém uma análise estatística com mais amostras deve ser conduzida a fim de elucidar se o método é sistematicamente superior aos outros. É interessante notar que em todos os casos houve concordância entre a AUC e UDR na seleção do “melhor” tipo de modelo e, ademais, seus hiperparâmetros são iguais ou semelhantes, indicando uma boa concordância entre ambos indicadores

VII. CONCLUSÃO

O artigo apresentou uma comparação exploratória entre diferentes técnicas para a detecção de novidades em

sinais de sonar passivo. Verificou-se que para todas as 8 classes de dados é possível obter detectores com, em média, um valor de AUC superior à 0,95, e UDR superior à 85%. Também se observou uma correlação entre as escolhas de tipos de modelos e seus hiperparâmetros para as métricas AUC e UDR avaliadas, indicando a possibilidade de se desenvolverem protocolos para a criação de detectores que maximizem as duas métricas simultaneamente. Estas medidas de performance foram levantadas considerando-se experimentos do tipo “um-contra-todos”, isto é, onde a classe conhecida é comparada com todos os dados das outras classes, e, portanto, classes facilmente separáveis e numerosas tendem a enviesar positivamente os resultados. Uma análise mais robusta envolvendo a performance para as classes duas a duas deve ser considerada futuramente.

AGRADECIMENTOS

Os autores agradecem ao CNPq, à CAPES e à FAPERJ pelo apoio financeiro e infraestrutural. Ao IPqM, agradecemos pela cessão da base de dados e cooperação técnica.

REFERÊNCIAS

- [1] A. Y. Ng and M. Jordan, “On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes,” in *NIPS*, vol. 14, 2001.
- [2] J. B. O. Souza Filho, “Classificação neural de sinais de sonar passivo,” Ph.D. dissertation, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2007.
- [3] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, “A review of novelty detection,” *Signal Processing*, vol. 99, pp. 215–249, 2014.
- [4] W. A. Struzinski and E. D. Lowe, “A performance comparison of four noise background normalization schemes proposed for signal detection systems,” *The Journal of the Acoustical Society of America*, vol. 76, no. 6, pp. 1738–1742, 1984.
- [5] T. Fawcett, “An introduction to roc analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [6] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [7] N. Bhatia *et al.*, “Survey of nearest neighbor techniques,” *arXiv preprint arXiv:1007.0085*, 2010.
- [8] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The elements of statistical learning*. Springer, 2009, vol. 2, no. 1.
- [9] R. Tibshirani, “Principal curves revisited,” *Statistics and Computing*, vol. 2, no. 4, pp. 183–190, 1992.
- [10] J. J. Verbeek, N. Vlassis, and B. Kröse, “A k-segments algorithm for finding principal curves,” *Pattern Recognition Letters*, vol. 23, no. 8, pp. 1009–1017, 2002.
- [11] B. Schölkopf, A. Smola, and K.-R. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [12] T. Hofmann, B. Schölkopf, and A. J. Smola, “Kernel methods in machine learning,” *The annals of statistics*, pp. 1171–1220, 2008.
- [13] H. Hoffmann, “Kernel pca for novelty detection,” *Pattern Recognition*, vol. 40, no. 3, pp. 863–874, 2007.
- [14] D. Posada and T. R. Buckley, “Model selection and model averaging in phylogenetics: advantages of akaike information criterion and bayesian approaches over likelihood ratio tests,” *Systematic biology*, vol. 53, no. 5, pp. 793–808, 2004.
- [15] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

³Definido como uma soma, multiplicação ou aplicação de uma função transcendental de números em formato de ponto-flutuante.

Tabela II. MÉDIA E DESVIO PADRÃO DAS AMOSTRAS DE MEDIDAS DE PERFORMANCE “ÁREA SOB A CURVA ROC” (AUC) E “TAXA DE DETECÇÃO DE CLASSES DESCONHECIDAS” (UDR) PARA OS DETECTORES DE CADA CLASSE.

		PCA	k-NN	Curva Principal	M. de Gauss.	KPCA
A	AUC	0.9595 ± 0.005	*0.9872 ± 0.002	0.9791 ± 0.004	0.9842 ± 0.004	0.9828 ± 0.004
	UDR	0.8386 ± 0.028	*0.9543 ± 0.010	0.9109 ± 0.020	0.9310 ± 0.016	0.9261 ± 0.013
B	AUC	0.9588 ± 0.004	*0.9914 ± 0.001	0.9565 ± 0.005	0.9783 ± 0.005	0.9594 ± 0.004
	UDR	0.8203 ± 0.032	*0.9648 ± 0.005	0.7672 ± 0.029	0.9093 ± 0.032	0.8234 ± 0.029
C	AUC	0.8763 ± 0.008	*0.9541 ± 0.005	0.9327 ± 0.006	0.9110 ± 0.008	0.9186 ± 0.007
	UDR	0.5830 ± 0.033	*0.8571 ± 0.024	0.7030 ± 0.028	0.6737 ± 0.035	0.6876 ± 0.024
D	AUC	0.9878 ± 0.003	0.9878 ± 0.002	0.9920 ± 0.002	*0.9931 ± 0.002	0.9930 ± 0.002
	UDR	0.9454 ± 0.019	0.9698 ± 0.006	0.9690 ± 0.011	*0.9774 ± 0.008	0.9723 ± 0.009
E	AUC	0.9526 ± 0.003	*0.9809 ± 0.002	0.9633 ± 0.005	0.9709 ± 0.003	0.9685 ± 0.002
	UDR	0.7764 ± 0.026	*0.9476 ± 0.007	0.8402 ± 0.030	0.8878 ± 0.022	0.8581 ± 0.010
F	AUC	0.9762 ± 0.004	*0.9958 ± 0.001	0.9858 ± 0.003	0.9903 ± 0.002	0.9834 ± 0.003
	UDR	0.8971 ± 0.016	*0.9842 ± 0.004	0.9312 ± 0.021	0.9623 ± 0.012	0.9257 ± 0.015
G	AUC	0.9670 ± 0.007	0.9818 ± 0.005	0.9536 ± 0.006	0.9836 ± 0.002	*0.9867 ± 0.003
	UDR	0.8686 ± 0.024	0.9265 ± 0.010	0.8671 ± 0.019	0.9434 ± 0.009	*0.9506 ± 0.006
H	AUC	0.9215 ± 0.007	*0.9728 ± 0.003	0.9615 ± 0.004	0.9455 ± 0.004	0.9581 ± 0.004
	UDR	0.7395 ± 0.015	*0.8917 ± 0.015	0.8404 ± 0.017	0.8098 ± 0.014	0.8415 ± 0.020

Tabela III. RESPECTIVOS HIPERPARÂMETROS DOS DETECTORES COM PERFORMANCE APRESENTADA NA TABELA II.

		PCA	k-NN	Curva Principal	M. de Gauss.	KPCA
A	AUC	30%	10	100%	7/(b)	60%/1,00
	UDR	45%	4	100%	7/(c)	80%/0,50
B	AUC	80%	2	100%	7/(c)	80%/0,10
	UDR	80%	2	100%	7/(c)	85%/0,10
C	AUC	50%	2	100%	7/(c)	60%/1,00
	UDR	65%	2	100%	7/(c)	60%/1,00
D	AUC	20%	50	100%	7/(a)	95%/1,00
	UDR	20%	50	100%	7/(b)	95%/1,00
E	AUC	85%	2	100%	7/(c)	75%/1,00
	UDR	85%	4	100%	7/(c)	75%/1,00
F	AUC	55%	2	100%	7/(c)	50%/1,00
	UDR	55%	2	100%	7/(c)	55%/1,00
G	AUC	30%	20	90%	7/(d)	100%/0,01
	UDR	30%	20	100%	1/(b)	100%/0,01
H	AUC	35%	10	100%	7/(d)	60%/1,00
	UDR	40%	10	100%	3/(c)	60%/1,00

Algoritmo 1 Função de score baseado na PCA.

```

1: function SCORE-PCA( $\mathbf{x}_*, \bar{\mathbf{x}} \in \mathbb{R}^D, \mathbf{W} \in \mathbb{R}^{L \times D}$ )
2:    $\tilde{\mathbf{x}}_* \leftarrow \mathbf{x}_* - \bar{\mathbf{x}}$   $\triangleright \approx D$ 
3:    $\mathbf{z}_* \leftarrow \tilde{\mathbf{x}}_* \mathbf{W}^T$   $\triangleright (2D - 1)L$ 
4:    $r_* \leftarrow \|\tilde{\mathbf{x}}_*\|^2 - \|\mathbf{z}_*\|^2$   $\triangleright \approx 5D$ 
5:   return  $r_*$   $\triangleright \mathcal{O}(DL)$ 

```

Algoritmo 2 Função de score baseado no método k-NN.

```

1: function SCORE-KNN( $\mathbf{x}_* \in \mathbb{R}^D, \mathbf{X} \in \mathbb{R}^{N \times D}, k$ )
2:   for i=1 to N do  $\triangleright \approx 5ND$ 
3:      $d_i \leftarrow \|\mathbf{x}_* - \mathbf{x}_i\|^2$   $\triangleright \approx 5D$ 
4:    $s \leftarrow \text{partial-sort}(\mathbf{d}, k)$   $\triangleright \mathcal{O}(D \log(k))$ 
5:   if  $k$  is even then  $\triangleright cte$ 
6:      $h \leftarrow k/2$   $\triangleright cte$ 
7:      $r_* \leftarrow (d_h + d_{h+1})/2$   $\triangleright cte$ 
8:   else
9:      $h \leftarrow \text{round}(k/2)$   $\triangleright cte$ 
10:     $r_* \leftarrow d_h$ 
11:   return  $r_*$   $\triangleright \mathcal{O}(ND)$ 

```

Algoritmo 3 Função de score baseado no método da Curva Principal.

```

1: function SCORE-CP( $\mathbf{x}_* \in \mathbb{R}^D, \mathbf{V}, \mathbf{U} \in \mathbb{R}^{k \times D}, \mathbf{b} \in \mathbb{R}^k$ )
2:   for i=1 to k do  $\triangleright \approx 12kD$ 
3:      $\mathbf{w} \leftarrow \mathbf{x}_* - \mathbf{v}_i$   $\triangleright \approx D$ 
4:      $\tilde{t} \leftarrow \mathbf{w} \mathbf{u}_i^T$   $\triangleright \approx 2D$ 
5:      $t \leftarrow \min(\max(0, \tilde{t}), b_i)$   $\triangleright cte$ 
6:      $\mathbf{p} \leftarrow \mathbf{v}_i + t \mathbf{u}_i$   $\triangleright \approx 2D$ 
7:      $d_i \leftarrow \|\mathbf{x}_* - \mathbf{p}\|^2$   $\triangleright \approx 7D$ 
8:    $r_* \leftarrow \min(\mathbf{d})$   $\triangleright \mathcal{O}(k)$ 
9:   return  $r_*$   $\triangleright \mathcal{O}(kD)$ 

```

Algoritmo 4 Função de score baseado na KPCA.

```

1: function SCORE-KPCA( $\mathbf{x}_* \in \mathbb{R}^D, \mathbf{X} \in \mathbb{R}^{N \times D}, \mathbf{W} \in \mathbb{R}^{L \times N}, \gamma, \mathbf{m}_S = \mathbf{1}_N \mathbf{K} / N, m_{SS} = \mathbf{1}_N \mathbf{K} \mathbf{1}_N^T / N^2$ )
2:   for i=1 to N do  $\triangleright \approx 5ND$ 
3:      $k_{*i} \leftarrow \exp(-\gamma \|\mathbf{x}_* - \mathbf{x}_i\|^2)$   $\triangleright \approx 5D$ 
4:    $m_* \leftarrow \text{mean}(\mathbf{k}_*)$   $\triangleright \approx N$ 
5:    $p_* \leftarrow 1 - 2m_* + m_{SS}$   $\triangleright cte$ 
6:    $\mathbf{k}_* \leftarrow \mathbf{k}_* - m_* \mathbf{1}_N - \mathbf{m}_S + m_{SS} \mathbf{1}_N$   $\triangleright \approx 3N$ 
7:    $\mathbf{z}_* \leftarrow \tilde{\mathbf{k}}_* \mathbf{W}^T$   $\triangleright (2N - 1)L$ 
8:    $r_* \leftarrow p_* - \|\mathbf{z}_*\|^2$   $\triangleright 3N$ 
9:   return  $r_*$   $\triangleright \mathcal{O}(ND)$  ou  $\mathcal{O}(NL)(L > D)$ 

```

Algoritmo 5 Função de score baseado em Mistura de Gaussianas.

```

1: function SCORE-MDG( $\mathbf{x}_* \in \mathbb{R}^D, \{\mu_k \in \mathbb{R}^D, \Sigma_k^{-1} \in \mathbb{R}^{D \times D}\}_{k=1}^K$ )
2:   for i=1 to K do  $\triangleright (a) \approx 3KD, (b) \approx 4KD$ 
3:      $\triangleright (c)/(d) K(D^2 + 3D)$ 
4:      $l_k \leftarrow -\frac{1}{2}(\mathbf{x} - \mu_k) \Sigma_k^{-1} (\mathbf{x} - \mu_k) + cte_k$ 
5:    $r_* \leftarrow -\log\text{-sum-exp}(\mathbf{l})$   $\triangleright \approx 3K$ 
6:   return  $r_*$   $\triangleright (a)/(b) \mathcal{O}(KD)$ 
7:    $\triangleright (c)/(d) \mathcal{O}(KD^2)$ 

```

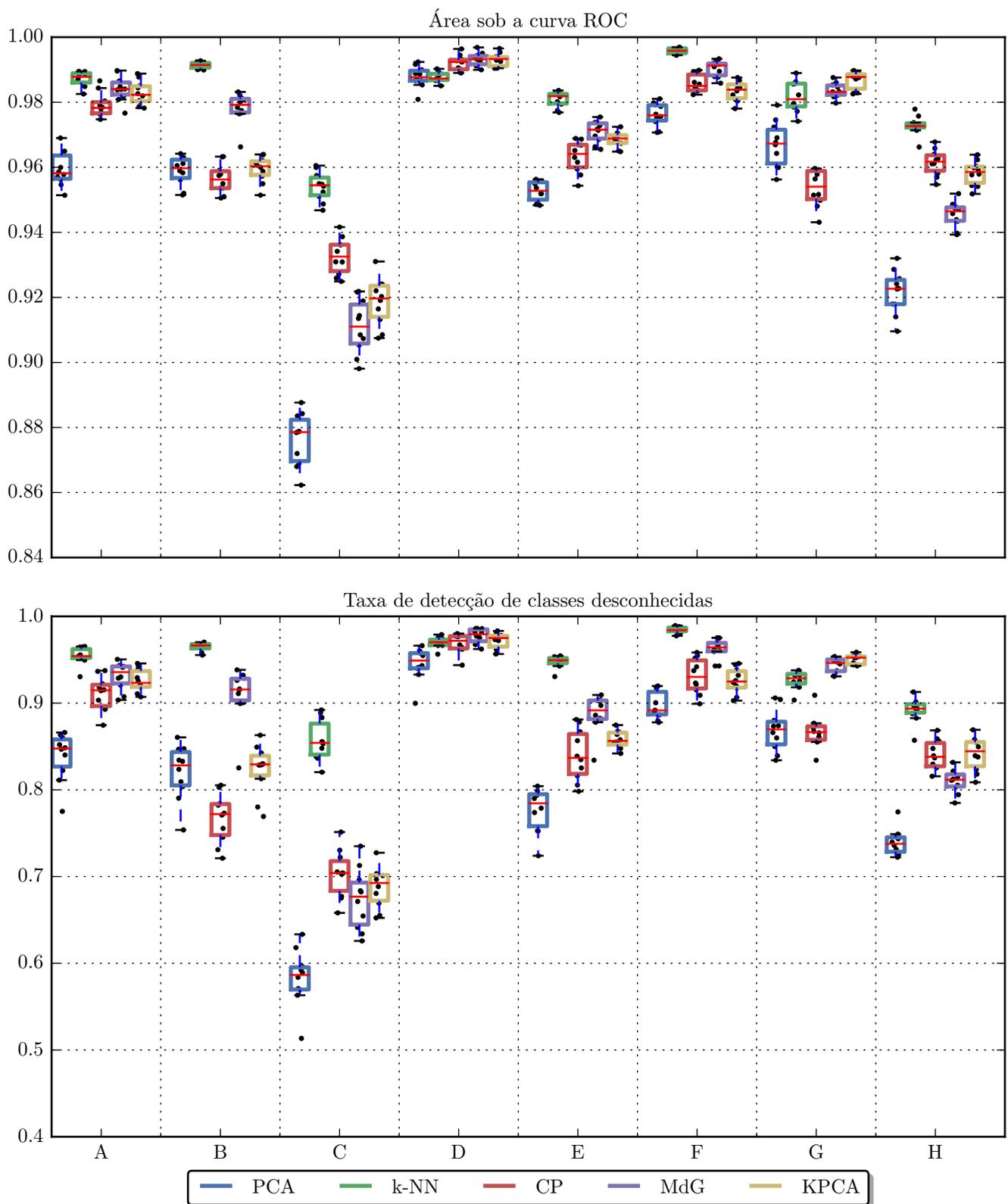


Figura 1. Performance dos melhores classificadores de cada tipo.